Università degli Studi di Camerino

School of Advanced Studies

Dottorato di Ricerca in Scienze e Tecnologie
Computer Science - XXXIV Ciclo

Knowledge Graphs, Automatic Feature Engineering and Machine Learning in Algorithmic Trading for Financial Markets

Relatore

Prof. Andrea Polini

Dottorando

Zaigham Abbas

Commissione Esaminatrice

Anno Accademico 2019-2022

University of Camerino

School of Advanced Studies

Doctor of Philosophy in Sciences and Technology

Computer Science - XXXIV Cycle

Knowledge Graphs, Automatic Feature Engineering and Machine Learning in Algorithmic Trading for Financial Markets

Supervisor

Prof. Andrea Polini

PhD Candidate

Zaigham Abbas

Doctoral Examination Committee

Academic Year 2019-2022

"If my mind can conceive it, if my heart can believe it then I can achieve it."

-Muhammad Ali

"I Dedicate this thesis to my Inspiring Father Haji Muhammad Shahbaz (Late), Mother Tahira Shahbaz, My Beloved Queen Fizzah Zaigham, Two princesses Shahezamn Abbas and Aira Abbas. "

-Yours Zaigham Abbas

Acknowledgments

I sincerely appreciate the esteemed PhD supervisor, Dr. Andrea Polini, whose invaluable guidance has been instrumental in shaping this research endeavor. His unwavering commitment to supporting early career researchers and exceptional expertise make him a true role model in the competitive realm of academia. His roadmap and the sharpening of my skills under his mentorship have fueled my dedicated efforts throughout this journey. In addition to my supervisor, I thank Dr. Massimo Callisto De Donato, Dr. Emanuele Pomante, and Cristina Soave for their significant contributions and insightful suggestions regarding the Corporate and Academic World. Their guidance has played a pivotal role in refining my research and development work, leaving an indelible mark on my academic pursuits. As I conclude this extensive Ph.D. journey, I take pride in my accomplishments and recognize their transformative impact on my research abilities and personal growth. The enduring commitment and determination demonstrated by those who supported me have profoundly influenced my character. I express heartfelt thanks to my parents, whose teachings on hard work and unwavering devotion to overcoming challenges have been a guiding light. Their dedication and resilience continue to inspire me as I reflect on my achievements. Gratitude also extends to my family, including my siblings and friends, who stood by me through thick and thin, offering moral support and celebrating each milestone. Special thanks to Mr. Ahmad Ali, Mr. Zahir Abbas, Mr. Muhammad Usman Mir, and Zohaib Hussain for being unwavering support pillars throughout this academic journey. Your encouragement and belief in me have strengthened and inspired me.

Thanks!

Abstract

Thesis consists of novel ideas and practical implementations 1.Automatic Features extractions using knowledge graph, proceeds new dataset with AutoML and Standard Machine Learning Pipeline for prediction. These new features enhance the predictive power of Machine Learning Algorithms to improve the accuracy. 2.Realtime price prediction of Cryptocurrency using ML algorithms. Existing Studies only predict the price of Cryptocurrency for static datasets. The proposed AI signal Machine predicts price of BTC for four timeframes of 1Minute,3 Minute ,5 Minute and 15 Minute simultaneously.3.Developed AI Automated Trading Algorithm which can be used for trading any instrument like Gold, Stock not limited to crypto currency. The algorithm is based on Feature Engineering, the features extracted using Automatic Features Engineering using knowledge graph.

Contents

List of Figures

List of Tables

PART I

INTRODUCTION

Chapter 1

Introduction

In the contemporary era of technological advancement, marked by the proliferation of innovations such as machine learning, cryptocurrency, digital markets, and the metaverse, the pursuit of rapid wealth accumulation has become a pervasive aspiration. Traditional business avenues have witnessed a transformation, as the allure of financial markets, renowned for their speed and volatility, promises earnings within minutes and seconds. Central to the realm of financial markets, the cryptocurrency market has emerged as a dominant force fueled by the groundbreaking concept of blockchain technology and its global accessibility. Operating incessantly, 24 hours a day and seven days a week, the crypto market has become a focal point for traders seeking swift returns on investment. However, the arduous nature of manual trading, demanding constant vigilance and rapid decision-making, poses significant challenges for individuals seeking to capitalize on financial market opportunities.

1.1 Motivations

The advent of algorithmic trading bots, powered by machine learning concepts, has revolutionized the landscape of financial trading. These AI-driven bots offer automated trading solutions, enabling traders to execute transactions without the need for continuous monitoring. The rationale for employing such bots is compelling: they operate with precision and objectivity, devoid of the emotional biases that often cloud human judgment. A critical examination comparing human traders to algorithmic trading bots underscores the vast disparity in trading capacity. While a human trader may be limited by physical and mental constraints, an algorithmic bot operates tirelessly, round the clock, throughout the year. In a typical workweek, a human trader may log approximately 40 hours of trading activity.

In contrast, an algorithmic trading bot operates non-stop, accumulating an astounding 8760 hours of trading activity annually, equivalent to approximately 4.2 times the trading capacity of a human counterpart. This thesis embarks on an exploration of the paradigm shift instigated by algorithmic trading bots within the context of cryptocurrency markets, analyzing the manifold advantages they offer over traditional trading methods. By harnessing the power of machine learning and automation, these bots exemplify the convergence of technological innovation and financial efficiency, reshaping the landscape of modern trading practices. The thesis unfolds across six comprehensive chapters, each contributing to a deeper understanding of the integration of knowledge-driven approaches and advanced machine-learning techniques in enhancing trading efficiency and automation in cryptocurrency markets. Chapter 1 lays the foundation by elucidating fundamental concepts of knowledge graphs and exploring their structure, semantics, and applications. Chapter 2 delves into the implementation of an automatic knowledge graph, highlighting systematic feature engineering and the extraction of influential features categorized based on their proximity to the root node. In Chapter 5, the focus shifts to Real-Time AI-trading signal Machine, where regression-based machine-learning models are trained using various methodologies alongside automated machine-learning techniques. This chapter also details the analysis of real-time data sourced from Binance and hyper parameter optimization using Optuna. Chapter 6 elaborates on the workings of the Time Machine Bot, an algorithmic bot driven by insights garnered from automatic machine learning. The bot facilitates automated trading decisions, leveraging indicators derived from the feature set and operating continuously with a target profit margin of 3% on a 24/7 basis. Through this comprehensive examination, this thesis underscores the transformative potential of integrating knowledge-driven approaches with advanced machine-learning techniques in revolutionizing cryptocurrency trading.

1.2 Contribution

- **Automatic Knowledge Graph and Feature Engineering**: Propose the implementation of an automated knowledge graph and methodical feature engineering procedure to extract supplementary features, classifying them according to their proximity from the root node. By identifying influential features and optimising feature selection, this methodology enhances predictive capabilities.

- **Systematic Literate Review (SLR)**: Performs an exhaustive Systematic Literature Review (SLR) on Automatic Feature Engineering and Automatic Knowledge Graph, detailing the development of these

concepts.

- **AI Real-Time Signal Machine**: The AI Real-Time Signal Machine is an innovative system that analyses real-time data from Binance using machine learning and automated machine learning techniques. It provides enhanced predictive capabilities to cryptocurrency trading.

- **Novel Trading Algorithm**: Presents an innovative trading algorithm that optimizes trading strategies over financial markets through the use of feature engineering, with the goal of enhancing trading decisions in response to market conditions.

1.3 Research Questions

1.Goal: Improved Decision-Making:
Research Question: How can we leverage knowledge graphs to enhance the decision-making capabilities of trading bots?
Contribution: By integrating various data sources and representing complex relationships, knowledge graphs can provide a richer informational context, leading to more informed and potentially more profitable trading decisions.
2. Goal: Real-Time Data Processing and Analysis:
Research Question: How can knowledge graphs help in processing and analyzing real-time data for trading strategies?
Contribution: Knowledge graphs can handle and integrate real-time data feeds, enabling the trading bot to respond swiftly to market changes with a comprehensive understanding of the context.
3.Goal: Risk Management and Anomaly Detection:
Research Question: In what ways can knowledge graphs aid in identifying risks and anomalies in trading activities?
Contribution: The ability of knowledge graphs to represent and analyze complex relationships can help in detecting unusual patterns and potential risks, thus enhancing the risk management strategies of the trading bot.
4.Goal: Enhanced Predictive Modelling:
Research Question: How can the incorporation of knowledge graphs improve the predictive models used by trading bots?
Contribution: Knowledge graphs enable the incorporation of a broader range of factors and their interconnections into predictive models, potentially leading to more accurate forecasts of market movements. Using knowledge graphs for representing knowledge in automatic trading bots offers a robust framework for integrating diverse data

sources, understanding complex relationships, and enhancing the analytical capabilities of the bot. These features align well with the primary goals and research questions in the domain, ultimately aiming to create more intelligent, responsive, and profitable trading strategies. some research questions specifically tailored for the development and improvement of automatic trading bots:

Market Data Analysis and Integration

1. How can we effectively integrate diverse data sources (e.g., financial news, social media sentiment, economic indicators) into a unified framework for an automatic trading bot?

This question explores methods to combine structured and unstructured data to enhance the bot's decision-making capabilities.

2. What are the most effective techniques for real-time data processing and analysis in the context of automatic trading? Investigating real-time data handling and analysis techniques to ensure the trading bot can respond promptly to market changes.

Machine Learning and Predictive Analytics

3. How can machine learning models be optimized for predicting short-term and long-term market trends?

- Focusing on improving predictive accuracy for different time horizons using advanced machine learning models.

4. What role do Machine learning and reinforcement learning play in the development of more sophisticated trading strategies?

- Examining the application of Machine learning and reinforcement learning in formulating advanced trading strategies.

Risk Management and Anomaly Detection

5. What methods can be used to enhance risk management strategies in automatic trading bots?

- Developing and testing new risk management techniques to minimize potential losses.

6. How can anomaly detection algorithms be applied to identify and mitigate fraudulent or erroneous trades?

- Exploring the use of anomaly detection to safeguard against fraud and errors.

Algorithmic Trading Strategies

7.What are the most effective algorithmic trading strategies for different market conditions (e.g., bullish, bearish, volatile)?

- Investigating the performance of various algorithmic strategies under different market scenarios.

8. How can sentiment analysis from social media and news impact trading strategy formulation?

- Studying the influence of sentiment analysis on trading strategies

and its potential to predict market movements.

Performance Evaluation and Optimization

9. What metrics should be used to evaluate the performance of an automatic trading bot?

- Identifying appropriate performance metrics and evaluation frameworks.

10. How can automatic trading bots be optimized for execution efficiency and latency reduction?

- Investigating methods to reduce execution latency and enhance the efficiency of trading operations.

Knowledge Graphs and Data Representation

11. How can knowledge graphs be utilized to represent complex relationships and dependencies in financial data for automatic trading?

- Investigating the application of knowledge graphs for a more nuanced understanding of market relationships.

12. What are the benefits of using knowledge graphs over traditional data representation models in the context of automatic trading?

- Comparing the effectiveness of knowledge graphs with other data models in enhancing trading bot performance.

Chapter 2

Introduction To Knowledge Graphs

2.1 History of Knowledge Graph

When discussing how to construct modular educational systems for courses, the Austrian linguist Edgar W. Schneider first used the word in 1972. To make it easier to perform algebras on graphs, researchers at the Universities of Groningen and Twente launched a cooperative project in the late 1980s called Knowledge Graphs, which focuses on creating semantic networks with edges restricted to a limited range of relations. There became less of a clear-cut difference between knowledge graphs and semantic networks in the following decades.[1] Wordnet, initially developed in English, was recognized as one of the earliest knowledge graphs in 1985. Wordnet database uses natural language processing to catalog semantic associations between words with structural characteristics. These similarities exist regardless of the language in which the words are conveyed. Wordnet is a database that associates words with their related words and contains hyponyms, meronyms, and synonyms. Wordnet can be recognized as a dictionary because synonyms use definitions and examples. This database is accessible to users via a web browser, and its role can be seen reflected in apps that deal with text analysis and artificial intelligence. [2] Twenty years later, in 2005, a man named Mark Wirk introduced another knowledge network. This one was called Geonames. A geographical database known as Geonames documented the relationships between various geographic names and locations and the linked entities. In 2007, two knowledge warehouses based on graphs were established to acquire generic knowledge. On the one hand, there was DBpedia, which obtained all its information from Wikipedia. On the other side, there was a database called Freebase that had a variety of datasets that were avail-

able to the public. These two datasets didn't contain knowledge graphs, but they defined and developed concepts related to one another. After another five years, in 2012, Google released its first knowledge graphs built on DBpedia and Freebase. After a few years had passed, they absorbed RDF, which was an extension of attributes that could be added to HTML, XHTML, and XML-based document formats to embed metadata in Web publications. In addition to this, they implemented microdata formats directly from web pages, which made use of vocabulary that was provided by schema.org. As more time passed, Google's knowledge graphs evolved into a complementary feature to the company's string-based search interface, and more individuals began to use the latter. Since that time, a great number of major corporations have begun to unveil their very own knowledge graphs. Some of the most well-known of these corporations include Facebook, LinkedIn, Airbnb, Microsoft, Amazon, Uber, and eBay.[3]

2.2 Introduction

Graphs are highly adaptable in presenting a wide range of data, excelling at visually representing intricate system architectures and interrelationships[4]. Over the last decade, considerable effort has been invested in developing search engines that can query graphs and comprehend their semantic relationships. This study contributes to the continuous efforts that seek to optimize the capacity of graphs to depict human knowledge[5]. A subset of RDF2 graphs is indispensable when developing knowledge bases. By ensuring semantic consistency in constructing knowledge repositories, ontologies improve accessibility[6]. RDF graphs enable the systematic depiction of domain expertise after establishing an ontology. Subsequent developments in machine learning facilitate the retrieval of concealed data from these graphs, thereby resolving the difficulty of constructing significant representations during entities and relationships. Subsequently, these representations may be employed to deduce new connections within a graph or to classify unidentified nodes.[7] In today's information-based society, data is the fuel that keeps everything moving. It is the fuel that drives today's information economy. Insights, well-considered judgments, and technical progress are all locked away in data. Our information and advancements in the digital age are only possible with data[8]. Information, whether numerical, linguistic, pictorial, auditory, or otherwise, is the building block of the data. These facts and figures, sometimes known as "data points," are the foundation for all subsequent learning and comprehension. There are two main types of data: structured, like database entries, and unstructured, like the words in a book or the pixels in an image. Information is the driving force behind many operations in the computer and IT industries[9]. Data is gathered,

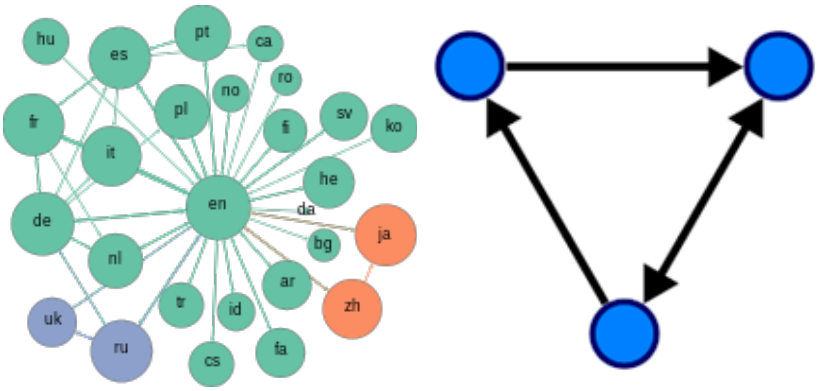
saved, processed, and analyzed so that conclusions can be drawn, decisions can be made, and intelligent applications may be run. Knowledge graphs, which provide a systematic approach to describe and connect data points to construct a web of knowledge, are one of the exciting ways data is used in the digital world. Knowledge graphs serve as a road map of interrelated data points for use in any number of contexts, including but not limited to research, corporate intelligence, and general information retrieval[10]. Understanding the basics of data is crucial to appreciating the value and potential of knowledge graphs. This chapter lays the groundwork for further investigation into knowledge graphs as we delve into the complexities of data and its revolutionary role in the modern digital era.

2.3 Understanding Graphs

2.3.1 Knowledge Graph

A visual created to represent words, particularly the relationship between two or more quantities, is known as a graph as shown in Figure 2.1a and Figure 2.1b.

A graph is a type of data structure that is used in the fields of mathematics and computer science. It comprises a collection of nodes (also known as vertices) connected by edges (also known as links). Every edge creates a relationship between two nodes, demonstrating that those nodes are linked or connected somehow.



(a) Knowledge Graph (b) Knowledge Node Connection

Figure 2.1: Knowledge Graph and Node Connection

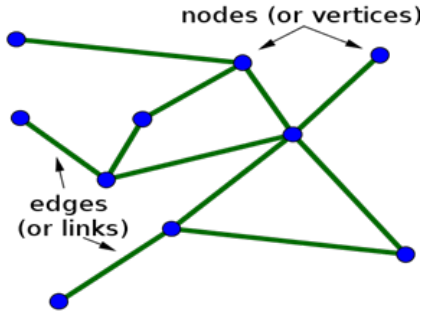


Figure 2.2: Knowledge Graph Nodes

Node

Nodes are the essential building blocks of a graph. They are often referred to as vertices. They are meant to symbolize separate entities or points of interest. One way to think of nodes as data points, while another is as locations. Nodes could stand in for a wide variety of things across various applications as shown in Figure 2.2. This includes cities on a map, individuals on a social network, and variables in a mathematical model. [1]

Edges

Edges, called links, are the connections or relationships between two graph nodes. They are symbolic representations of the relationships[11], or interactions, between the entities portrayed by the nodes. Edges in a graph can be directed or undirected, weighted or unweighted, and their meanings might vary depending on the context in which the graph is being interpreted. For instance, edges in a graph representing a social network[12] might stand for undirected friendships. In contrast, edges representing a transportation network denote roads or routes and are normally intended to be followed as shown in Figure 2.3 [2] Graph technologies have emerged as a game-changing means for businesses of all sizes to solve needs that can't be adequately addressed by traditional approaches, at least not effectively[13]. Gartner has chosen graphs as one of their top analytics and data trends for the past two years due to their substantial potential for disruption. In the modern world, businesses understand they must innovate or risk falling behind the competition.[14, 15] Relationships and linkages between different data elements can be visualized using graphs. Data analysis can take advantage of the connections and relationships in question[16]. A significant amount of data is interconnected, and graphs are gaining more and more significance because they make it simpler to investigate these interconnections and arrive at

[1] $https://mathinsight.org/definition/network_node$
[2] https://mathworld.wolfram.com

fresh inferences.In Figure 3.2 Relationships can be represented using graph models obtained from graphs and graph databases. They enable users to perform pattern recognition, classification, statistical analysis, and machine learning on these models, enabling more efficient analysis at scale when applied to huge amounts of data[17]. Algorithms investigate the pathways and distance between the vertices, the relevance of the vertices, and the clustering of the vertices when it comes to the analysis of graphs. When identifying a node's significance, the algorithms will frequently consider incoming edges, the relevance of neighboring vertices, and several other signs. Queries and algorithms that use the connection between vertices can be executed in a matter of sub-seconds rather than hours or days, thanks to graph databases explicitly preserving relationships. Users no longer need to run many joins, and the data can be used more for analysis and machine learning. This makes it possible to understand more about the world in which we live.

A knowledge graph is a knowledge base that integrates information through a graph-structured data model or topology[18]. Knowledge graphs are commonly used to hold interrelated descriptions of phenomena, such as things, events, circumstances, or abstract concepts, while also capturing the

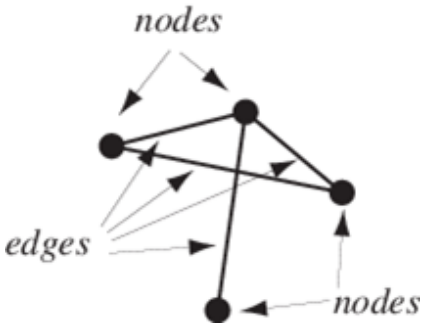


Figure 2.3: Knowledge Graph Edges

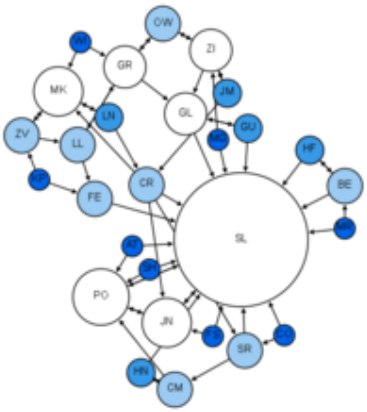


Figure 2.4: Knowledge Graph

semantics behind the terminology employed. KG consists of interlinked descriptions of entities, objects, events, and concepts. The knowledge graph is a framework for Linking semantic metadata that provides data integration, analytics, unification, and sharing[19]. Descriptions are formal semantics that process the information efficiently and ambiguously. Entity descriptions form a network in which each entity represents a related description of entities and the context of their interpretation.[20] Since the advent of the Semantic Web, knowledge graphs have frequently been linked to open data initiatives. These projects center on the connections that exist between various concepts and entities. They are also prominently associated with and used by search engines such as Google, Bing, Yext, and Yahoo; knowledge engines and question-answering services such as Wolfram Alpha, Apple's Siri, and Amazon Alexa; and social networks like LinkedIn and Facebook. In addition, they are prominently associated with and used by search engines such as Google, Bing, Yext, and Yahoo.

2.3.2 The Birth of Machine Learning: Arthur Lee Samuel's Pioneering Work

With his pioneering self-learning English draughts software, IBM computer scientist Arthur Lee Samuel coined "machine learning" in 1959. This distinguished machine learning from classical AI, which focused on mimicking sophisticated cognitive tasks like reasoning and language processing. The paradigm change that emphasized machine learning directly from data defined machine learning and sparked controversies[21]. To gain insights from data, machine learning practitioners used pattern recognition, a subfield of statistics. Pattern recognition, which finds latent data features, inspired machine learning algorithms.

2.3.3 Evolution of Data Representation in Machine Learning

Early machine learning relied less on prior knowledge and used pattern recognition rules like attribute-value or propositional data. This pattern persisted in the 20th century[22]. Around the turn of the 21st century, interest in more expressive data representations and models that could handle complex data structures returned. The knowledge graph is a potential data architecture for representing complex data interactions. This newfound interest in knowledge graphs and expressive data models marks a major shift in machine learning [19]. Google's 2012 use of the term "knowledge graph" to describe a data architecture in which knowledge, information, and data [23] are encoded in graph form has helped popularize the word. The adaptabil-

ity of graph models relieves knowledge engineers of the obligation to force their information into a predetermined shape. It allows them to organize their data in any way they see suitable. This natural manner of modeling knowledge allows knowledge engineers to articulate complicated items with minimal information loss.

2.3.4 Significance of Knowledge Graphs

Knowledge graphs, which have recently become an important component of artificial intelligence, have been the subject of much academic investigation. [10]Even though knowledge graphs are inextricably tied to machine learning and Semantic Web technologies, big data and analytics, and cloud computing, the concept of a knowledge graph has existed since 1972. When there was a demand for instructional systems development for different types of classes, the conversation around knowledge graphs began. The universities of Groningen and Twente in the Netherlands initiated a workshop known as Knowledge Graphs around the end of 1980. The workshop aimed to design semantic networks with limited relations solely to simplify algebras on the graph. As a result of this endeavor, semantic networks, and knowledge graphs[10] became confused with one another.

2.3.5 Knowledge Graphs in the Semantic Web and Beyond

Knowledge graphs, which are the most relevant exponent of the application of Semantic Web technologies, have gained speed as a result of the explosion of available data and the necessity for expressive formalisms to combine factual knowledge that is scattered across a variety of data sources .The increasing number of scientific publications that reference public knowledge graphs like DBpedia2 and Wikidata3 and the growing number of daily hits on these sites provide evidence of the spectrum of opportunities these graphs bring into the industrial, public administration, and scientific landscape[10]. Even though these findings vouch for the viability of Semantic Web technologies, they also call for creating computational tools that will enable knowledge graphs to accommodate the massive increase in data anticipated over the next few years. According to Bizer et al. (2011), the first step that needs to be solved to start viewing the Web as one integrated overall database is the proposal of reliable methods capable of integrating numerous data sources across the Web.

2.4 Key characteristics

The knowledge graph contains the following data management paradigms. Database data can be manipulated via structured queries[24]. The graph is a collection of nodes that can be analyzed as a network data structure. The knowledge base comprises formal semantics, which can be used to infer new facts and interpretations. Knowledge graphs are represented in RDF, the best framework combining data integration, unification, linking, and reuse[25].
Expressivity: The standards Semantic web stack RDF and OWL. OWL is the language that represents various types of data or content. Content comprises data schema, metadata, taxonomies, and vocabularies reference to master data. The RDF is the language which is used to model structured metadata.
Performance: it is an important metric in KG to efficiently manage graphs, graphs containing billions of nodes, facts, and properties. Interoperability is specifications for data serialization, access, and management (SPARQL Graph Store). Globally unique identifiers are used to facilitate data integration and publishing.
Standardization: all the requirements from logicians to enterprise data management professionals and system operations teams are satisfied according to the W3C community process. [3]

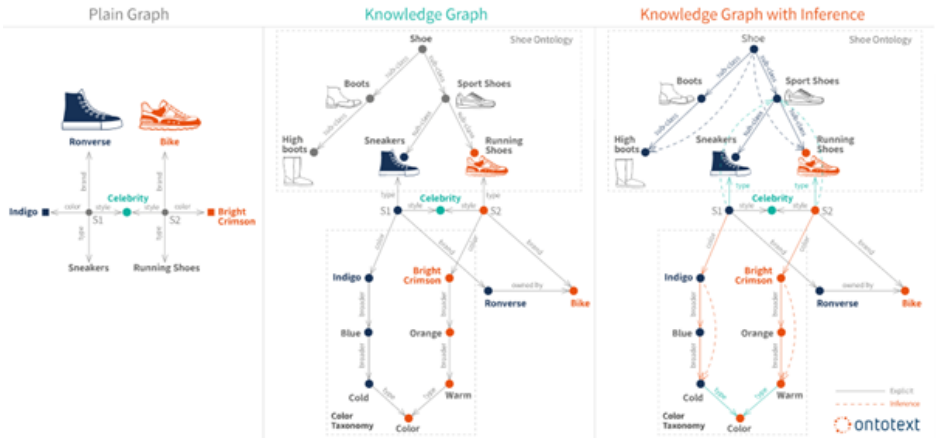


Figure 2.5: Example of Knowledge Graph

[3]https://www.ontotext.com/knowledgehub/fundamentals/what-is-a-knowledge-graph/

2.5 Ontologies and Formal Semantics

Knowledge graphs and ontologies are fundamental components that underpin the structure and depiction of data in the semantic web domain. Notwithstanding the continuous discourse surrounding their subtle distinctions, both are indispensable constituents striving to formalize entities within a graph[26]. Ontologies frequently founded on taxonomies such as the Web Ontology Language (OWL), which is extensively utilized, provide a structured schema. They fulfill a pivotal function within knowledge graphs by serving as the contractual link that connects the graph to the semantic significance of its data[27]. Ontologies, which comprise categories, classes, relationship types, and free text descriptions, offer a comprehensive structure for the accurate analysis and interpretation of data. In contrast to popular belief, not all graphs possess the qualities that define a knowledge graph. For example, statistical RDF data might not possess the essential level of semantic profundity. One possible limitation of knowledge graphs is that they may capture extra semantic information, leading to redundant data and heightened intricacy[26]. Prominent instances of large knowledge graphs comprise Fact Forge, Google Knowledge Graph, DBpedia, Geonames, and WordNet, all of which contribute substantial domain expertise. RDF and RDF graph databases are essential in integrating and extracting diverse data from numerous sources, guaranteeing the resilient management of enormous datasets. Incorporating text-mining methodologies improves knowledge graphs' functionalities by extracting significant insights from factual data and methodically presenting them. Using semantic annotations obtained through text analysis enhances knowledge graphs' functionality in search, visualization, and analytics. Knowledge graphs find wide-ranging utility in various domains, including but not limited to recommendation systems, semantic search, investment market intelligence, and advanced drug safety analytics. Knowledge graphs remedy difficulties associated with content labeling, classification, and recommendations by providing interconnected descriptions of entities. Fundamentally, knowledge graphs and ontologies establish a mutually beneficial association, collaborating to enhance the accuracy, consistency, and integration of information depiction within the continuously developing domain of the semantic web. [4]

[4] *https://accidental-taxonomist.blogspot.com/2019/05/knowledge-graphs-and-ontologies.html*

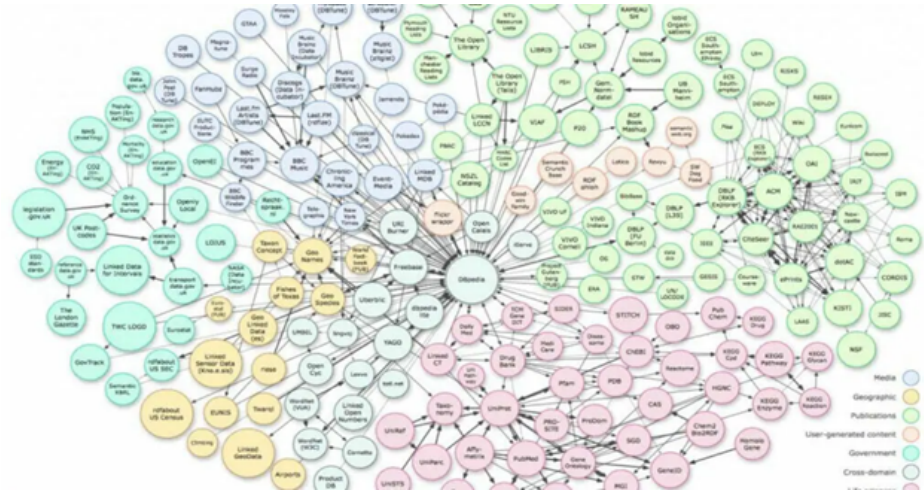


Figure 2.6: Example of Knowledge Graph

2.6 Methodology of Building Knowledge Graph

There are ten steps to follow to create a KG. In the first step, we clarified the business domain and elicited the requirement. Requirements belong to the domain experts and the knowledge domain. The collection of the requirements is based on Establishing the goal. The requirements Are the most important component in building our knowledge graph. We collect the information and define this research question to be answered. In step two, we gather and analyze the relevant data; we discover the data set and the relevant information, which is open source and commercially available at our convenience. Data would facilitate the developer in achieving its goal in terms of domain, scope, and maintenance. The third step is data cleansing to ensure its quality. Data preprocessing Is the most complicated and complex job. It takes 80% of the whole effort to develop a knowledge graph. In this phase, we remove the redundant and duplicate data to maintain quality and integrate[28] it to make it applicable to the task. This includes removing meaningless or invalid entries, fixing inconsistencies, and adjusting the data fields. In the first step, we get a semantic data model by analyzing it thoroughly. The purpose of this model is to prepare for harmonizing the data. Ontologies are Engineered by the domain experts Using RDF. Formalize the data models. Enable standards such as OWL and RDF schema. Determine which datasets, taxonomies, and other types of information (proprietary, open, or commercially available) would be most helpful to you in achieving your goal in terms of domain, scope, provenance, maintenance, and other factors. Fixing any data quality problems will make it more relevant to

the work you need to do. This includes eliminating entries that are invalid or have no meaning, modifying data fields so that they may support numerous values, resolving inconsistencies, and other similar tasks. Conduct a careful analysis of the various data schemata to prepare for the data harmonization process. Reuse or reengineer ontologies, application profiles, RDF shapes, or any other mechanism to use them together in conjunction with one another. Create a formal representation of your data using industry standards such as RDF Schema and OWL. In the fifth step, we integrate data with etl or virtualization by applying etl tools. Etl tools convert data to RDF. Further, it is also used by noETL, OBDA, and GraphQL Technologies for data virtualization's usability. Semantic metadata is created for easier data re-usability, update, and discovery. In the sixth step, we harmonize the data via fusion reconciliation and alignment. The same entity Descriptions match across the data set. It handles overlapping scopes and attributes and merges the information to map their taxonomies. In the seventh step, we Formulate the data management and search layer by merging different graphs via RDF. The data is stored in graph DB efficiently by keeping the Semantics of the data model in place. It provides consistency, reasoning, and checking the validation of the data. It is a cluster that efficiently assists us in data search with search engines and harnesses its performance. In the eight steps, we create graphs by applying reasoning and text analytics techniques. We extract the data from the entities and their relationships from the text data. We create new information with inference and graph analytics techniques. Now, the graph has more meaningful information than the previous graph. The graph is well connected, well managed, and has more content ready for deeper analytics. In the ninth step, we maximize the usability of the data and are ready to provide the answers to the original questions Via SPARQL queries. It ensures the data is FAIR. It is easy to use, finds the optimal semantic search, and provides GraphQL interface end data visualization. In the final step, the knowledge graph is ready to use and is easy to maintain and evolve.KG Always needs to be updated and maintained according to the new requirements So it is never outdated. Knowledge graphs are the ideal tool for knowledge reasoning, such as making inferences from the data represented in machine learning. In a crucial blog post 2012, Google announced that its knowledge graph would allow users to search for objects, not strings. Previously, when a user searched for "Taj Mahal," the search engine tried to match both parts (words) with all the documents it possessed. The user is looking for a specific "thing," such as the Agra monument, an Indian restaurant, or a Grammy–winning performer. The results customers want when looking for certain objects include celebrities, cities, geographical characteristics, events, and movies. Getting back information that is relevant to the inquiry improves the user experience. Google used it in its core business, web search.

A structured knowledge card is a small box containing summary information about things that may correspond to the search query. The search was the start. After Google's blog post, knowledge graphs appeared in databases, the Semantic Web, AI, social media, and enterprise information systems. Various initiatives have extended and developed Google's basic concepts over time. With new methodologies and technologies, the knowledge graph concept has become substantially larger. Various types of entities and their attributes make up a knowledge graph. According to this definition, an entity can be an individual, a company, a site (like the Taj Mahal), a sporting event, or even a book or movie (like in the case of a recommendation engine). There are a variety of attributes that could be assigned to an object. When it comes to an individual, these characteristics include their name, address, birth date, and so on. Relationships connect entities. For example, a person who works for a corporation likes a page or follows another individual. Using relations to connect two different knowledge graphs is another useful application of this concept. Knowledge graphs differ from other knowledge-oriented information systems because they use a unique set of structures and reasoning to store and use knowledge. Some of these structures and reasoning are languages, schemas, standard vocabularies, and hierarchies of concepts. Information management is a set of steps (how information is ingested and transformed into a knowledge graph) Patterns that can be accessed and processed, such as query mechanisms, search algorithms, and pre-and post-processing techniques. Knowledge graphs are often represented using a Resource Description Framework (RDF) data format, but we will use a label property graph (LPG) instead.[29] As a W3C standard for data interchange, RDF is a language for expressing web resources, such as the name of a web page or copyright and license information for web content. RDF can represent information about other objects, such as an online shop's products or a user's preferences for information delivery, by generalizing the concept of a web resource. Any RDF statement is made up of triples, each consisting of a subject, predicate,[30] and object. The subject is a resource (a node in the graph), the predicate is an arc (a relationship), and the object is another node or a literal value. An RDF document is meant to be processed by applications rather than just displayed to users. So that data may be exchanged between applications, it provides a common framework for representing data. An LPG is designed to store complex graphs by compactly leveraging relationships and nodes with their characteristics.LPGs can express knowledge graphs better than RDF graphs. This chapter focuses on creating and accessing a knowledge graph using textual data. Making knowledge graphs from structured data is much easier. After obtaining a knowledge graph from text, it is postprocessed or enriched to extract insights and wisdom. Recognition of key entities and their relationships will be incorporated into the techniques for extracting

structures from text. A knowledge graph can only be created using these strategies. It is important[31] to infer a general model that encapsulates this information, abstracting from the instances in the text, because the same entities and connections tend to recur. An inferred knowledge graph is the name given to this model. Machine learning techniques and AI applications can all benefit from this knowledge base, which can be employed in various ways. A semantic network is one of the most prevalent ways to express a knowledge base.

2.6.1 Knowledge graph building: Entities

When creating a knowledge graph from textual data, it is essential to recognize the entities in the text. Consider the following scenario: you have a set of papers (for example, from Wikipedia), and you are tasked with identifying the names of persons or other entities related to your domain mentioned in those articles, such as locations, organizations, and so on. When this information has been extracted, it is necessary to make it easily available using a graph for further research. Recognizing and classifying named entities in texts is the goal of named entity recognition (NER) [32].According to the domain in which you are working, you should include things like street addresses and times as well as chemical formulas and mathematical formulas in your NE types; in short, anything important to your application. Anything that can be identified by its proper name and relevant to our analysis can be considered a NNE. It's possible to recognize patients, ailments, treatments, drugs, hospitals, and so on if you're processing electronic medical information for some healthcare use case, for example. Extralinguistic structure is a term that refers to the fact that many named items are composed according to rules that differ from the general laws of language. Non-entities like dates, periods, and currencies can also fall under the umbrella of the term "entity" in some contexts. Types like PERSON, DATE, NUMBER, or LOCATION are all associated with specific NEs. Different sorts of an entity can be associated with it depending on its domain. For example, when all the nouns (NEs) in a text are retrieved, we may conclude that "United Airlines," "United," and "United" all refer to the same corporation .Assume for a moment that you have this document: Figure 2.18 Pierre and Marie Curie's wife received the Nobel Prize in Chemistry in 1911 for her pioneering work. In 1903, she received the Nobel Prize in Physics for her work in atomic physics.[33] If we are interested in all the entities, a proper NE recognition result should be able to recognize and classify the names "Marie Curie" and "Pierre Curie" as person names, the names "Nobel Prize in Chemistry" and "Nobel Prize in Physics" as prize names, and "1911" and "1903" as dates. In contrast to people, machines have a difficult time with this task. Use an NE visualizer like the free and open-source display to have a go. Although the awards

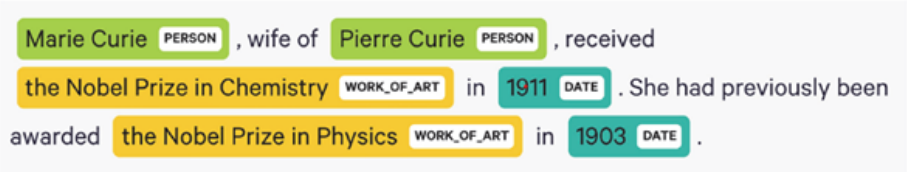


Figure 2.7: Marie Curie

are considered "pieces of art," it's interesting to note that the service could identify all the sentence elements without any tuning. The NER job can easily be added to the graph model. New nodes with the label NamedEntity should be added, including the entity's name.

Extraction of the document's entities. In the case of Marie Curie, for example, "Marie" and "Curie" are two Tag Occurrence nodes linked to each other. In the Named Entity nodes, "Marie Curie" can appear multiple times in the text. "Marie Curie" as a Person is a special node that can be linked to all of them. Our data model may easily be extended to store the results of an NER task as a graph. As you can see, the modifications are minimal in the pipeline and the code for saving the NER result. SpaCy includes its own basic NE models, which we utilized in this code, but it also allows you to create your own. 12.1 NER task addition to model. Extracts and stores named entities as a new phase. The function extracts named entities from the NLP result. The function saves the graph's entities. [34]The query cycles across the entities, creating new nodes and linking them to the NE's tags and creating a K-graph Entities 397 train new NER models with annotated sentence samples. See the spaCy documentation. The name is commonly abbreviated when a person, place, or thing is mentioned more than once. We might see a shortened name ("Mme. Curie"), a pronoun ("she"), or a descriptive phrase ("the recognized scientist") instead. The issue now is identifying and extracting these associations from plain text. We can add another requirement to our scenario. Assume you want to optimize access patterns by including all named entity mentions. The following phrase connects "she" with "Marie Curie" as an example: In 1903, Marie Curie won the Nobel Prize for Physics. She was the first woman and individual (male or female) to win the award twice. This is done by coreference resolution, which is the challenge of recognizing relationships between entity references in a text, whether they are nouns or pronouns. We prefer subjects over objects, words closer to the pronoun in the text, and words that can plausibly arise in the context of the pronoun. However, the ultimate criteria for coreference resolution are usually based on gathering statistics across many texts from the corpus or utilizing machine learning classifiers. Co-referential word linking is more challenging. A few simple examples employ the same

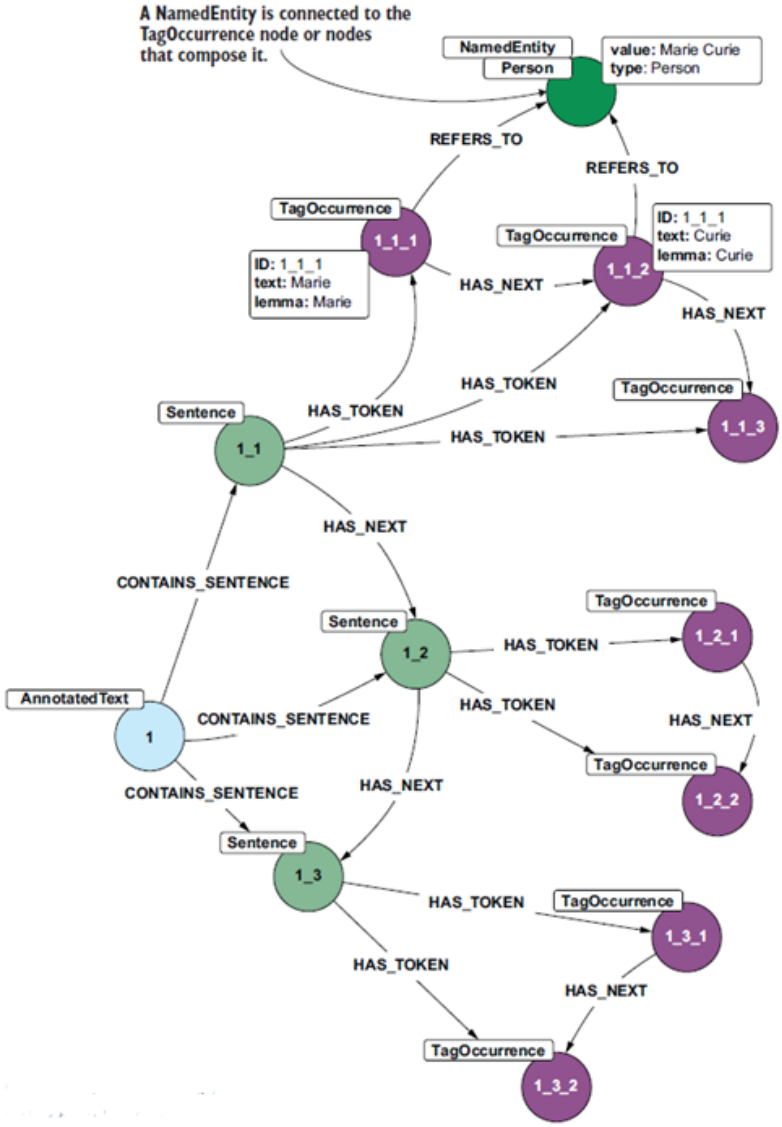


Figure 2.8: Graph Model with co-reference Resolution

noun multiple times, but most involve a basic understanding of the world, depending on what words have been used elsewhere. This method converts "the recognized Polish scientist" to "Marie Curie" and "the prize" to "the Nobel Prize." This book does not cover these methods because the NLP libraries used in our codebase have their own implementations for coreference resolution. The focus is on how to model and make the most of the results.

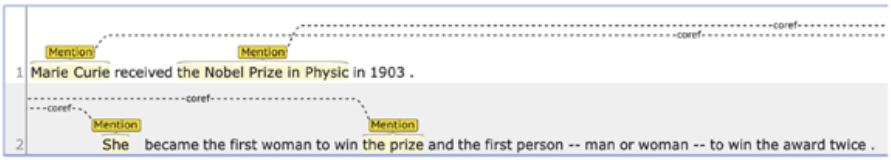


Figure 2.9: Example of Marie Curie

For example, we can link pronouns and other references to the real entities they refer to in our network model. A co-reference resolution has been added to our model in Figure. When it comes to adapting a model to new requirements, graphs provide the required flexibility to do so with minimal effort while maintaining the existing access patterns. Named Entity nodes are connected via the MENTIONS relationship. When the key NEs are not explicitly named, the coreference connections can link them to the sources. NEs and coreferences are vital in knowledge graph construction. Both are first-class objects that represent entity occurrences in text and their relationships. To increase the graph's knowledge extraction quality, we must abstract from these text occurrences and repeatedly identify the essential entities alluded to in the text. The discourse model is a mental model that the system builds incrementally as it processes text from a corpus (or, in the case of a human hearer, from a dialogue) and contains representations of the entities referred to in the text, as well as properties of the entities and relations among them. Two mentions co-refer if they refer to the same entity. The discourse model concept can be used to increase access to knowledge in a knowledge network. As shown in the Figure, we can construct an inferred knowledge graph containing unique representations of the entities alluded to in the processed text and their attributes and connections.

In contrast to the main body of the knowledge graph, which consists of the decomposition of text in the corpus and, eventually, the structured data reorganized in the graph model, this second part, which is connected to the first, provides answers to different questions and supports multiple services without the need to navigate the graph. This inferred knowledge network comprises an easy-to-share representation of knowledge not directly tied to the individual instances (the documents) from which this knowledge is extracted. You can do the following queries in our graph database: Find the distinct types of named entities produced. Count the times each kind is used,

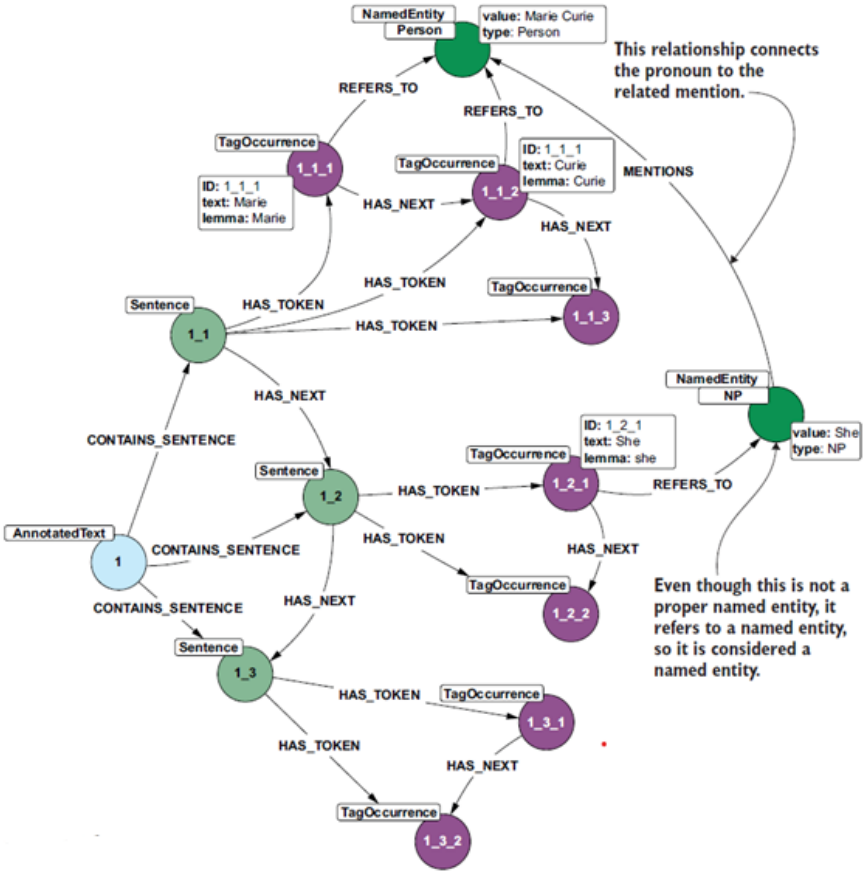


Figure 2.10: Example of Marie Curie

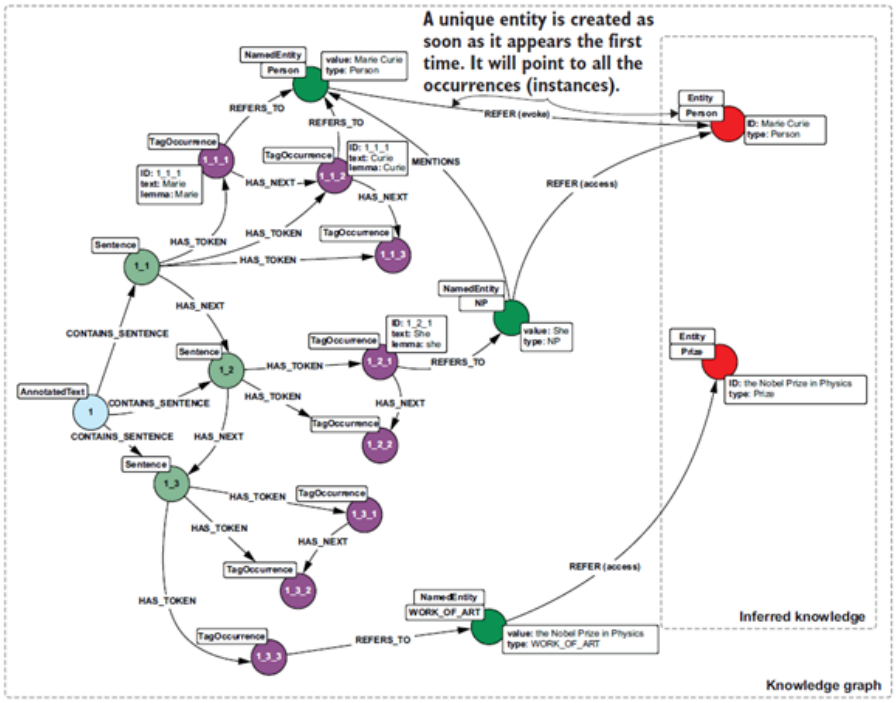


Figure 2.11: Resulting Graph

then take the first three. Count the number of instances of Organization entities in the inferred knowledge network. How many people are there? Because the system should aggregate these while generating the inferred graph, fewer of them should exist.

2.6.2 Knowledge graph building: Relationships

A knowledge graph's value is greatly enhanced when the relationships between the entities that have been identified can be detected, as this greatly enhances both the insights that can be obtained from the graph and the accessible access patterns. This step is essential to generate a meaningful graph from a text since it allows you to connect the entities and browse them effectively. The variety of queries you can run, and the variety of questions it can answer will greatly rise[35]. In order to make the content more accessible for both humans and machines, consider identifying relationships between the extracted entities that highlight links between them—such as linkages between a reward and who received it, or between a corporation and the people who work there. You must be able to identify individual entities and the links between them to answer questions like, "What is the most likely diagnosis based on the symptoms a patient has?" There are various methods for accomplishing this work, some more complicated than others, and some ne-

cessitate supervision (labeling sample relationships to create a training set). Regarding relation extraction, lexico-syntactic patterns are the most common. Syntactic relationships between tokens or specified tag sequences are mapped into a collection of meaningful (for the use case) relations between key named entities[36]. This task may appear difficult, and it is in some cases, but having a graph model aids greatly. A set of semantic analysis rules can approximate this by mapping a subgraph of the syntactic graph (such as a piece of the graph having syntactic relationships between important entities) into a database relation applied to the associated entities. Consider the following sentence: Marie Curie received the Nobel Prize in Physics in 1903. According to the results of the syntactic analysis, the verb "received" has the subject "Marie Curie" and the object "the Nobel Prize." A possible pattern would map this syntactic structure into the semantic predicate.

```
(verb: receive, subject: p: Person, object: a: Prize) •
(relationship:\Rightarrow RECEIVE_PRIZE, from p, to:a)
```

Here, "receive" is a verb in English, while RECEIVE PRIZE is a relationship type (a semantic relation) in English. Cypher, a graph-based query language, makes expressing these kinds of patterns easy. We're going to make a graph out of the data we have.

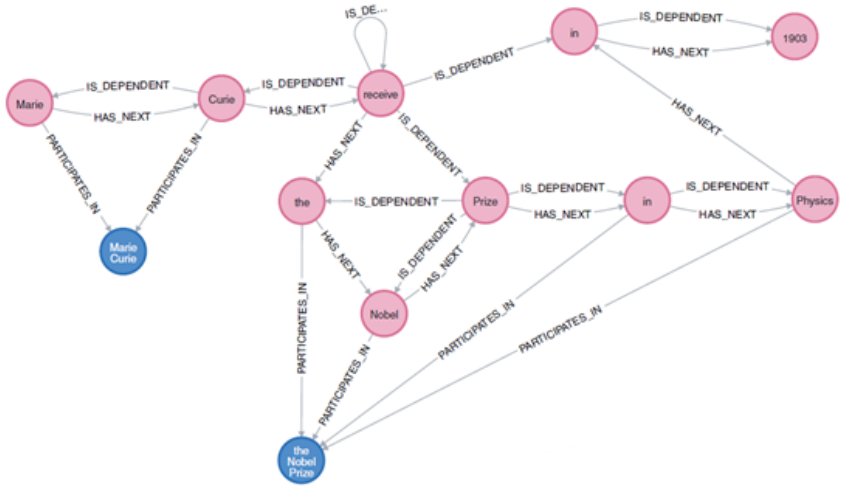


Figure 2.12: graph-based query language

It is essential to obtain all the sub graphs matching the pattern we are looking for with the following query.

```
MATCH (verb: TagOccurrence {pos: "VBD," lemma:"receive"})
WITH verb
MATCH p=(verb)-[:IS_DEPENDENT {type:"nsubj"}]->
```

```
(subject)-[:PARTICIPATES_IN]->
\Rightarrow (person:NamedEntity {type: "PERSON"})
MATCH q=(verb)-[:IS_DEPENDENT {type:"dobj"}]->
(object)-[:PARTICIPATES_IN]->
 (woa:NamedEntity {type: "WORK_OF_ART"})
RETURN verb, person, woa, p, q
```

Figure 2.13: Query

For example, the pattern must define that Person and Prize are the subject and object, respectively; this is done by specifying their semantic kinds, which is to say, the entity type. For our case, the type of art is "piece of art," and having a better NER model would be helpful. When it comes to receiving, "receive" indicates a wide range of relationships, and we don't want those relationships to be converted into the RECEIVE PRIZE relationship. Many alternative patterns are needed to capture the wide range of expressions representing this information, such as using various words and phrases.

```
(relationship: "win", subject: p:Person, object: a:Prize) →
(relationship: RECEIVE_PRIZE, from: p, to:a)
(relationship: "award", indirect-object: p:Person, object: a:Prize) →
(relationship: RECEIVE_PRIZE, from: p, to:a)
```

It's worth noting that ("The Committee awarded the prize to Marie Curie") is the indirect object in this case. As well as a pattern for the active form ("The prize was awarded to Marie Curie"), we'd need a syntactic regularization phase to turn passive phrases into active ones.

```
(relationship: "was awarded", object: a:Prize,
indirect-object: p:Person) →
(relationship: RECEIVE_PRIZE, from: p, to: a)
```

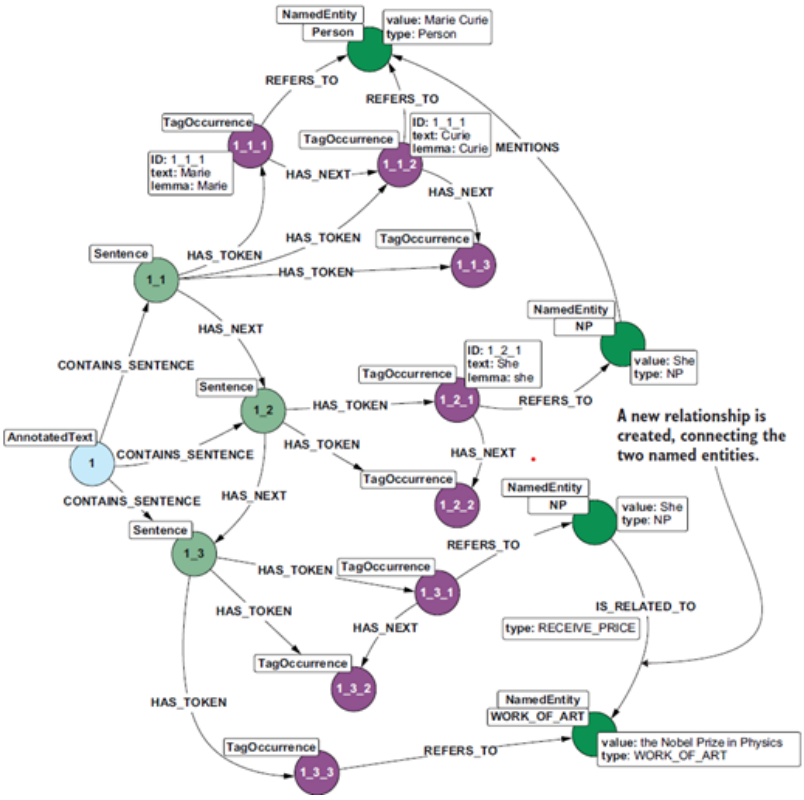


Figure 2.14: Example of Marie Curie

After extracting these relationships, they must be stored in the graph model we created. Schema changes are shown in

After extracting these relationships, they must be stored in the graph model we created. Schema changes are shown in Figure. A list of the rules for transforming semantic relationships into relationships of interest poses a dual problem: Many ways of expressing semantic relationships make it difficult to comprehensively address them. For example, a single predicate may have various meanings in different contexts. For example, if we're compiling records of military operations, we'll likely want to include references to "strike" and "hit" in war literature but not in sports accounts. Alternatively, we may want to require some arguments and allow others to be left up to the reader. Other methods may be employed to overcome these issues. Most of these methods are supervised, requiring human guidance to learn. The most typical method uses classifiers to determine the link (or lack thereof) between NEs. It can be done using machine learning or deep learning algorithms, but what should the input be? A corpus of entities and relations is needed to train the classifier. Then, for each pair of entities in a sentence, we either record the relation connecting them or indicate that they are not connected. The former are positive training examples, whereas the latter are negative. Applying the classifier to a fresh test text extracts associations between entity mentions in the same sentence. Despite its effectiveness, this method has a major drawback: the volume of data necessary for training. A third strategy combines pattern-based and supervised approaches, employing pattern-based to bootstrap the classifier. If the interconnections are kept as connections between NEs in the knowledge graph representing the text, it is possible to project these relationships onto the previously stated inferred knowledge network. Here, the links should connect entities[37]. In defining the model, remember that the relationship must be defined. One issue is that most graph databases today, including Neo4j, only allow two-way relationships. But in this situation, we want to connect more and return to the source. Solves this problem: It's possible to trace the relationship between two entities by adding characteristics such as a list of IDs identifying the NEs connected. Include some information in the relationship as characteristics that will help us to trace back to the point of genesis of the connection between two entities, such as a list of IDs denoting the NEs that are connected. Create nodes to represent the relationships between the nodes. However, these Relationship nodes will materialize the connections between entities, and we can connect them to other nodes, even though we can't connect to relationships. As a result, the recommended schema (Figure) is based on the second method, which is less verbose regarding the number of nodes but significantly harder to navigate. Inferred knowledge graphs significantly impact your graph's navigability and the types of access patterns it can allow.

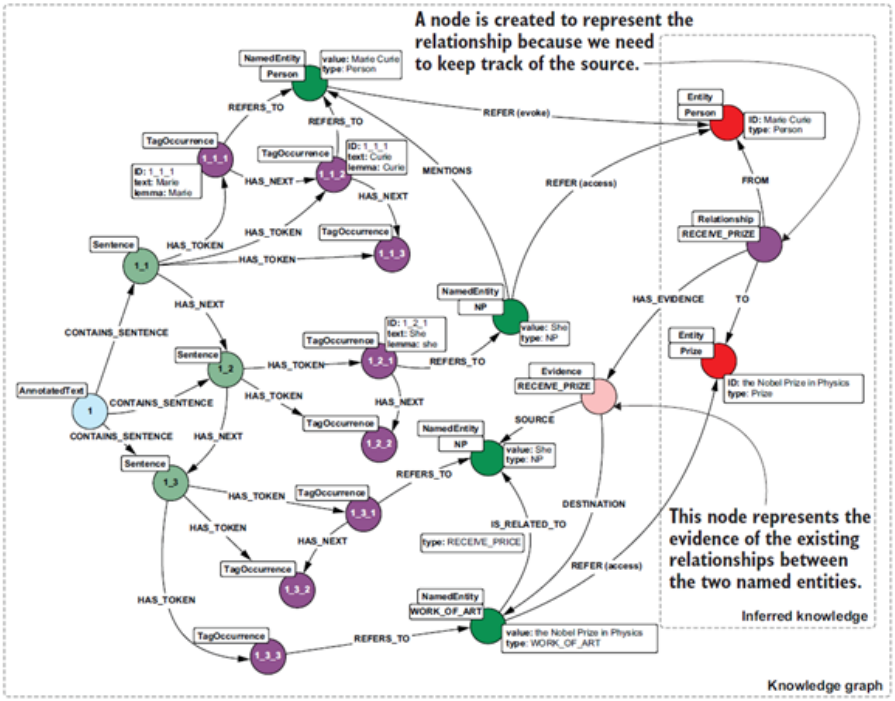


Figure 2.15: Example of Marie Curie

In our example with Marie Curie, suppose you've analyzed a big corpus of writings and want to discover which physicists have earned a Nobel Prize.

```
MATCH (Nobel Prize:Entity {type:"WORK_OF_ART"})
<-[:TO]-(rel:Relationship
\Rightarrow {type: "RECEIVE_PRIZE"})-[:FROM]->
(winner:Entity {type: "PERSON"})
WHERE Nobel prize.id CONTAINS
"the Nobel Prize in Physics."
RETURN winner
```

The rel node can also locate all the text that demonstrates this relationship. There are many ways in which the original content in the corpus may be accessed, and questions can be answered using the knowledge graph we have constructed. We must convert the original text into a different format to perform most of these tasks. The inferred knowledge graph offers a second layer that makes it easier to traverse the distilled knowledge by identifying NEs and the interactions between them. In a text, a graph visually represents the most important ideas.

Semantic Network

Much information has been retrieved from the text and converted into knowledge ready to be utilized in our current knowledge graph. This purified information is discovered when more and more texts are parsed and processed into the inferred knowledge graph. This knowledge structure should be examined when it comes to creating new, enhanced services for end consumers. Many fascinating features, such as automated reasoning, may be added on top of a knowledge graph. Symbolic AI's knowledge representation and reasoning branch tries to develop computer systems that can reason (like humans) based on an interpretable machine representation of the subject of interest[38]. There are no physical objects, events, relationships, or other domain artifacts to represent in this computer model. Graphs with nodes representing concepts and arcs representing relationships between those concepts are standard representations for such knowledge systems. Through semantic networks, it's possible to "abstract from natural language, representing the knowledge that is captured in text in a form more appropriate for computers".[39] Concepts are used to represent nouns, while verb phrases are used to represent relationships in such a text. Let's revisit the previous example to see what I mean. In the illustration, a semantic network is depicted.[39] "Marie Curie, the famous scientist, received the Nobel Prize in Physics in 1903."

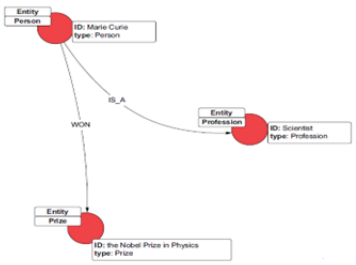


Figure 2.16: Nodes of KG

That's exactly how the inferred knowledge graph is structured. Keeping track of sources is essential if we want to understand why relationships are formed; thus, we make the relationships tangible. As a result, the inferred knowledge graph is a semantic network, although a reduced one, because, in our schema, we materialize the links to keep track of the source of each inferred relation. When it comes to extracting the semantic network from an inferred knowledge graph, we consider it a specific process that removes all the overhead associated with mapping relationships to their source and keeps only those concepts and relationships relevant, such as by considering how frequently they appear in the original corpus.

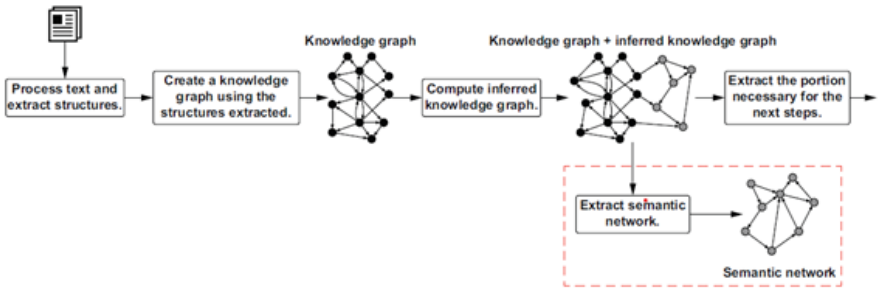


Figure 2.17: Knowledge graph semantic network

A semantic network's content is determined based on the concepts and relationships relevant to your field of study and your desired result. The inferred knowledge graph derived from the existing big graph is what we're using to construct our semantic network. As part of the extraction process, removing some of the relationship nodes and replacing them with more appropriate ones may be possible. Using the corpus, you have always been able to create a semantic network that meets all your requirements. Generic semantic networks are available to the general population. Many people utilize[40] ConceptNet5 as "a knowledge representation project, providing a vast semantic graph that describes generic human knowledge and how it is expressed in natural language" (Speer and Havasi, 2013) by its developers. Expert-created resources, crowdsourcing, and games are all included in the graph's collection of knowledge. Concept Net's ability to better understand the meanings of words people use can improve natural language applications.

2.7 Categories of Knowledge Graph

As of 2012, the Google Knowledge Graph had been announced "[A graph] that understands real-world items and their relationships to one another" is how the knowledge graph is described in the initial announcement. It wasn't long before "knowledge graphs" began to appear often in academic papers. In the words of Bergman Google's announcement of the "knowledge graph" was a watershed moment [19]. Although the statement was made informally, a technical description was not provided. Because of the increasing attention given to knowledge graphs in academic literature, the need for formal definitions arose to accurately characterize what they were, how they were built, how they could be used, and more generally to enable their study in a precise way. We can classify definitions into four general types.

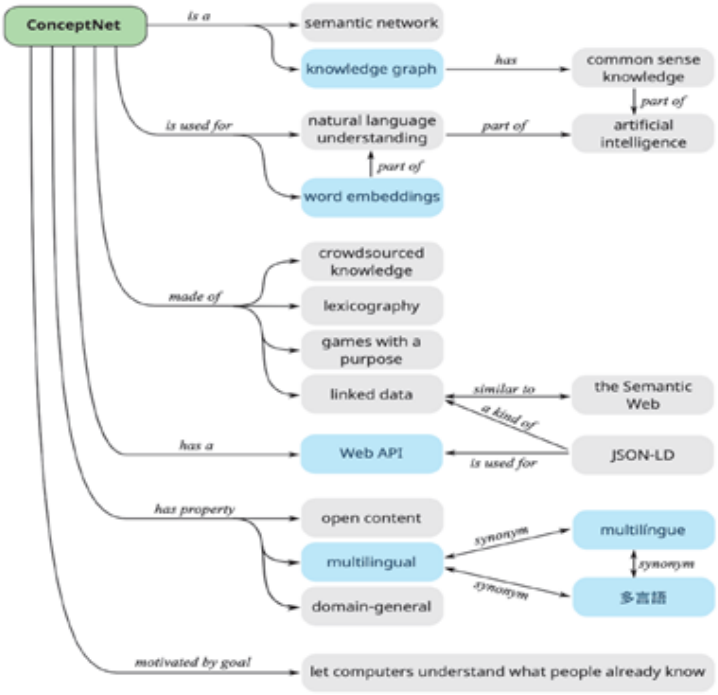


Figure 2.18: Categories of Knowledge Graph

2.7.1 Category 1

Knowledge graphs are graphs with nodes representing entities and edges representing relationships between those nodes. In many cases, it is assumed that a directed edge graph with named edges exists (analogously, a set of binary relations or triples).[41] As a short-hand way to describe the data structure upon which these embeddings work, some seminal articles on knowledge graph embeddings popularized this definition. Many subsequent studies on knowledge network embeddings have continued to adopt this definition, as shown in the survey by Wang et al. Despite its simplicity, there are some questions about the Category I definition: What is the difference between a graph (database) and a knowledge graph? What is the role of knowledge in this situation?

2.7.2 Category 2

One of the most prevalent definitions of this category is as follows: As far as we know[23], the first academic papers to use the term "knowledge graph" (a graph-structured knowledge base) (interestingly, a knowledge graph is defined analogously to a directed edge-labeled graph in the formal notation of these initial papers). So, what exactly is a "knowledge base" if this is

the case? The term "knowledge base" was first used in the context of rule-based expert systems in the 1970s (or earlier)[42], and later in the context of ontologies and other logical formalisms . Is it possible to maintain the original definitions while maintaining a graph-structured database (or any other type of structure, for that matter) without a logical formalism? On the other hand, the notion of a "knowledge base" has been a source of confusion in the past (KB). Notably, Brachman and Levesque (1986) argue that "if we ask what the KB teaches us about the world, we are asking about its Knowledge Level" in their report following a workshop on this topic.

2.7.3 Category 3

The third category of definitions outlines other technical features that a "knowledge graph" should meet. Paulheim lists four characteristics that characterize the knowledge graphs analyzed for the paper. Specifically, a knowledge graph "mainly describes real-world entities and their interrelations, organized in a graph; defines possible classes and relations of entities in a schema; allows for potentially interrelating arbitrary entities"; he thus rules out ontologies without instances (e.g., DOLCE) and word sense graphs (e.g., WordNet) as not meeting the first two criteria, while relational databases do. Ehrlinger and Wöß examine "knowledge graph" definitions, criticizing Category II definitions since knowledge graphs are not synonymous with ontologies38, and Google for calling its knowledge graph a "knowledge base". Ontology, knowledge graph, and knowledge base have all been defined before, however they offer the following definition: To generate new knowledge, a knowledge graph collects and organizes data into an ontology. The supply of reasoning capabilities distinguishes a knowledge graph from an ontology (considered synonymous with a knowledge base). a knowledge graph is "...a semi-structured data model characterized by three components: an extensional component, a set of relational constructs for schema and data (which can be effectively modeled as graphs or generalizations thereof); According to them, a knowledge graph is a knowledge base supplemented by reasoning, as defined by Ehrlinger and Wöß . Bergman provides a list of further meanings. While having a formal description for knowledge graphs helps to solidify their study, as Bergman points out, many of these definitions don't seem to fit current knowledge graph usage. For example, it is unclear which of these definitions Google Knowledge Graph would fit (if any). Moreover, many of the requirements given by such definitions contradict the vast literature on knowledge graph embeddings.

2.7.4 Category 4

Unlike the previous three categories, this one uses an extensional definition of knowledge graphs, defining them by example. Examples of knowledge graphs are DBpedia, Google's Knowledge Graph, Freebase, and YAGO. This category perhaps avoids the challenge of defining a knowledge graph rather than supplying one. "Many actual implementations place limitations on the interconnections in knowledge networks by establishing a schema or ontology," the paper notes. This allows diverse products and applications to share a common language and reuse definitions and descriptions created by others. Their short formal formulation allows developers to infer new facts and build knowledge". While ontologies and formal representations are not required for a knowledge graph, we view this concept as matching Category I.

Formal Definitions

Directed Edge-labeled Graph: A directed edge-labeled graph is a tuple $G = (V, E, L)$, where $V \subseteq Con$ is a set of nodes, $L \subseteq Con$ is a set of edge labels, and $E \subseteq V \times L \times V$ is a set of edges. Con is a Universal set.Con:Denotes the set of all possible nodes or vertices in the graph G

Set of Directed Labeled Edges: A set of directed labeled edges $E \subseteq V \times L \times V$ forms a directed edge-labeled graph (V, E, L), in which case we refer to the graph induced by E if V and L contain all and only those nodes and edge labels, respectively, used in E. Set operators can be applied to directed edge-labeled graphs; for example, given $G1 = (V1, E1, L1)$ and $G2 = (V2, E2, L2)$, $G1 \cup G2$ means the directed edge-labeled graph induced by $E1 \cup E2$ in $G2$.

Heterogeneous Graph: A heterogeneous graph is a tuple $G_B = (V, E, L, \ell)$, where $V \subseteq Con$ is a set of nodes, $L \subseteq Con$ is a set of edges and node labels, $E \subseteq V \times L \times V$ is a set of edges, and $\ell : V \to L$ maps each node to a label.

Property Graph: A property graph is a tuple $G_B = (V, E, L, P, U, e, \ell, p)$, where $V \subseteq Con$ is a set of node ids, $E \subseteq Con$ is a set of edge ids, $L \subseteq Con$ is a set of labels, $P \subseteq Con$ is a set of properties, $U \subseteq Con$ is a set of values, $e : E \to V \times V$ maps an edge id to a pair of node ids, $\ell : V \cup E \to 2L$ maps a node or edge id to a set of labels, and $p : V \cup E \to 2P \times U$ maps a node or edge id to a set of property-value pairs. P: Properties of nodes or edges , U:Values Associated with properties.

Graph Dataset: A named graph is a pair (n, G) where G is a graph, and $n \in Con$ is a graph name. A graph dataset is a pair $D = (G_D, N)$ where G_D is a directed edge-labeled graph called the default graph, and N is either the empty set or a set of named graphs $\{(n_1, G_1), \ldots, (n_k, G_k)\}$ $(k > 0)$ such

that $n_i = n_j$ if and only if $i = j$ $(1 \leqslant i \leqslant k, 1 \leqslant j \leqslant k)$.

RDF Dataset: In an RDF dataset, each graph is an RDF graph, and the graph names might be blank nodes or IRIs, as defined by the W3C.

Deductive Knowledge

Graph Interpretations: An interpretation for a graph is defined as a pair $I = (\Gamma, \cdot_I)$ where $\Gamma = (V_\Gamma, E_\Gamma, L_\Gamma)$ is a graph called the domain graph, and $\cdot_I : Con \rightarrow V_\Gamma \cup L_\Gamma$ is a partial mapping from constants to terms in the domain graph.

Graph Models: Let $G = (V, E, L)$ be a directed edge-labeled graph. An interpretation $I = (\Gamma, \cdot_I)$ satisfies G if $V \cup L \subseteq \text{dom}(\cdot_I)$, and for all $v \in V$, $v_I \in V_\Gamma$; for all $l \in L$, $l_I \in L_\Gamma$; and for all $(u, l, v) \in E$, $(u_I, l_I, v_I) \in E_\Gamma$. If I satisfies G, we call I a model of G.

Graph Parallel Frameworks

Directed Vector-labeled Graph: A directed vector-labeled graph $G = (V, E, F, \lambda)$ is defined, where V is a set of nodes, $E \subseteq V \times V$ is a set of edges, F is a set of feature vectors, and $\lambda : V \cup E \rightarrow F$ labels each node and edge with a feature vector.

Graph Parallel Framework: A graph parallel framework accepts a directed graph labeled with feature vectors as input.

Chapter 3

Automatic Knowledge Graph and Automatic Feature Engineering

This chapter is dedicated to extracting pertinent features through utilizing a unique conceptual framework, with a primary emphasis on harnessing the abundant information accessible on Wikipedia. The method of featured gathering is closely associated with the notion of scanning, in which we methodically traverse Wikipedia to acquire relevant features crucial to our research goals. [43]This methodological framework is a fundamental basis for amassing a wide range of characteristics encompassing all-encompassing insights. By employing crawling techniques, we can methodically navigate through Wikipedia, extracting significant features that cover a wide range of subjects and domains. The attributes sourced from the vast repository of Knowledge on Wikipedia are crucial in forming a resilient and enlightening knowledge graph. The knowledge graph depicts the interdependencies and correlations between the collected attributes, providing a comprehensive perspective of the information environment. As the discussion progresses through the following chapters, the knowledge graph's multifaceted applications gain greater clarity regarding its significance. The knowledge graph contains abstracted and organized features that form the basis for numerous advanced ideas and approaches that will be investigated in succeeding sections of our research. An aspect worthy of mention is the utilization of machine learning algorithms, in which the knowledge graph serves as an abundant repository of annotated and interrelated data points. This process aids in constructing predictive models, which empower the system to identify trends, patterns, and connections within the vast dataset. Furthermore, incorporating automated artificial intelligence (AutoML) enhances our investigation into the automation domain by optimizing the model-building and refinement procedure. The ramifications of the produced knowledge graph transcend the realm of machine learning and encompass a wide range of practical applications and

concepts. The extensive capabilities of this system provide opportunities for implementing natural language processing, semantic search, and semantic analysis. Enhancing comprehension and promoting innovation in various domains, such as information retrieval and knowledge representation, are facilitated by the complex network of connections that constitute the knowledge graph. This chapter establishes the foundation for comprehensively examining characteristics acquired via perusing Wikipedia. The forthcoming chapters will reveal the latent capabilities of these attributes, revealing profound understandings using machine learning, AutoML, which is an assortment of sophisticated methodologies. The knowledge graph proves to be an adaptable and dynamic resource, guiding our investigations in domains such as automation and analysis of semantics, among others.

3.1 Architecture of Automatic Feature Engineering

Knowledge is an essential part of generating the automatic knowledge graph. For our research, we use BTC crypto data for feature extending and for developing an automatic knowledge graph of the BTC result data. This can be done in different steps, as shown in the Figure 4.5.

A critical initial step in our process entails the establishment of a precise articulation of both business and expert requirements. Once the business and expertise requirements have been firmly established, the following phase entails evaluating and selecting datasets. The primary objective is to identify and select datasets that align with the project's goals. This may entail conducting a search of preexisting databases or commencing the accumulation of data. An evaluation, which examines the data for consistency, accuracy, and completeness while resolving any anomalies, is essential to this stage. Insights are revealed during the data analysis exploratory phase via statistical summaries or visual representations. After refining raw data, feature engineering generates appropriate variables for analysis. Thirdly, the integration of subject expert Knowledge is an essential component. This entails the integration of data-driven analyses with subject matter expertise through expert consultation. In collaboration with domain specialists, domain-specific feature selection prioritizes variables pertinent to the business context. By performing a thorough integration, the dataset is brought into conformity with the complexities of the field, thereby improving the comprehensibility of subsequent findings. Domain specialists are of utmost importance when validating and interpreting outcomes, guaranteeing that they align with practical requirements. The iterative process is a vital link between insights derived from data and decisions that can be implemented. Following this, as part of our

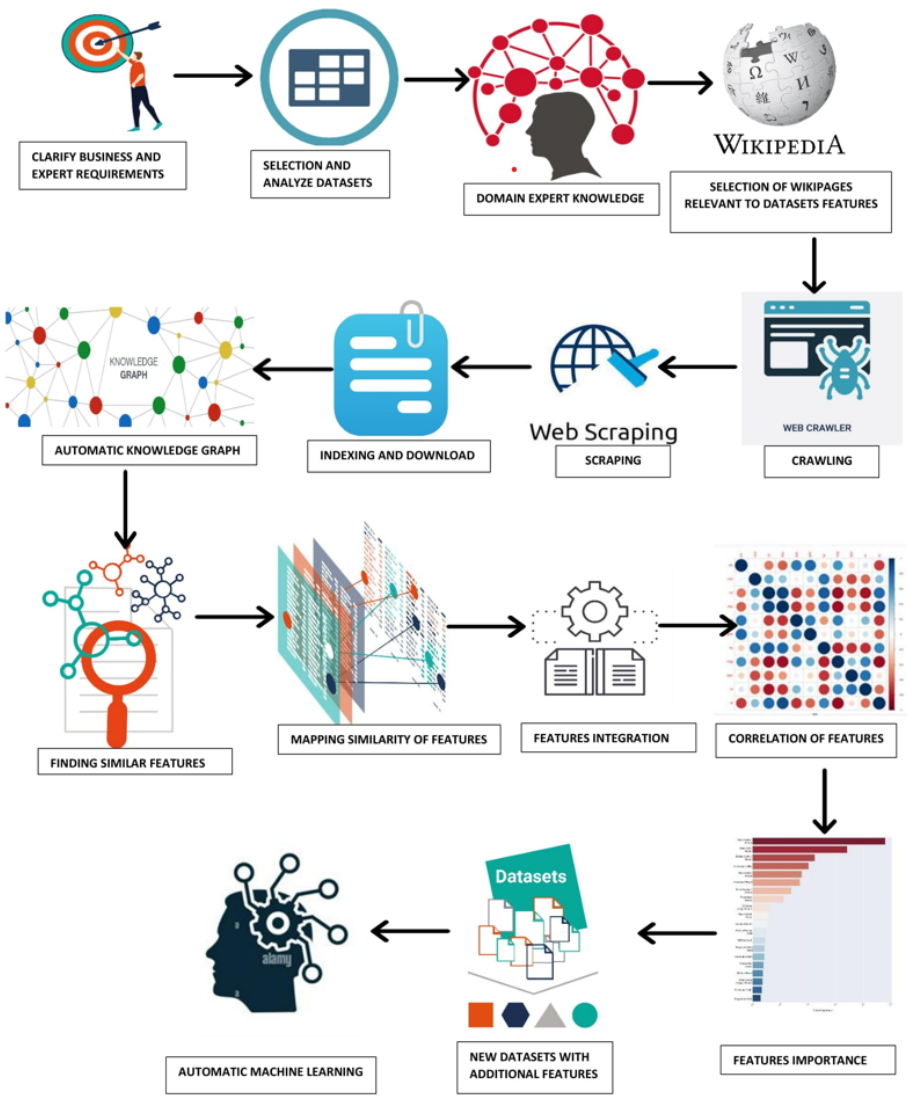


Figure 3.1: Automatic Feature Engineering

methodology, we identified and included pertinent attributes from Wikipedia pages. This stage entails utilizing the extensive collection of data accessible on Wikipedia to extract relevant characteristics that are important for the intended analysis. The selected attributes are paramount in influencing the knowledge graph and contributing to the subsequent phases of the undertaking. A hierarchical approach is utilized in the feature selection procedure, predicated on the distance term from the root node. The features are categorized into discrete levels within this hierarchical structure, where each level corresponds to a unique degree of closeness to the core node. Level 1, Level 2, and Level 3 are the three distinct levels. Level 1 consists of attributes that are in closest proximity to the root node, thereby signifying the aspects that are most instant and directly related. As one advances to Levels 2 and 3, the traits exhibit a progressive increase in distance from the root node, thereby incorporating more extensive and potentially abstract connections. The purpose of this multitiered structure is to capture the subtleties of proximity and differentiate characteristics that have more distant connections to the central node from those that are more intricately intertwined with it. Implementing a hierarchical structure enables a more intricate depiction of the interconnections among the elements in the dataset, providing a comprehensive viewpoint that corresponds to the different levels of significance.

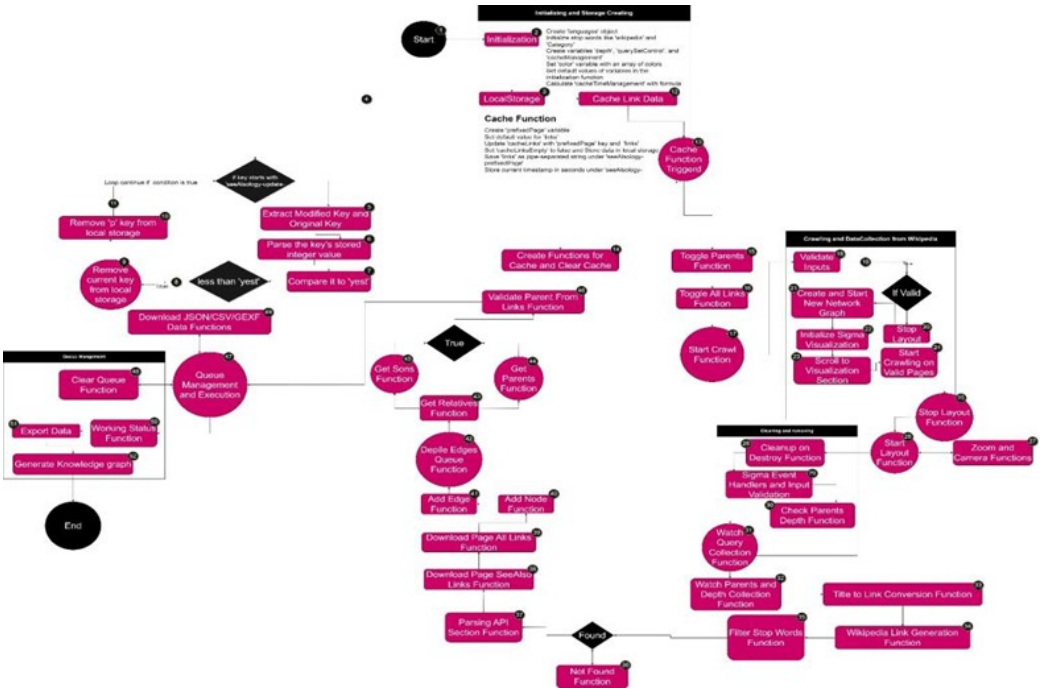


Figure 3.2: Automatic Knowledge Graph

Once the fundamental business and expert criteria are met, the critical stage of feature acquisition commences with the scouring of Wikipedia

pages. This pivotal stage is essential for acquiring the necessary characteristics that aid in the subsequent development of an understanding graph. The scanning mechanism methodically extracts pertinent data and features from Wikipedia pages to the project's defined objectives. An essential component of this crawling procedure entails extracting relevant data via the "See also" sections of Wikipedia articles. This segment generally comprises hyperlinks to associated subjects, serving as a valuable resource for identifying interrelated characteristics. The links extracted from this segment form the basis for constructing a graph that incorporates the interconnections among different attributes, thereby facilitating a holistic comprehension of their associations. During the crawling process, the collected data is employed for the immediate construction of graphs and for strategic storage in the cache and local storage. The proactive caching mechanism facilitates posterior processes, specifically indexing and downloading. Through the implementation of data storage, the system guarantees streamlined retrieval and application in subsequent stages, thereby refining the workflow. This algorithm demonstrates the working of the tool from the initial stage to the crawling step.

Algorithm 1: Automatic Knowledge Graph

Input : Functions and their descriptions
Output: None
Variables Initialization:

- **language:** Object
- **Stop words:** ["Wikipedia," "Category"]
- **Data management:** depth, querySetControl, cache management
- **color:** Array of colors
- **cacheTimeManagement:** Time management calculation

Crawling and Layout Functions

- Initiate crawling and layout setup
- Halt the current layout if running.
- Initialize new layout settings.

Event Handlers and Input Validation

- Manage click events and legend drawing.
- Validate input parameters and set stop words.

Utility and Conversion Functions

- Include link conversion and stop word filtering.

Management Functions

- Handle not found pages and retries.
- Parse API response and cache links.
- Retrieve and filter links from Wikipedia pages.

Graph Node and Edge Management

- Add nodes and edges to the graph.
- Process and add pending edges to the graph.

Relative Page Retrieval Functions

- Retrieve links for a page.

Validation and Queue Management

- Ensure parent links integrity.
- Periodically process edges queue and execute tasks.
- Stop the crawling process and clear the queue.

The readily available data that was previously recorded plays a crucial role in constructing the knowledge graph, which is an essential element of the automated knowledge graph procedure. Initially, this entails augmenting the functionalities of BTC cryptocurrency information by implementing crawling methodologies on Wikipedia pages to acquire insights about pertinent attributes. The features are subsequently extracted, yielding significant insights into potential attributes relevant to the data of the BTC cryptocurrency. This procedure improves our comprehension of the field and enables the recognition of crucial attributes necessary for a thorough examination. Initially, we acquired BTC data from Yahoo Finance to construct the knowledge graph. This data generally includes fundamental attributes such as high, open, and closed. By capitalizing on the Knowledge acquired from prior sources concerning pertinent characteristics, we enhance this dataset by including further attributes extracted from the previously mentioned compilation. By incorporating an extensive array of attributes, this procedure increases the dataset's comprehensiveness and suitability for examination in the framework of the knowledge graph.

	Date	Open	High	Low	Close	Adj Close	Volume
2436	2021-05-19	42944.976563	43546.117188	30681.496094	37002.441406	37002.441406	1.263581e+11
2437	2021-05-20	36753.667969	42462.984375	35050.617188	40782.738281	40782.738281	8.828194e+10
2438	2021-05-21	40596.949219	42172.171875	33616.453125	37304.691406	37304.691406	8.205162e+10
2439	2021-05-22	37371.031250	38831.054688	35383.683594	37536.632813	37536.632813	5.737727e+10
2440	2021-05-23	37026.246094	38265.753906	31227.339844	33877.511719	33877.511719	7.410413e+10

Figure 3.3: Automatic Feature Engineering

When these features are extended in the BTC dataset, The resulting knowledge graph generated from these features is shown in the figure below.

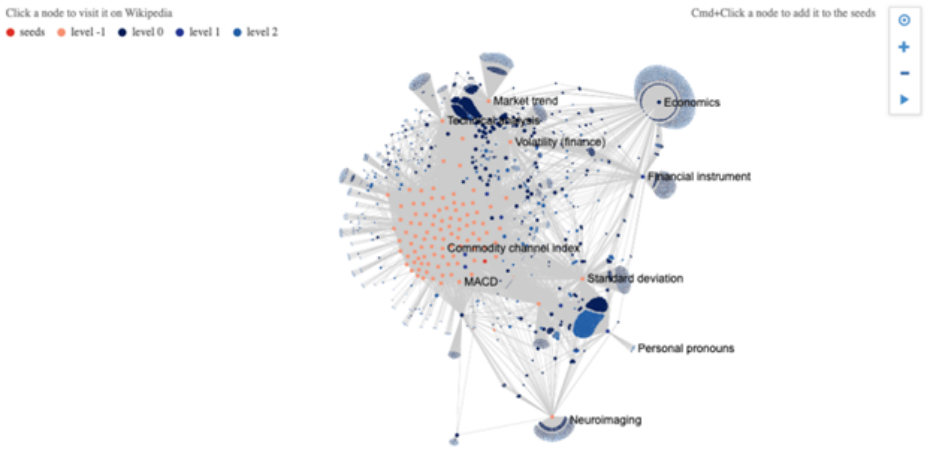


Figure 3.4: Automatic Knowledge Graph

The constructed knowledge graph incorporates every conceivable attribute pertinent to the BTC cryptocurrency. The hierarchical structure of this knowledge graph, which is predicated on the notion of levels, is an integral component. As previously explained, the profundity of the crawling process determines which level each individual reaches. Features near the seed node are denoted by Level 0, while subsequent levels indicate increasing distances from the seed. Anchoring the complete graph, the seed node functions as the principal root; each feature is classified into its corresponding level according to its proximity to the seed. Using hierarchical structuring in the knowledge graph enables a more comprehensive comprehension of the interconnections between features. Within the data visualization domain, the

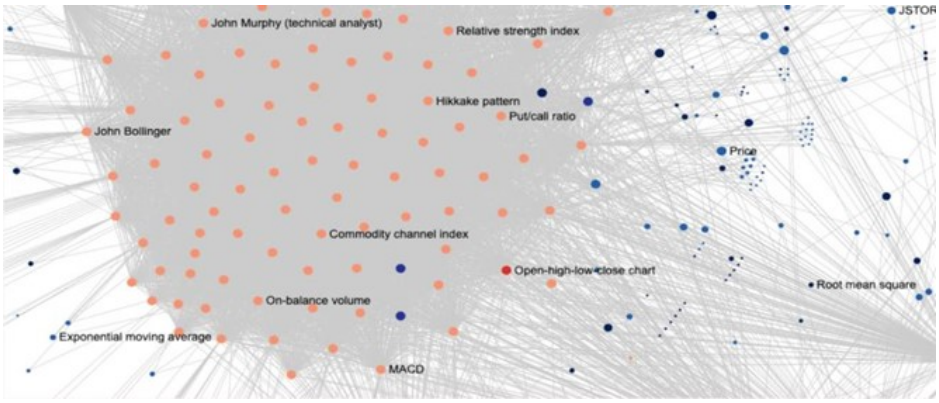
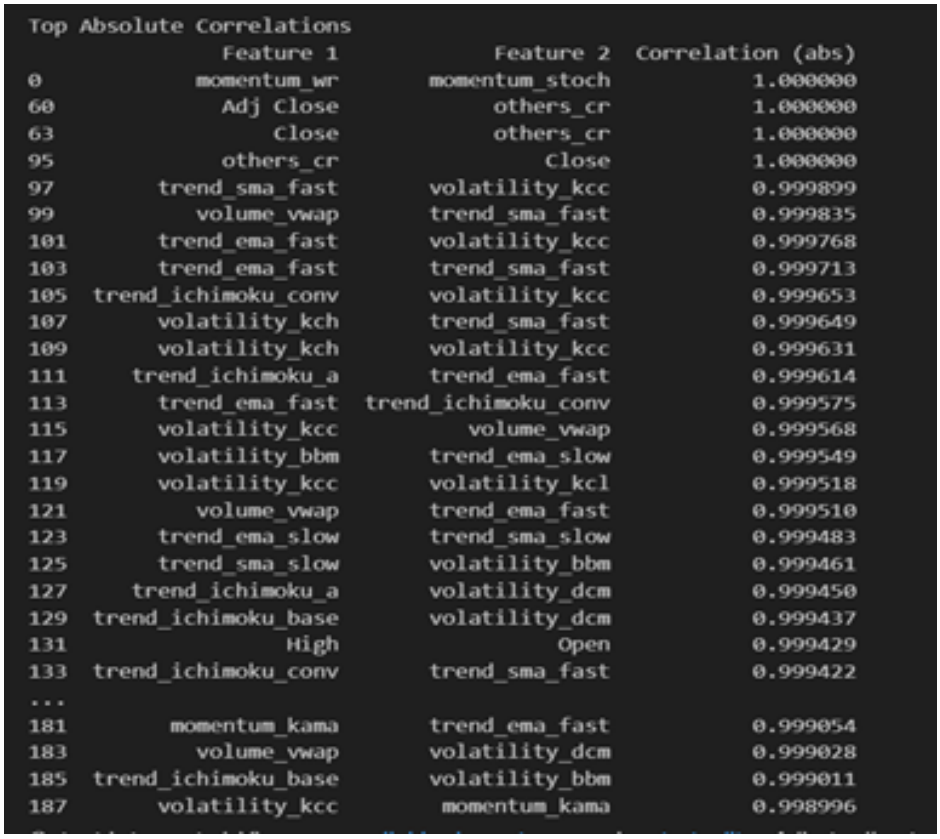


Figure 3.5: Automatic KG with Levels

knowledge graph functions as an all-encompassing repository of the characteristics of the BTC cryptocurrency, providing a systematic structure for comprehending its complex attributes. By employing precise color coding, the graph effectively distinguishes the fundamental characteristics from the supplementary ones while also communicating the hierarchical structure of their interconnections. This graphical depiction enables analysts to determine the proximity of attributes to the seed node, thereby facilitating the recognition of crucial components and peripheral connections. In addition to its visual attractiveness, the graph contains many practical insights that guide succeeding stages of data refinement and analysis. As the graph undergoes feature engineering and data purification, its importance surpasses a simple visual representation, transforming into a dynamic instrument for optimizing data. By systematically eliminating inaccurate data and consolidating similar attributes, this procedure enhances the precision and integrity of the dataset. Additionally, by strategically organizing features according to their significance, stakeholders can develop a more profound comprehension of the dynamics of Bitcoin. This empowers them to prioritize critical variables and draw meaningful conclusions. When combined with feature

engineering, generating a heatmap is pivotal in the analytical procedure. By integrating statistical methods and Python libraries, this visual representation unveils the interconnectedness of Bitcoin attributes, thereby unveiling correlations and patterns that may elude conventional analysis. The heatmap furnishes actionable insights to decision-makers, illuminating the fundamental dynamics of the cryptocurrency environment. Therefore, utilizing feature



Top Absolute Correlations

	Feature 1	Feature 2	Correlation (abs)
0	momentum_wr	momentum_stoch	1.000000
60	Adj Close	others_cr	1.000000
63	Close	others_cr	1.000000
95	others_cr	Close	1.000000
97	trend_sma_fast	volatility_kcc	0.999899
99	volume_vwap	trend_sma_fast	0.999835
101	trend_ema_fast	volatility_kcc	0.999768
103	trend_ema_fast	trend_sma_fast	0.999713
105	trend_ichimoku_conv	volatility_kcc	0.999653
107	volatility_kch	trend_sma_fast	0.999649
109	volatility_kch	volatility_kcc	0.999631
111	trend_ichimoku_a	trend_ema_fast	0.999614
113	trend_ema_fast	trend_ichimoku_conv	0.999575
115	volatility_kcc	volume_vwap	0.999568
117	volatility_bbm	trend_ema_slow	0.999549
119	volatility_kcc	volatility_kcl	0.999518
121	volume_vwap	trend_ema_fast	0.999510
123	trend_ema_slow	trend_sma_slow	0.999483
125	trend_sma_slow	volatility_bbm	0.999461
127	trend_ichimoku_a	volatility_dcm	0.999450
129	trend_ichimoku_base	volatility_dcm	0.999437
131	High	Open	0.999429
133	trend_ichimoku_conv	trend_sma_fast	0.999422
...			
181	momentum_kama	trend_ema_fast	0.999054
183	volume_vwap	volatility_dcm	0.999028
185	trend_ichimoku_base	volatility_bbm	0.999011
187	volatility_kcc	momentum_kama	0.998996

Figure 3.6: Features

engineering, data purification, and heatmap analysis to enhance the knowledge graph becomes a fundamental component in the data-driven investigation of the Bitcoin cryptocurrency. In addition to its visual assistance, it serves as a conduit for actionable intelligence, enabling strategic planning and well-informed decision-making within a constantly evolving digital environment. As a result, risk management protocols and strategic initiatives are informed by these insights. Feature selection aims to identify the most pertinent characteristics for modeling purposes while removing redundant or superfluous elements. Feature selection can be accomplished according to various criteria, including feature importance, recursive feature elimination, and correlation-based feature selection. The criteria employed are established in accordance with the research's specific objectives and the dataset's

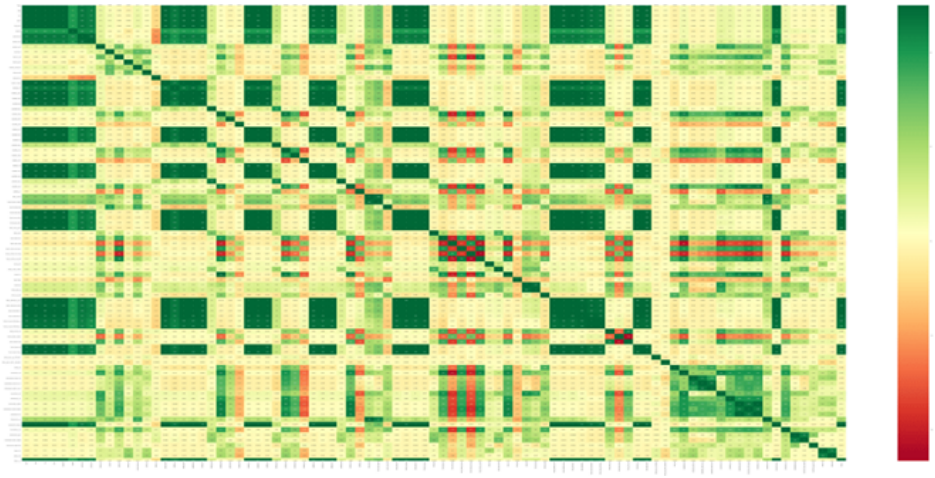


Figure 3.7: HeatMap of Features

attributes. The RandomForestClassifier is employed to ascertain the significance of individual attributes. The purpose of this classifier is to evaluate the importance of features. High-quality features have a greater impact on generating accurate forecasts. Feature importance scores in decision tree divides are frequently determined by how much a feature reduces an impurity, such as the Gini impurity. Utilizing this visualization facilitates the feature selection process to incorporate them into the ultimate prediction model. It discloses which characteristics exert the most significant impact on Bitcoin price predictions. Additional methods of feature selection may be implemented to enhance the feature set. Utilizing methodologies such as recursive feature removal and establishing feature relevance thresholds empowers the exploration of diverse feature subsets. This enables one to evaluate the features' influence on the model's performance and improve the approach taken for selecting the features. The dataset, consisting of a comprehensive assortment of data concerning the attributes of the Bitcoin cryptocurrency, is a foundational resource for subsequent analyses and discussions. Through meticulous scrutiny, we have managed to condense this extensive compilation into a specific subset comprising 54 Features that are paramount and near the foundations of Bitcoin dynamics. This meticulously curated collection represents a deliberate aggregation of information strategically positioned to produce actionable intelligence and offer direction for tactical undertakings. This is the list of 54 extracted features using the feature importance above. After being identified, these crucial characteristics are extracted and stored in a fresh dataset. This new dataset is then formatted and saved as a CSV file to facilitate their seamless integration into subsequent phases of analysis and modeling. By reducing the size of the dataset, computational efficacy is not only improved but relevant insights are also preserved for subsequent

Table 3.1: Feature Description

Feature	**Feature**
Date	trend_adx_pos
Open	trend_adx_neg
High	trend_vortex_ind_pos
Low	trend_vortex_ind_neg
Close	trend_vortex_ind_diff
Adj Close	trend_trix
Volume	trend_mass_index
volume_adi	trend_cci
volume_cmf	trend_dpo
volume_fi	trend_kst
volume_mfi	trend_kst_sig
volume_em	trend_kst_diff
volume_sma_em	trend_visual_ichimoku_a
volume_vpt	trend_stc
volume_nvi	momentum_rsi
volatility_atr	momentum_stoch_rsi_k
volatility_bbw	momentum_stoch_rsi_d
volatility_bbp	momentum_tsi
volatility_kcw	momentum_uo
volatility_kcp	momentum_stoch
volatility_dcw	momentum_stoch_signal
volatility_dcp	momentum_wr
volatility_ui	momentum_ao
trend_macd	momentum_roc
trend_macd_signal	momentum_ppo
trend_macd_diff	momentum_ppo_signal
trend_adx	momentum_ppo_hist

analysis and reference. As we enter Chapter 3, which serves as a foundation

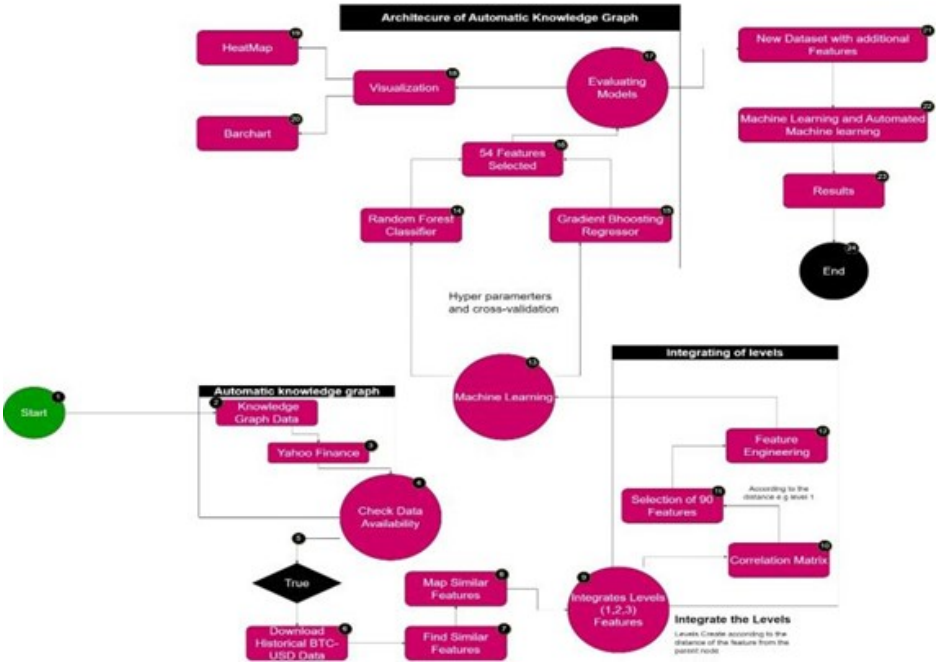


Figure 3.8: Architecture Diagram

for traditional and automated machine learning approaches, the enriched dataset is well-positioned to provide resources for exploratory analysis, algorithmic optimization, and predictive modeling. This enhanced dataset allows interested parties to utilize machine learning methodologies, including automated model selection and sophisticated algorithmic configurations. This fosters innovation and enables the disclosure of more profound insights into the intricacies of the Bitcoin cryptocurrency.

Chapter 4

Automated Feature Engineering: A Classification of Strategies from a Systematic Literature Review

In recent years, there has been an incredible boost in adopting machine learning, and more in general artificial intelligence approaches, in many different contexts. The need for ML and A.I. experts has rapidly increased so that the demand cannot currently cope with the offer, and many positions in such a domain still need to be assigned. ML opens new horizons for research both in industry and academia. It is constantly evolving, bringing challenges as well as getting more innovative. Concurrently, the ML community has put much research into trying to equip experts with more powerful and automatic tools that support the typical steps of an ML pipeline. The goal is to provide users with increased capability in determining which approach best suits a specific data set. These efforts resulted in the establishment of a new branch named AutoML, which aims to study and provide tools to automatize ML activities as much as possible to reduce the need for human intervention or, better yet, to provide more powerful tools.

To alleviate and make the data expert activities more focused, Such an effort has produced clear results and state-of-the-art tools, such as AutoSciKit Learn [44] and T.P.O.T. [45], to cite a few, are clear and relevant examples. Nevertheless, among the various steps constituting a typical ML pipeline, the feature engineering step manifested a different level of advancement concerning the other phases, and the activity is still mainly left to the data scientist with little availability of automatic support. This paper intends to investigate the proposed solutions for automatic support for feature engineering [46] activity. This activity has the strictest dependencies on the application domain, for which knowledge and human intuition can be invaluable tools to achieve satisfactory results. Nevertheless, automatic mechanisms can help

the data scientist easily explore possible solutions. The investigation has been carried out, taking advantage of available digital libraries, and it has been structured according to a Systematic Literature Review (S.L.R.) protocol, as suggested in [47], to avoid any bias due to our previous knowledge of the domain. In this study, we will examine the solutions that have been proposed to help data scientists with feature engineering. Many different proposals are today available in the domain of feature engineering. This paper explored different automatic feature selection approaches in machine learning. To identify such a proposal, we conducted a systematic literature review that resulted in the final identification of 63 papers proposed from 2000 to 2021 concerning such a topic. The study of such research has led us to the definition of a classification framework that tries to cluster the different approaches according to the strategy they adopt to support the activity. The survey enabled us to identify 1432 papers retrieved from the selected digital library and then apply the steps foreseen by the chosen S.L.R. protocol to restrict to 68 papers. After carefully reading the selected papers, we defined a framework to cluster such approaches, and we identified eight different clusters that can be used to classify the works published on this topic. The rest of the paper is organized as follows. Section 2 discusses the methodology and protocol we adopted to run the S.L.R., reducing, at minimum, any influence from our previous knowledge of the domain. Section 3 discusses the results of the review we conducted just from a numerical point of view, while Section 4 discusses the proposed classification. Successively, we close the paper, touching upon possible future work directions.

4.1 Methodology

A systematic literature review (S.L.R.) is an unbiased method to find, evaluate, and interpret research relevant to a specific topic. In this study, we adapt the idea and the methodology from Kitchenham et al.[47], which presents an articulated and controlled protocol for performing S.L.R. Even though the methodology has been mainly conceived to run S.L.R. in software engineering, we consider this method conducive for conducting S.L.R. in our research topic, as the method has been adopted in many other domains. Kitchenham's methodology consists of three phases: ing, conducting, and reporting. In the following, we describe each step as it has been performed in our research activity.

4.1.1 Planning the S.L.R

This section outlines the stages of planning the S.L.R., such as defining research questions and selecting a search strategy. The choice of perform-

ing an S.L.R. is mainly due to Because the ML domain is nowadays quite widespread, possible contributions could come from many different sources that would have been difficult for us to identify. Running a literature review without adopting a precise and mechanical process would have resulted in identifying a restricted number of works mainly related to our communities of research. On the other hand, any S.L.R. protocol has its pitfalls, so we tried to carefully check the results we obtained. Therefore, to run our S.L.R., we had to clarify our objectives precisely regarding research questions. After several iterations in which we tried to synthesize and focus our objectives, we ended up with the identification of the following three questions:

4.1.2 Research Questions

R.Q.s are the most essential component of an S.L.R. Authors collaborate to develop research questions during brainstorming sessions. According to our objectives, we ended up with the identification of the following three questions:

- **RQ1:** What are the proposed approaches for automatic feature engineering?
- **RQ2:** What are the theoretical and technical bases of the proposals supporting the automatic exploitation of feature engineering?
- **RQ3:** Are there ideally suitable tools and libraries for feature engineering?

The first question concerns discovering available methods for automatic feature engineering (AutoFE). This emerges as a new domain after the invention of AutoML and constitutes a relevant component. The second question instead refers to identifying classes of suitable methods to perform feature engineering automatically. The focus here is more on the theoretical and technical basis of the identified approaches to derive a possible classification for the proposed approaches. Finally, the last question aims to define the best suitable tools and libraries available to manipulate Data and apply feature engineering techniques for ML according to the different domains.

4.1.3 Search Strategy

This step involves the technique used to acquire relevant research papers from the literature, particularly emphasizing the definition of the search query. The selection of digital libraries to retrieve relevant research work is also critical. Another relevant step refers to defining the criteria for including and excluding research items from the survey and, finally, the performance

of the snowballing techniques. After an intense literature review and discussion with authors and domain experts, keywords are defined and filtered according to our research works. This is an iterative procedure to select the optimal search string, which has to be validated repeatedly by applying it over a digital library to attain quality and relevant results. Our queries apply to three primary areas: ”automatic feature selection,” ”automatic Feature Engineering,” and "automatic Machine Learning”. We identified appropriate terms to include in our query based on those domains. The words for the search query were devised and discussed among the paper's authors during a brainstorming session. The authors passed through several iterations before agreeing on the final set of terms mentioned in the following. Referring to the area of Automatic feature engineering, we devised the terms: ”Automated Feature”, "Matic feature extraction”, AutoML” ”Auto Data Science”, ”Deep Features synthesis”, and ”Data Science Machine" These keywords are frequently used in the most cited or state of

Art papers relevant to Automatic feature engineering. We included "T.P.O.T." and ”AutoSklearn." "auto-learn" and "explore kit" are famous tools and libraries that are widely used for Automatic feature engineering in AutoML. Finally, we constructed a search string to join terms of the same area with the Boolean operation OR.

Digital Libraries

We used the search string to query the digital libraries. The query string is applied on Scopus; Scopus was chosen since it is the most reputable library in the research fraternity. We did not explore publications from other large digital libraries, such as IEEE and A.C.M., because Scopus also indexes them. The query was restricted to the title, abstract, and keywords. We need all the numbers; 1432 documents were obtained. The papers are published in journals and conference proceedings. The publication period is restricted to 2000 to 2021 in the query string. We must justify the query's terms, particularly the years considered (we reasonably think to retrieve important contributions via snowballing). Remove duplicates once we've clustered by year of publication.

Inclusion/Exclusion Criteria

A set of inclusion/exclusion criteria has been developed as part of the protocol implementation to ensure the selection of only relevant research publications. We examine two inclusion and two exclusion criteria, which are given below. IC.1 Criterion states the research work is a primary study (surveys are excluded)". We include only new, unique literature. IC.2 decreasing function for citations (paper in 2001 with at least ten citations and till 2021 zero cita-

tions). E.C.1 The criterion states that the research work needs to be written in English. E.C.2 On the topic - papers reporting the usage of ML in specific domains are not considered. Application papers using ML are excluded.

Conducting

This phase refers to searching and analyzing the research. The search query is applied over the specified digital libraries defined in the review protocol. After the search is completed, the research works are analyzed, and only the research works relevant to the S.L.R. topic are chosen based on the inclusion/exclusion criteria. The refinement is a rigorous, quite complicated and critical task because the quality of research is based on selected papers; the final results may deviate if refinement is compromised. All the checks are applied to search work to ensure the quality of the research to be included without any biases. Finally, data is extracted from relevant research works, ensuring not only paper classification but also the collection of data needed to answer the research questions. Reporting: This final phase consists of writing up the review's findings; it effectively deals with replies to research questions.

4.2 Conducting the S.L.R.

This section outlines how we conducted the review. Specifically, we provided the details of each phase we covered to conclude the S.L.R. The phases are named as identification, screening, eligibility and inclusion. The final result of all phases was concluded in 63 papers. Potentially, 1432 relevant records were identified after applying the initial search query on Scopus. We found 124 articles as duplicate items identified in different digital libraries. There were 1308 articles remaining. After duplication, 802 articles were screened based on title and abstract. Five hundred six articles were qualified based on set eligibility criteria.450 full-text articles They were rejected due to missing data and the need to be relevant to the specific area of feature engineering. research articles were included in the study in the inclusion phase. Backward snowballing: We consider the related work section of the resulting paper set to identify additional papers. Forward snowballing: We access the digital library on each paper identified to see which papers refer to this one. The final result of papers shortlisted after applying the whole procedure is 63. These 63 papers are classified into 8 clusters. We applied snowballing techniques, backward snowballing, and forward snowballing on the final selection set. The aim was to improve the accuracy of the review by applying these strategies. Reviewing citations the articles identified in the previous step to determine relevant works not already identified with the

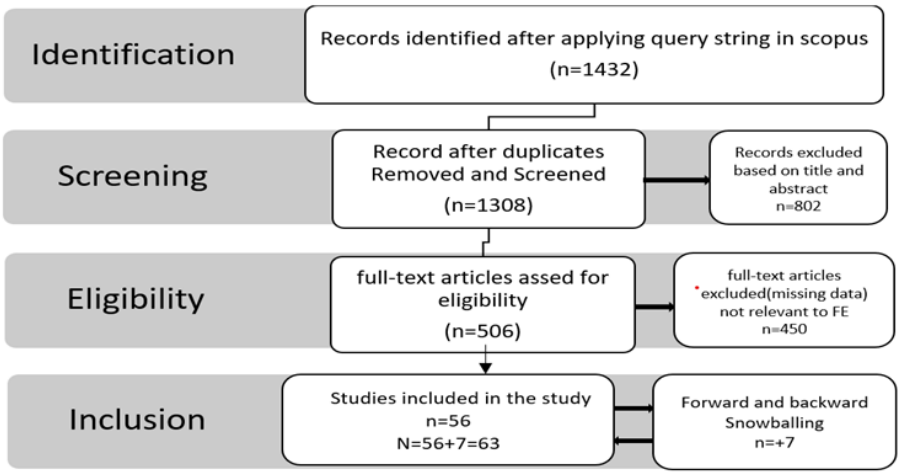


Figure 4.1: stages of SLR

search query or contributions antecedent 1999 is the snowballing backward method; the snowballing forward method requires digital libraries to identify articles citing the relevant works identified in the previous step is the snowballing forward method. The snowballing technique can be used numerous times, which means it was reapplied to the relevant papers found during the prior iteration of snowballing. We uncovered seven fresh research works in the first iteration of the snowballing (5 backward and two forward); we halted at the second iteration because no additional relevant publication had been located.

4.2.1 Resulting Query String

TITLE-ABS-KEY (”automatic feature selection” OR ”automatic Feature Engineering” OR ”automatic Machine Learning” OR ”AutoML” OR ”Auto Data Science” OR "Deep Features synthesis" OR "Data Science Machine" OR "T.P.O.T." OR "AutoSklearn” OR auto learn OR explore kit) OR TITLE-ABS-KEY ("Automated Feature" OR "automatic feature extraction”) AND DOCTYPE (ar OR cp) AND PUBYEAR $>$ 2000 AND PUBYEAR $<$ 2022 AND (LIMIT-TO (S.R.C.T.Y.P.E., "j") OR LIMIT-TO

(S.R.C.T.Y.P.E., "p")) AND (LIMIT-TO (DOCTYPE, "cp") OR LIMIT-TO (DOC- TYPE, "ar")) AND (LIMIT-TO (SUBJAREA, "COMP") OR LIMIT-TO (S.U.B.- J.A.R.E.A., "E.N.G.I.") OR LIMIT-TO (SUBJAREA, "MATH")) AND (LIMIT-TO (LAN- G.U.A.G.E., "English”)) AND (EXCLUDE (EXACTKEYWORD, "Image Segmentation”) OR EXCLUDE (EXACTKEYWORD, "Image Analysis”) OR EXCLUDE

(E.X.A.C.- TKEYWORD, "Image Processing") OR EXCLUDE (EXACTKEYWORD, "Signal Processing") OR EXCLUDE (EXACTKEYWORD, "Remote Sensing") OR EXCLUDE (EXACTKEYWORD, "Computer Vision") OR EXCLUDE (EXACTKEYWORD, "Speech Recognition") OR EXCLUDE (EXACTKEYWORD, "Brain") OR EXCLUDE (EX- ACTKEYWORD, "Biomedical Signal Processing") OR EXCLUDE (EXACTKEYWORD, "Bioinformatics") OR EXCLUDE (EXACTKEYWORD, "Image Enhancement") OR EXCLUDE (EXACTKEYWORD, "Three Dimensional Computer Graphics") OR EXCLUDE (EXACTKEYWORD, "Forestry") OR EXCLUDE (EXACTKEY- WORD, "Medical Imaging") OR EXCLUDE (EXACTKEYWORD, "Computer Simulation") OR EXCLUDE (EXACTKEYWORD, "Fuzzy Systems") OR EXCLUDE (E.X.A.C.- TKEYWORD, "Electroencephalography") OR EXCLUDE (EXACTKEYWORD, "Face Recognition") OR EXCLUDE (EXACTKEYWORD, "Fuzzy Clustering") OR EX- C.L.U.D.E. (EXACTKEYWORD, "Signal Encoding") OR EXCLUDE (EXACTKEY- WORD, "Computer Aided Design") OR EXCLUDE (EXACTKEYWORD, "Fault Diagnosis") OR EXCLUDE (EXACTKEYWORD, "Diseases"))

4.3 Results of the Automatic Search

In this section, we presented the quantitative findings of our study. We found 1432 publications published between 2001 and 2021: 588 in journals and 844 in the conference proceedings, as shown in the pie-chart inset in Figure 2). The graph in Figure 2 shows the trend of the publications in this field, and it is worth noticing that it can be divided into two parts: the first one, representing the years between 2001 and 2014, shows an increasing linear trend; on the contrary, the second one, between 2015 and 2021, presents an exponential trend, see the lower inset in Figure 2. Moreover, in the first part of the graph, i.e. the linear one, the average number of papers is 23.57 against an average of 157.42 for the exponential part. This shows that, in the last seven years, the number of published papers in the field had an average increase of 567.88After applying exclusion criteria and snowballing, we got 63 eligible papers. Figure 3 shows the trend of the eligible papers: also, in this case, the linear and exponential increasing trends are peculiar, but 2020-2021, a sudden decrease is recorded. The average number of published papers in the field between 2001 and 2014 was 1.5, while 6.71 papers appeared, on average, from 2015 to 2021; the average increase is 347.33As shown in Figure 4, the eligible papers (n=63) have been divided into 8 clusters according to the procedure used for the automatic feature selection (A.F.S.) :

- A.F.S. based on Machine Learning (ML)

Cluster	Papers	Approaches
1.	[48, 49, 50, 51, 52, 45, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66]	AFS based on ML
2.	[67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77]	AFS based on DL and ANN
3.	[78, 79]	AFS based on DR
4.	[80, 81, 82, 83]	AFS based on Relation Database
5.	[84, 85, 86, 87, 88]	AFS based on Knowledge Graph
5.	[89, 65, 90, 91, 92, 93, 94, 95]	AFS based on Prob and Stat
7.	[96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106]	AFS based on Hybrid Methods
8.	[107, 108]	AFS based on RL

Table 4.1: Cluster Papers

- A.F.S. based on Artificial Neural Network (ANN) or Deep Learning (DL)
- A.F.S. based on Dimensional Reduction of data
- A.F.S. based on Relational Databases
- A.F.S. based on Knowledge Graph
- A.F.S. based on Probabilistic and Statistical Methods
- A.F.S. based on Hybrid or Other Methods
- A.F.S. based on Reinforcement Learning

In Table 1, the reader can find the list of papers belonging to each cluster. It's worth noticing that the three most crowded clusters regarding ML, DL/ANN and Hybrid contain 20,11 and 11 papers, respectively. Those groups are followed by a fourth cluster, based on Probabilistic and Statistical Methods, that contains eight papers. All the papers where the authors developed approaches that can not be considered part of the other clusters were grouped in that cluster.

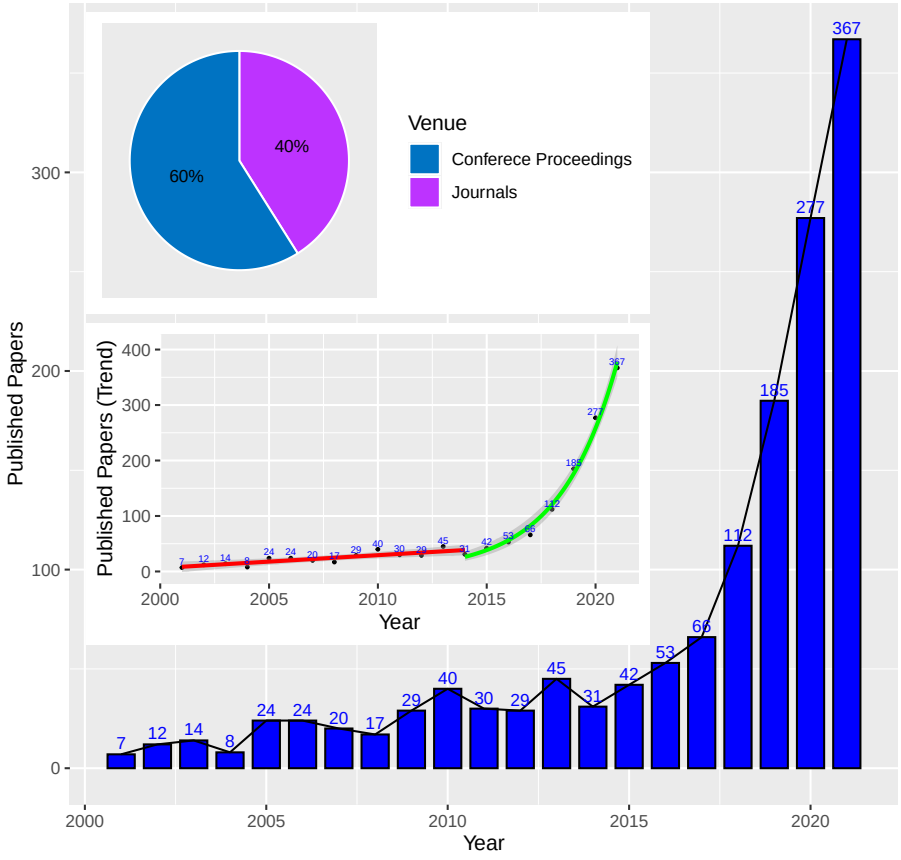


Figure 4.2: yearwise distrubution of papers

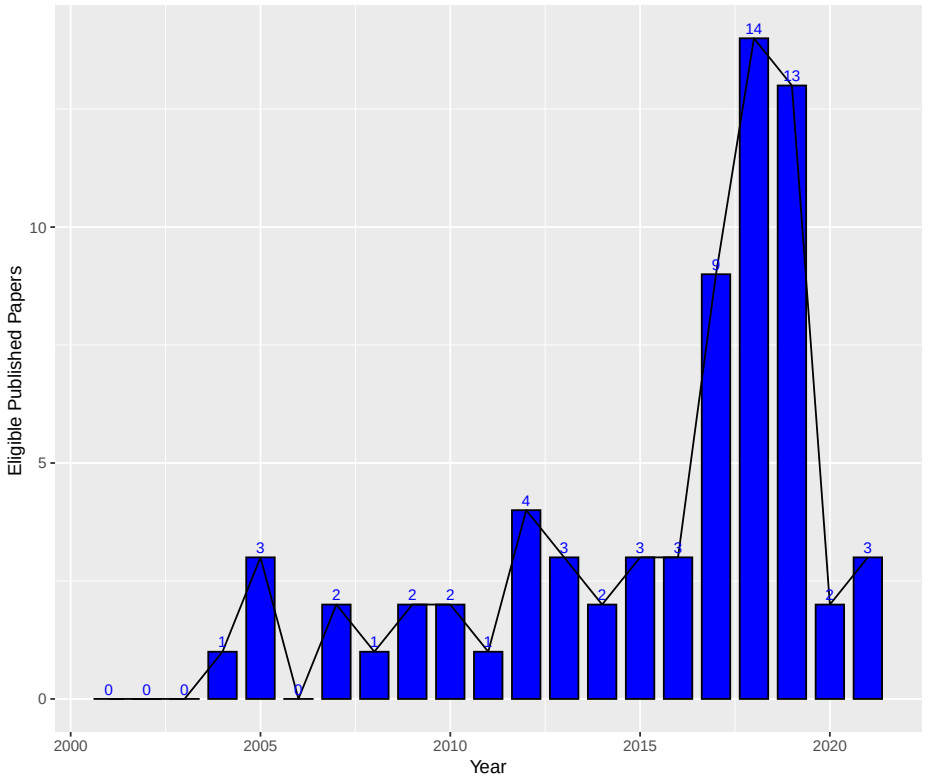


Figure 4.3: selection of eligible papers

4.4 Classification of Auto Feature Engineering approaches

In this section, we present an overview of the papers that we assign to each cluster. It would be useful to understand the hypothesis made by each approach. As shown in Figure 5, Feature selection techniques have always been divided into three categories:

Filter methods - They use a proxy measure to score a feature subset instead of minimizing the error rate. This measure, computed in preprocessing step, is chosen to be fast to compute as well as while still capturing usefulness of feature set. According to this method, features are ranked on the basis of apriori criteria; among the others, we recall 2, mutual information, Fisher discriminant criterion, and the Kruskall-Wallis test. Since filter methods are implemented in the preprocessing phase, they do not consider the impact on the classification algorithm. **Wrapper methods** - Wrapper methods use the classification algorithm as a black box; they use a predictive model to score feature subsets. Thus, the obtained subset is optimized for the particular classification algorithm [80]. Since wrapper methods train a new model for each subset, they are very computationally expensive but usually provide the best-performing feature set for that particular model type. **Embedded methods** - They are quite similar to wrapper methods since they are also used to optimize a learning algorithm's objective function or performance. The difference is that embedded methods use a metric that acts during learning. The LASSO method, as well as the Random Forest and the Recursive Feature Elimination (R.F.E.), are embedded methods. For a more exhaustive revision, please refer to [17]. Wrappers are computationally expensive, while Filters, on the contrary, are usually not so computationally expensive, but they produce a feature set that is more general than the set from a wrapper. In this review, we propose another taxonomy based on the technique used for performing A.F.S.

4.5 Automatic feature selection based on Machine Learning Approaches

According to our findings, techniques based on machine learning are widely utilized for automatic feature selection. We found 20 publications that use ML for Feature Selection. Among those articles, we can identify three main ML algorithms (G.A.,kNN and SVM) employed as a base for implementing anomaly detection techniques and others that use Miscellaneous techniques depending on the applicative domain.

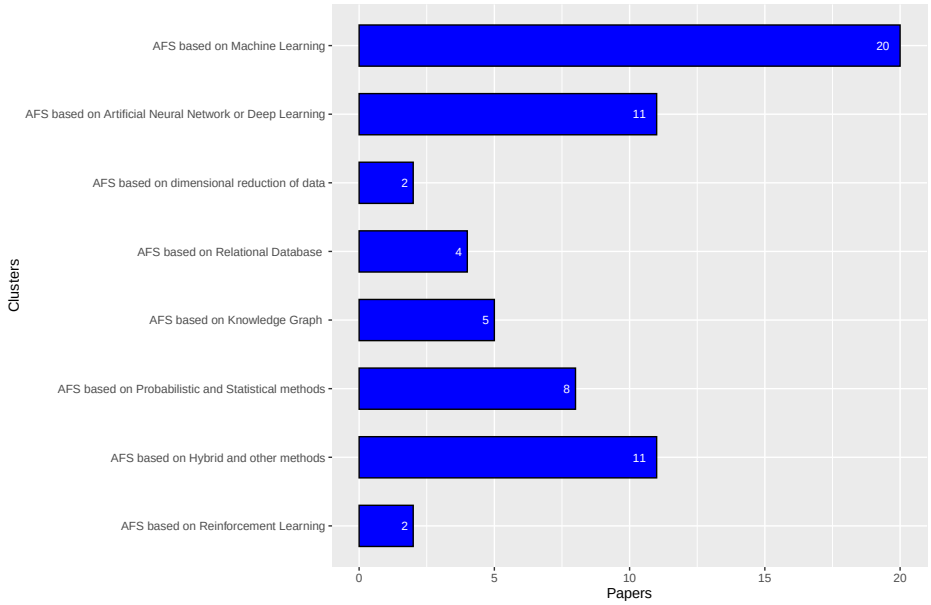


Figure 4.4: classification of clusters

4.5.1 Genetic-based algorithms

Genetic algorithms (G.A.s) are adaptive and are used to determine the optimal function combinations to solve a global optimization problem with selecting the best subset of features and rejecting the low-quality ones. In this study, [48] the authors investigated using a genetic algorithm for automatic feature selection in N.E.R. (Named Entity Recognition). G.A. is commonly used in feature learning in terms of computational time and criteria for evaluating an individual. Two machine learning algorithms are used to measure the precision of the N.E.R. method: K.N.N. and Conditional Random Fields. The suggested solution, as compared to a hill-climbing algorithm and a backward one, selects optimal features and produces promising results. In this method, population and data size may change during the learning process, minimize the computational cost, and avoid repeated fitness tests.

In recent years, genetic programming (G.P.) has gained much attention for obtaining optimal features for Regression and classification. Representation learning is the process of automatically engineering functions in feature spaces. The Feature Engineering Automation Tool (FEAT) is based on multidimensional genetic programming(M.G.P.).Framework is proposed in this work [49]. This framework is used in G.P. for feature creation and representation learning, as well as techniques that use machine learning as a heuristic for recognizing building blocks. This approach deals with a new multi-objective method and new semantic crossover operators. Educational Data Mining (E.D.M.) is commonly utilized in education to predict student achievement.

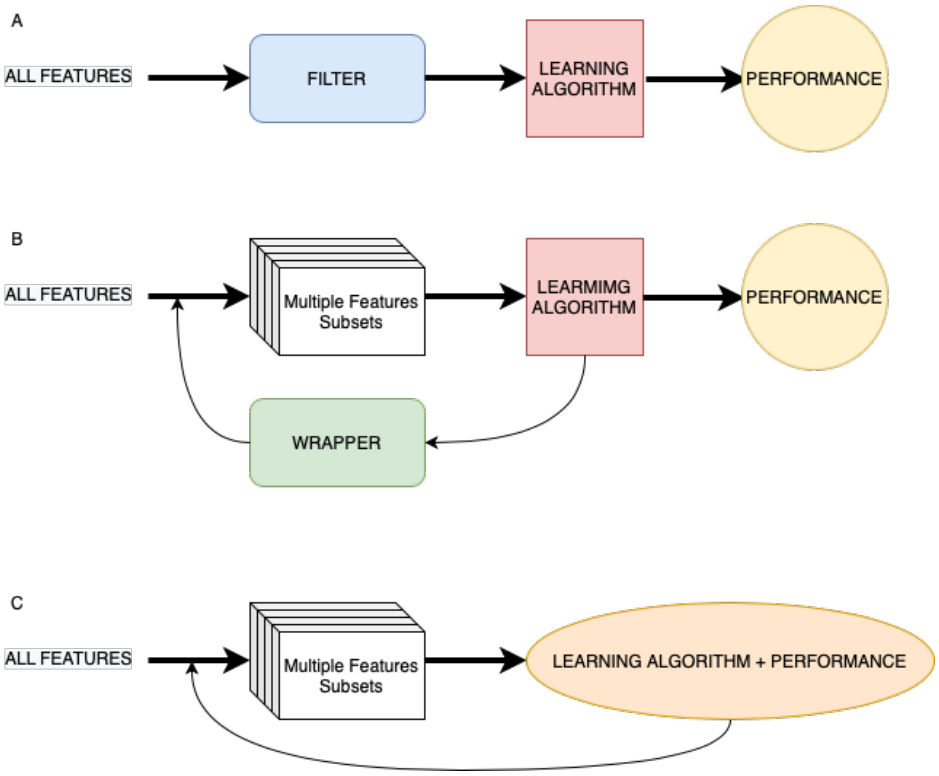


Figure 4.5: Types of features

Classification and clustering are two popular E.D.M. approaches. One prominent classification method is K-Nearest Neighbor (K-NN), modified in this research[50].To increase accuracy, we can use the best features that reflect the entire object; hence, feature selection methods are required. This work proposed a genetic algorithm with strong heuristic properties for automatic feature selection. The evolutionary algorithm improved the modified K-NN feature selection with 82.6 per cent accuracy. Classification and regression tasks require maximum information and lowest dimension feature sets. While several methods exist to reduce feature set dimensionality and improve existing features, only some generic techniques exist for designing features from numeric sequences. Autofead [51] is a unique genetic programming technique for automated feature design. Using a library of sequence-handling tools, a genetic programming variant generates a population of candidate features. Numerical optimization approaches ensure that the fitness of candidate algorithms is measured using optimal parameter values. Autofead is the first automated numeric feature design system to combine numerical optimization and traditional pattern recognition algorithms. T.P.O.T. (Tree-Based Pipeline Optimization Tool)[52]is a well-known tool based on G.A.s to solve the AutoML problem. T.P.O.T. is a framework that automatically constructs pipelines from input datasets. It includes cleaning, preprocessing, feature engineering, and model selection. It uses two operators to work on features: feature preprocessing and feature selection operators. An open-source Python (T.P.O.T.) implementation is evaluated on simulated and real-world benchmark data sets. It can create machine-learning pipelines that outperform basic machine-learning analyses with minimal user input and no prior information. It also incorporates Pareto optimization, which provides compact pipelines without sacrificing classification accuracy. Thus, this approach contributes to fully automating machine learning pipeline design. An advanced version of T.P.O.T. [45], Layered T.P.O.T. (L.T.P.O.T.), a T.P.O.T. variant that seeks to produce pipelines faster, improves computational performances by evaluating pipelines on large datasets. Using a modified evolutionary algorithm, this approach compares the fitness of candidate pipelines trained on different sample sizes. Despite the fact that L.T.P.O.T. is not necessarily better than T.P.O.T. in terms of overall performance, it often finds good pipelines faster and, on average, outperforms T.P.O.T.

KNN-based algorithms

Computational performance is one of the most important aspects of dealing with A.F.S., especially when wrapper-based methods evaluate thousands of features. The authors[53] set up a technique for reducing the time complexity of wrapper-based feature selection methods based on K-nearest neighbours (K.N.N.). The authors proposed building a classifier distance ma-

trix and incrementally updating the matrix to speed up calculation of the relevance criteria in evaluating the quality of the candidate features. The suggested approach significantly speeds up the wrapper-based feature selection process without compromising classification accuracy. The ReliefF method can be enhanced by using a new feature selection method based on kNN model(K.N.N.M.F.S.)[54] to select more meaningful representatives to replace the original data for feature selection; this method also incorporates the Heterogeneous Value Difference Metric to handle applications with both ordinal and nominal features. The Phenols dataset was used to test the effectiveness of the suggested K.N.N. model-based feature selection(K.N.N.M.F.S.)strategy. It is based on the computation of a similarity matrix that considers a related feature type in the neighbourhood. Experimental results show a considerable improvement in classification accuracy with the proposed feature selection strategy applied to the trial dataset. The performance of this approach was tested by using some metrics such as Mean Absolute Error (M.A.E.), Correlation Coefficient (CC), Relative Absolute Error (R.A.E.), Root Mean Squared Error (R.M.S.E.), and Root Relative Squared Error (R.R.S.E.) and average accuracy. According to the authors, K.N.N.M.F.S. shows improvement in terms of all the above evaluation metrics; in particular, it achieves an increase of 8.1% in average classification accuracy with a relatively small range of variance. Feature Selection via Supervised Model Construction (F.S.S.M.C.)[55] is presented. It is based on a classifier and is similar to the Relieff algorithm. The key idea of original Relief is to estimate the quality of attributes according tothe how efficient their values distinguish among instances.

They are near to each other, exploiting the K.N.N. algorithm. F.S.S.M.C. can handle many datasets, whereas Relief is sensitive to large ones. Various datasets are tested and compared with this framework, but F.S.S.M.C. outperforms. This proposed method efficiently transforms data and generates high accuracy for the training data. F.S.S.M.C.[56] is used for feature selection, while the original Relief is applied for ranking the features.

SVM-based algorithms

The predictive forward selection strategy is examined in this research [21] to enable SVM. The state vector machine is a sensitive classifier that selects the optimal feature from redundant and irrelevant features. For feature selection, two strategies are typically used. Although the filter approach is fast, the wrapper provides high accuracy. Predictive forward selection is based on the wrapper method. This strategy outperforms the traditional forward selection and brute force methods. Kernel functions are investigated using an SVM classifier to generate automated features.SVM kernel methods transfer features from nonlinear correlations in lower dimensions to linear relation-

ships in higher dimensions. While the method was designed to help classify features in a linear hyperspace, it can also help other classification methods by producing new features.[65]

Miscellaneous

E.H.R. Electronic Health Record [57] is the repository of patients, lab tests and medical his- tory. E.H.R. is made up of a massive database, and retrieving data for prediction is difficult. An automatic feature selection technique is proposed, which picks the important features from each file or dataset and stores them in E.H.R. This approach is based on logistic Regression. The Deep Learning Neural network assists in predicting the accuracy of the model. The transfer learning approach is used for features because extracting features from E.H.R. every time is a time-consuming step. Feature selection seek to reduce the feature space. Therefore, the L1 sparse logistic regression is appropriate for selecting features with small features. The scores of the selected features are saved in vector format. In order for the D.N.N. model to take in account associated parameters when predicting hypertension risk. Pattern recognition is achieved in D.N.N. due to an enhanced Transfer learning technique. A Hidden Markov Model (HMM) [58]is used to derive new features or better to incorporate sequential knowledge in the dataset made by a transaction sequence. These HMM-based features enabled a Random Forest, it is a nonsequential classifier, to use sequential information for the classification. The HMM-based approach provides help for automated feature engineering that can be used to model temporal correlations to improve the classification task and increase the detection of fraudulent transactions when combining with state-of-the-art expert-based feature engineering strategy for credit card fraud detection. This study[59]proposed the automated feature selection method for high-dimensional data based on weighting deals in Naive Bayes(N.B.). N.B. is a common classifier in supervised learning. Bayesian learning is used in high-dimensional spaces to achieve a soft feature selection scheme. This algorithm seeks weights in linear time complexity, which is a aided by lognormal prior distribution and posterior methods. Automatic feature Selection (A.F.S.) can be the result of the application of well-known regularization [60]methods (Ridge,

Lasso, and ElasticNet). Those methods were tested on linear and Poisson Regression as bug predictors for five open-source Java systems. Results showed that the three regularization methods reduce the prediction error of the regressors, improving their stability. In feature selection, the wrapper method is important for selecting optimal features from redundant data. This work[61]introduces ensembles of wrappers for an automated feature selection framework based on classifiers. Two classifiers, State Vector Machine(SVM) and Neural Network(N.N.), are utilized to obtain automated fish age classifi-

cation. Statistical pattern recognition techniques are applied for the validation of SVM and N.N. This method outperforms the other manual approach for feature selection in the wrappers method. F.E.A.I.D.S. [62] is a novel feature learning approach that improves learning algorithms. Features are made up of mathematical functions like mod, ceil, and sin. These characteristics increase learning efficiency in iterative ways. For nominal, numeric goal function, and periodical datasets, F.E.A.D.I.S. improves the performance of learning algorithms. F.E.A.D.I.S. improves the efficiency of both meta and active learners. The authors[63] investigated a machine-learning paradigm designed to deal with logs in enterprise security. It detects malicious activity from the log at an enterprise level. This approach generates new features from the raw data by applying new operators. The formula is produced from the raw features with the assistance of the boolean function. This method applies Fourier analysis to ranks based on their highest coefficients. Feature engineering is enhanced with a wide range of classification and clustering algorithms. It improved by 50.6 per cent as compared to raw features. Emperor Penguin Optimizer(EPO)[66] is a metaheuristic algorithm inspired by penguins' hud- dling behaviour. There are eight EPO variants, four of which are studied. The suggested approach considers four transfer features, S-shaped and V-shaped, to map the search space into a distinct research space. A binary emperor penguin algorithm is proposed in the selection of features. It's a combinatorial NP-Hard Problem. The binary emperor, penguin optimizer approaches, are utilized to discover the best possible subset of features. Feature count and classification error rate are factors in the fitness function applied by the proposed method.

4.5.2 Automatic feature selection based on Deep learning and Neural Network Approaches

Learning Feature Engineering (L.F.E.)[67]is an automatic feature engineering framework specific to classification problems only. It performs interpretable feature engineering for classification based on learning from past feature engineering experiences. Using a collection of datasets, a set of neural networks (Multi-Layer Perceptron) is trained to predict the transformation that positively impacts classification performance. Results show that L.F.E. outperforms other feature engineering approaches for a majority (89%) of the datasets from various sources while resulting in significantly less expensive with respect to computational costs. This study [68] aims to propose a neural feature learning framework for automatic feature selection in the relational dataset. In the first stage, a Rule-based trans- formation tree is constructed with aggregation functions max, avg, sum, and min based on deep feature synthesis (D.F.S.). In the second stage, R2N employs supervised learning to

train a relational recurrent neural network to learn transformations. There are some deficiencies reported in R2N. It could be more efficient in autotuned. Joining a smarter graph is an NP-hard problem. Explorekit [69] is a novel automatic feature generation framework. The purpose of this framework maximizes the predictive power by combining candidate features and original features. It is a learning-based approach that predicts new features and provides improved results compared to existing features. Classification techniques are applied to reduce errors. Learning Automatic Feature Engineering Machine(L.A.F.E.M.)[70]deals with feature engineering problems over Heterogeneous Transformation Graphs (H.T.G.).A deep Q-learning on H.T.G. that can pass engineering optimal function knowledge from a collection of datasets to previously unseen datasets. The H.T.G. is a directed heterogeneous acyclic graph; each node in H.T.G. represents a feature (feature node) or a dataset (dataset node), and each function node corresponds to one of two features: original features and derived features. L.A.F.E.M. is a modern architecture for automated feature engineering that combines feature formulation, feature selection, and model evaluation (F.E.). It includes a heterogeneous transformation graph (H.T.G.) that organizes the processing of F.E. and deep reinforcement learning algorithm that can be perform automatic F.E. on the H.T.G. Machine learning play roles in solving the Twitter spam detection problem. Feature engineering is crucial in ML for spam detection. Existing work to detect spam tweets is mainly based on basic features, meta-data, and social relationships, which require intensive human manual efforts. This study[71] recommends a deep learning framework for automated feature engineering that extracts features from text data. This approach is depends on a deep neural network trained with Bi-LSTM. Deep-learnt features are those features that can detect spam on Twitter. According to the results, it outperforms both word2vec and statistical features. The deep learning framework Midway Neural Network (M.N.N.)[72]for automatic feature engineering is proposed. This method uses a manual feature, feature engineering, and log preprocessing. It is used to analyze communication network logs, which contain high-dimensional data that is rapidly growing and diverse and is based on L.S.T.M.The raw data is encoded and sent to the midway, where it is further processed and predicted. Authors[73] presented an automatic feature selection method based on the neural network Multi-Layer Perceptron(M.L.P.), which extracts complementary and diverse classification features from high dimensional data. It reduces the dimension of data and increases the accuracy. M.L.P. is set up for the classification of data. In the first stage, the hidden layer is labelled singlet. In the second stage, an ensemble of 3 networks is labelled as small-scale. In the third stage, an ensemble of networks, each with 30 hidden nodes, is labelled as large-scale. This article[74]discussed automatic feature engineering using ex-

ploratory and learning techniques. Technique is an efficient and cost-effective learner. In the explorer, a transformation tree is constructed using feature transformation space and pruning to find the optimal features. The Learner-Predictor features are learned through their historical data and correlation of features. The transformation occurs, which the classifier determines and may or may not apply. This work[75] defines the novel Neural Aggregate Generator (N.A.G.), a neural network based feature extraction module that learns feature aggregates end to end on the fraud classification tasks. In contrast to other automatic feature extraction approaches the network architecture of the N.A.G. closely mimics the structure of feature aggregates. Furthermore, the N.A.G. extends learnable aggregates over traditional ones through soft feature value matching and relative weighting of the importance of different feature constraints. It compares the performance of the N.A.G. to the state-of-the-art approaches on a real-world dataset with

Millions of transactions. More precisely, it shows that features generated with the N.A.G. guid for the improvement of results over manual aggregates for fraud classification, thus demonstrating its viability to replace them. Moreover, N.A.G., compared to other automatic approaches such as the L.S.T.M. or a generic CNN, outperforms the results. An Interactive Reinforced Feature Selection (I.R.F.S.)[75] framework that guides agents through self-exploration and external professional trainers to enhance feature exploration learning. In this framework, there are two model search strategies: KBest and Decision Tree : (1) identify aggressive and hesitant agents to diversify agent training, and (2) allow both trainers to teach at different phases to combine their experience and diversify teaching. This hybrid teaching technique can help agents learn more and become more effective. Finally, extensive tests are performed on real-world datasets to verify our method's advantages over the reinforced selection and conventional feature selection. The security of BGP (a boarding gateway protocol) is a huge concern in the network domain. This work [77] introduces a framework of automatic feature extraction of BGP anomaly detection that detects and alerts the security threat caused by the events. It is based on a neural network approaches which extracts features from large-scale raw data.

4.5.3 Automatic feature selection based on Dimensional Reduction of Data

Approaches Condition-based maintenance (C.B.M.)[79] is an essential component of Condition Monitoring. This automated feature engineering framework can manage a comparison of dimensionality reduction and feature selection algorithms that collect information automatically during monitoring. The purpose of this approach is to address concerns with anomaly detection

in the monitoring system. C.B.M. is a sensor-based IoT system. It evaluates assets, detects anomalies, and even predicts them before they occur. It is a data-driven process that can handle labelled and unlabeled data. The P.C.A. and LDA algorithms are applied to different datasets and evaluated for accuracy. This article is about the source code of Python. Pattern identification derives commonly used operations from source code and notebooks—pattern recognition. The framework suggests a new frequent system to draw from a list of source codes based on the frequent method for extracting frequent characteristics engineering. The three-stage analysis process of the strategy is as follows: (1) A.S.T. conversion source code, (2) tree node type/semantics inference, and (3) regular A.S.T. mining subgraph pattern. This approach seems fine for a small-scale problem, but pattern detection is a complex task for high-dimensional data. [78]

4.5.4 Automatic feature selection based on Relational Database Approaches

This article[80] describes extracting features' nominal attributes from a relational database. New features are created based on functions MIN, M.A.X., S.U.M., A.V.G., COUNT, etc, which are highly correlated and increase the accuracy and predictive power. Similar work has been done on the state-of-the-art paper Deep Feature Synthesis. The proposed approach produces novel features that are highly associated with the target tribute. The tests yielded a better result after the addition of the new feature. Three types of databases are used in the experiments: a financial database, a medical database, and an East-West database, all of which have nominal attributes. Some disadvantages, such as the virtual joining paths in the data model, add to the difficulty. The approach also has the drawback of assuming attribute independence. Dependencies between multiple attributes need to be captured, leading to the absence of potentially relevant concepts.

One Button Machine (OneBM)is a highly cited article[81] for automating feature engineering in a relational database. As the name shows, one button keys data scientist activities perform automatically. In this framework, key features are generated and transformed from the relational database. OneBM automatically joins database tables and applies advanced data transformations to derive important features from data. The findings of this paper have been published in the Kaggle competition. The results suggest that both experts and novices can use OneBM. It allows data scientists to explore more ideas in less time. A graph-based approach has been used; the tables in the dataset are represented as nodes, and relationships are denoted with the edges. Automated Feature Synthesis (A.F.S.)[82]is proposed for the relational database using dynamic programming. This article is based[81]

on a data science machine and a single button. Their research revolves around feature synthesis for relational data from structured and unstructured data forms, efficiently automating a time-consuming manual activity for non-experts while consuming a reasonable amount of resources. It covers automatic feature synthesis for structured and unstructured data for both classification and regression models. Three datasets have been tested and compared with one button and a data science machine. This is the most significant article as it is a highly cited paper. This study proposed a Data science machine (DSM)[83]. it harnesses the predictive power of modelling automatically from raw data. Deep feature synthesis (D.F.S.) is an algorithm that generates new optical features automatically from the relational dataset by applying sequential mathematical functions and then creating a machine learning pipeline that tunes with a novel Gaussian copula approach. D.F.S. is an automatic algorithm that selects features similar to feature selection supported by human intuition. The contribution of this article is to design a D.F.S. algorithm that generates features automatically from a large feature space and auto-tunes an ML pipeline to extract optimal synthesized features. The case study of commerce is discussed in this document, including whether the customer will purchase inexpensive or luxurious items. In the dataset table, it is considered an entity feature (defeat), which is joined in two ways, forward and backwards relationships; features are derived from instances using mathematical functions (min, max, and count).

4.5.5 Automatic feature selection based on Knowledge Graph approaches

. K.A.F.E. [84]is an automatic feature engineering enhancement framework based on knowledge engineering. It is an automatic technique for improving features that makes use of knowledge graphs, web tables, and a repository hosted on the internet. structural knowledge links with the knowledge database to enhance new features automatically. This framework reduces the hectic load on data scientists; feature engineering is tedious and time-consuming. The Input features to the K.A.F.E. system feature process from the semantic feature index Wikipedia, DBpedia, etc. feature lookup in the knowledge repositories and map the relevant content on features. An excessive number of features are mapped, and redundant features are eliminated based on a correlation matrix. Optimal features added to the existing input features applied to models. The model's predictive power in the form of accuracy has been increased significantly. A novel technique, but it is time-consuming and expensive to compute the data across the internet. Automatically extract features for general learning tasks. This work[85] addresses the issue of constructing good features from semantic knowledge bases—an au-

tomated feature learning framework [86]for unstructured data over the web. Data is collected with expert knowledge and tracked features in an automated manner. These are unsupervised learning techniques without

They are using Euclidean distance to build feature representations automatically with expert knowledge. Two main issues arise in this method. Identifying significant similarities in sparse features in high-dimensional data is difficult; the relationships between categories are extremely complex. A mixed-initiative framework [87] for feature engineering is a semi-automatic approach. It is based on domain knowledge and capture knowledge graph and is supported by an interactive visualization method. This framework consists of the integration of complex heterogeneous data and builds a knowledge graph. Feature engineering techniques are applied to the K.G. a new feature is generated with an interactive graph. Feature analysis and mining techniques are applied to refine the features iteratively. Brainwash[88] is the data system for automatic feature engineering based on a recommendation system. It covers web search, Google's Knowledge Graph, I.B.M.'s Watson, and other ML-enabled rich databases. This framework is a trained system for automatic feature engineering for huge datasets. It facilitates the developer to provide a hint on how their code interacts with data and reduces the effort of a long wait and debugging. Brainwash is an easy-to-explore, extract and evaluate interaction loop for a trained system.

4.5.6 Automatic feature selection based on Probability and statistics approaches

This is the base and pioneer paper of the auto-learn framework. Autosklearn is a modern AutoML framework built on Python's sci-kit-learn machine learning package. Autosklearn is based on efficient Bayesian optimization methods. In addition to Bayesian optimization, the meta-learning technique is used to optimize the ML method. ML machine is recommended for meta-learning. Bayesian optimization (B.O.) takes a long time to start for hyper-parameter spaces in entire ML systems. B.O. provides benefits by selecting configurations with meta-learning and then seeding with data.[89] The AutoLearn[65] feature learning algorithm is regression-based, a data-driven strategy without domain knowledge. Establishing pair attribute correlations improves the prediction accuracy; it identifies linear or nonlinear relationships between individual pairs. Autolearn has four components: (1) preprocessing, (2) mining correlated features, (3) feature generation, and (4) feature selection. It searches the original feature space for pair-side correlation features using distance correlation. To assess the predictive relationship between correlated features, regularised regression models transform the original feature space accessible into a new feature space. Two types of features exist:

The first sort searches in feature pairs for hidden patterns, and the second measures the distance between the projection and the independent variable. B.M.I.L. [90]is based on a Bayesian algorithm. This is a novel approach proposed in this paper, Automatic Feature Selection Bayesian multiple instance learning algorithm. It is a classifier that selects optimal datasets by using inductive learning. B.M.I.L. is highly efficient in selection as it selects a small dataset and performs better than previous M.I.L. algorithms. In B.M.I.L., a list of selected features and weight vectors for the linear classifier, the redundant selected features are removed using the M.A.P. estimator. This process is iterative; the hyper-parameters are updated until the optimal features are selected. This article focuses on reinforcement learning(R.L.); the study proposes[91] using conditional shared knowledge to test the independence of return and state-feature sequences explicitly. It creates a Markov decision process for the R.L. problem. Conditional Mutual Information is used to select features. It selects the function that mostly increases conditional mutual knowledge. An extended version of

Joint Mutual Information (J.M.I.)[92] is proposed in this work, which is called Historical Joint Mutual Information (H.J.M.I.). This method solves the J.M.I. score-stopping criteria in feature selection. In J.M.I., the users are unable to define the stopping criteria; as a consequence, they need to evaluate the whole learning chain to select the optimal features on the criteria of the lowest training error in the learning. It develops stopping criteria without exploring the entire chain, and developers choose the best feature before the actual learning is completed. The decision of the sliding window period depends on the domain knowledge and calls for trivialities. Currently, no automatic method to handle the sliding window aggregate features selection[93]. Feature generation with different periods and also sliding windows takes a long time. It is very hard to enumerate them all and then select them. the study proposes a general framework using Markov Chain for solving this problem. This framework is very efficient and has high accuracy; it can perform feature selection on various features and period options. The general selection of features for functions extracted by sliding windows includes time-consuming iterations to generate features in time series, followed by standard approaches feature selection for the classification. The selection of the major parameter, namely the sliding time, is subject to domain knowledge and needs triviality. There is currently no automatic method for selecting the aggregate function of a sliding window. Because listing and selecting the duration of feature production with multiple periods and sliding windows is difficult, this article [94] presents a general approach to solving this problem with the Markov Chain. This frame is incredibly efficient and precise, allowing a variety of features and time options to be specified. Applying the current Markov chain theory can add more sliding windows and

aggregation operators. Vector of Statistics from Time series(VEST) [95]is a novel and automated feature-engineering approach. VEST is designed to perform feature engineering using univariate and numeric time series to solve forecasting problems. VEST is composed of three components: transformation, summarization and selection. Time series is transformed into several representations; Data is summarized using the statistics functions mean and standard deviation (S.D.).A feature selection filter is used to cope with high dimensional problems, and the final set of features is added to the original representation. After feature selection, an auto-regressive model is applied to the final set of features.

4.5.7 Automatic feature selection based on hybrid Approaches

Four feature extraction methods are introduced in this study[96]. All of them are fully automated to reduce the data dimension. Sensor and cyclical time series data are used to generate big data. This method is essential in machine learning to avoid model over-fitting and the curse of dimensionality. These algorithms can extract features from cyclical time series and retrieve data from the frequency or time-frequency domain's local and overall cycle shape. It is a reliable method in decision-making for predictive maintenance of the machinery equipped with sensor data. A collaborative data scientist paradigm [97]is presented in this research. Independent data scientists intervene together to produce new features in real-time. Feature engineering source code is generated in the context of a single predictive machine learning model by independent data scientists. This approach is highly human-dependent. Autofeat[98] is a library in the sci-kit learn package that automatically creates features in regression and classification problems. Complex nonlinear machine learning models are difficult to train. The Autofeat library generates nonlinear features that improve the prediction accuracy of a linear model while retaining its interpretability through a multi-step feature engineering and selection process. The AutoFeatRegressor and AutoFeatClassifier models in the AutoFeat library create and select additional nonlinear input features based on original data. Models are available with a familiar interface of scientific learning. The autofeat library is particularly useful for heterogeneous data sets such as sensor measurements in various physical units. The Automatic Feature Engineering Machine framework (A.F.E.M.) is a novel strategy for automatic feature selection. This paper[99] includes an understandable framework to compute derived features. The approach uses a set of predefined families of operators to derive new features. It is considered a mixed form of top-down and bottom-up approach in which features are added to an initial set based on the application of specific operators. It

seems like a numerical approach in which iteratively new features are added. Features learnt by deep learning are opaque. In contrast, derived features have explicit meaning. Symbolic Regression (S.R.) is a famous task in Evolutionary Computation (E.C.). Traditional Regression is always preferred over E.C. due to its sound mathematical foundation. The E.C. method is used for automatic feature engineering to make smaller solutions easier to understand. The Kaizen Programming (K.P.) [100]method is proposed, which is based on a hybrid approach of E.C. and statistics. E.C. creates features and statistical methods to build efficient models. K.P. provides a quality solution compared to the traditional G.P. Brain-Computer Interface(BCI), which is based on neuron signals generated by the brain. BCI is used as brain E.E.G. signals communication via commands. Automatic feature selection methods for BCI are introduced in this paper[101], which use correlation of s-coefficients and t-statistics; these are ranking-based methods of features. High-speed feature selection takes place in BCI as compared to other methods. S-coefficients are graded according to the degree of correlation, and t-t-coefficients are summed (t-statistics). Automatic feature engineering is a time-consuming task; therefore, a method is proposed that deals with the interaction of features to improve performance and prediction accuracy. Three heuristic methods are also introduced to reduce the measurement cost during feature interaction. The average accuracy is reported as 95 per cent.[102] The scalable Automatic Feature Engineering Framework SAFE[103] is proposed for large-scale and massive data. It is an efficient framework for AutoML feature engineering, but its practical application in industry is still a way to go. xgboost model is applied for the feature selection. It is an iterative algorithm that repeats until it finds the best feature from a large dataset. An extended framework of automated feature engineering based on UML models is proposed[104]. To date, a fully automated feature generation approach has yet to be applied using the UML model approach because UML itself is a complex task in software engineering and needs domain model experts to construct UML modelling. This paper deals with new kinds of use cases in the requirement engineering of product lines, classification of an activity diagram and transformation of use cases and activity diagrams into a feature model.AutoML explored high-order interactions based on automated function selection. AutoGroup[105] is an end-to-end model that performs a selection of feature interactions. AutoGroup automatically generates interactions of feature sets using a novel function. It consists of feature groups in several function sets. AutoGroup can concurrently resolve dimensions reduction and selection difficulties not seen in prior models. Offline trials on three large-scale public benchmark data sets reveal the offline tests demonstrating AutoGroup's performance and efficiency over state-of-the-art models. Data preprocessing, feature engineering, and hyperparameter optimiza-

tion approaches are implemented for ML-supervised learning; the methods only apply to tabular data in the textual format. It is primarily concerned with prediction and classification tasks. Data scientists spend much effort in preprocessing and Feature engineering. Automating these processes would greatly improve the ML pipeline's efficiency. In this study[106], four methods are applied to features. 1. Principal Component Analysis (P.C.A.) to reduce the dimension of the data.2. Analysis Of Variance (ANOVA) This statistical method compares the target variable to the predicted variable. It assumes that the group variance is 0. The alternative is that at least one variable's variance is different- ent.3. Pearson correlation: When two features are highly correlated, one can be eliminated without compromising the data. 4. Select From the Model class of the Scikit-Learn feature. The Selection module is a meta-transformer that removes unnecessary features from data. It can be used with a model that allows feature coefficients or weights. A threshold parameter removes features whose weight falls below it.

4.5.8 automatic feature selection based on Reinforcement Learning

The study examines the effectiveness and efficiency of automated feature selection balance. It addresses an optimum feature selection computational problem. The classic feature selection techniques are effective, yet the optimal feature selection is difficult. The reinforced techniques of selection of features detect the best subset but could be more efficient. This study [107]offers a new feature size with an I.R.F.S. framework to resolve the issue of the computation. I.R.F.S. assists learning agents in improving the investigation of features. In this context, two trainers, KBest-based and Decision Tree-based trainers, are competent in two searching techniques. I.R.F.S. has exceeded the classic selection process and the current approach to reinforced selection. It is more cost-effective and efficient. Generating useful features from raw data is extremely difficult, as the search space contains many non-informational features. This study proposes an automated new performance-driven framework [108]to provide discrimination based on raw data using reinforcement learning, which helps enhance the downstream classification prediction by default. This framework unifies the characteristic structure, interpretation, and computation logic. Financial data shows that the offered approach has improved significantly over our years of field expert knowledge and genetic programming. Furthermore, this Financial Data Reinforcement Learning (F.D.R.L.) frame can be easily modified for various purposes because of its adaptability.

Chapter 5

Introduction to Financial Market

5.1 Overview of the Financial Market

Considering our current understanding of features and possession of all the requisite features for our study, it is critical to grasp the characteristics and dynamics of the financial market. Numerous underlying concepts impact the financial market; these will be elaborated upon in the following sections. Nevertheless, it is critical to commence by comprehending the foundational concept of a market in finance.[109] The financial market operates as a dynamic ecosystem wherein individuals, organizations, and entities exchange assets associated with the financial market. Cryptocurrencies,[110] currencies, and equities are included (coin pairings). Presently, every country exerts a substantial influence on the worldwide financial market. This market is worth trillions of dollars on a global scale. The facilitation of capital allocation and [111] promotion of economic expansion are critical functions financial markets perform. The financial markets exert an indirect impact on the national economy.The cryptocurrency market is considered a subsidiary sector within the broader financial market. It is an emerging and unique industry that conducts transactions solely in cryptocurrencies, unlike the traditional financial market, which encompasses various assets, including equities, bonds, commodities, and fiat currencies.[112] The cryptocurrency market is devoted exclusively to virtual or digital currencies for personal use. Implementing cryptography for security purposes, the cryptocurrency market comprises numerous concepts, including digital assets, the residences of market participants, market mechanisms, risk, and return. These components contribute to the expansion and heightened scale of the financial market. A significant proportion of investors, banks, organizations, and speculators from around[112] the world participate in crypto market activities. Globally, numerous operational stock markets exist, one of which is the market

in the United States. Markets: cryptocurrency exchanges facilitate trading on the European and Japanese markets; Binance, Kucoin, and Coinbase are renowned exchanges utilized for trading. Understanding the fundamental and most critical concept in cryptocurrency trading is the comprehension of candlesticks. What are candlesticks and their behaviors? The market trend, which the market observes, is the second most essential concept to comprehend.

5.2 Introduction

It is among the earliest forms of commerce that the commodities exchange has been conducted for millennia. Humans initiated commerce to acquire necessities or desires that surpassed their production capabilities. Trade became more sophisticated, and humans devised novel methods of exchanging products and services as societies progressed.[110] Bartering was among the earliest forms of commerce, in which goods or services were transferred for other goods or services. Services rendered. Although this system functioned admirably for centuries, locating a person who desired what you had to give or who possessed the item you sought was frequently challenging. To solve this dilemma, individuals started employing various objects as forms of currency. As trade expanded and societies grew more complex, individuals initiated the development of more advanced financial systems. An early illustration of this phenomenon can be observed in the evolution of financial systems in ancient civilizations like Babylonia. Merchants and traders deposited money and merchandise with temple clerics who served as bankers. During the Middle Ages, guilds were established to oversee commerce and safeguard the interests of merchants as finance and trade continued to advance[113]. International commerce necessitated the development of modern banking. Merchants required a secure method to transfer funds across borders. Present-day financial markets are sophisticated and complex, offering an extensive selection of instruments and trading strategies. Facilitating capital flow and providing businesses with the necessary funding for expansion and innovation fulfill an indispensable function in the global economy. The financial sectors have since the earliest days of [114]bartering and trading, and they continue to develop alongside the introduction of new technologies and innovations. The financial market operates as a dynamic ecosystem wherein individuals, organizations, and entities exchange assets associated with the financial market. Cryptocurrencies, currencies, and equities are included (coin pairings). Economically speaking, various markets exist, ranging from airport candies to the global forest industry. The market participants increase their profitability by maximizing their positioning and geographic area. Presently[115], every country exerts a substantial influence on the worldwide financial market.

This market is worth trillions of dollars on a global scale. The facilitation of capital allocation and promotion of economic expansion are critical functions financial markets perform. The financial markets exert an indirect impact on the national economy[116]. The cryptocurrency market is considered a subsidiary sector within the broader financial market. Investors trade ownership in companies on the equity market, also known as the stock market. A person's stock ownership signifies partial ownership in that company; if they hold every stock, they acquire ownership of the complete organization. It is an emerging and unique industry that conducts transactions solely in cryptocurrencies, unlike the traditional financial market, which encompasses various assets, including equities, bonds, commodities, and fiat currencies. The cryptocurrency market is devoted exclusively to virtual or digital currencies for personal use. Implementing cryptography for security purposes, the cryptocurrency market comprises numerous concepts, including digital assets[117], the residences of market participants, market mechanisms, risk, and return. These components contribute to the expansion and heightened scale of the financial market. A significant proportion of investors, banks, organizations, and speculators from around the world participate in crypto market activities. Globally, numerous[118] operational stock markets exist, one of which is the market in the United States. Numerous cryptocurrency exchanges facilitate trading on the European and Japanese markets; Binance, Kucoin, and Coinbase are renowned exchanges utilized for trading. Understanding the fundamental[119] and most critical concept in cryptocurrency trading is the comprehension of candlesticks. What are candlesticks and their behaviors? The market trend, which the market observes, is the second most essential concept to comprehend.

5.3 Crypto Market

Cryptocurrencies, digital or virtual currencies protected by cryptography, represent an unprecedented financial innovation. A decentralized network of computers enforces an open ledger using blockchain technology, guaranteeing the integrity and security of transactions. Cryptocurrencies[120] are not minted by a central authority, unlike conventional currencies, which renders them impervious to government intervention or manipulation. The designation "crypto" refers to the cryptographic methods and encryption algorithms that protect these currencies, enabling secure online transactions without intermediaries. Blockchain is fundamental to the operation of cryptocurrencies;[121] it stores authenticated transactions in a chain of connected blocks on a distributed ledger. Every individual block is subjected to autonomous verification by validators on the network, resulting in an immaculate record of transactions. The potential of blockchain technol-

ogy transcends cryptocurrencies, as it finds utility in diverse sectors, supply chains, and operational procedures, including online voting and crowdfunding. Blockchain technology is utilized by financial institutions[122] such as JPMorgan Chase & Co. to reduce transaction costs and expedite payment processing.

Figure 5.1: Financial market

Diverse varieties of cryptocurrencies function for objectives within their blockchains[123]. Utility tokens consist of XRP and ETH, which execute operations on their respective blockchains, and transactional tokens, such as Bitcoin, which are intended to make payments. The diverse categories include tokens for governance, platform tokens that facilitate blockchain applications, or security tokens that signify []asset ownership. Comprehending these classifications enables investors to evaluate the potential and intent of a cryptocurrency, differentiating those that possess a distinct utility from those that lack a distinct function[124]. Although the legal standing of cryptocurrencies is still an intricate subject, they function independently of conventional financial infrastructure and do not receive support from either public or private organizations. The legitimacy of fiat currencies is derived from monetary or governmental authorities, while cryptocurrencies continue to encounter obstacles to establishing their legal status in various financial jurisdictions around the globe.

5.3.1 Candlestick

A candlestick is a graphical depiction of price fluctuations in financial markets frequently employed in technical analysis to examine and interpret past price data. Each candle offers insights[125] into the opening, closing, high, and low prices for a designated time interval, a minute, day, week, or any other time period. An individual candlestick Has numerous elements. The body is the initial and primary component of the candlestick. A variety of financial

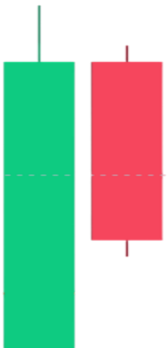


Figure 5.2: CandleStick

charts and candlesticks illustrate the price fluctuations of an asset over a specified period[126]. Each candlestick represents a time period—hourly, daily, weekly, and so forth. The opening, highest, lowest, and closing prices for that period are presented. The candlestick's body is[127] rectangular, symbolizing the prices at opening and closing. Furthermore, two wicks are present to symbolize the highest and lowest prices, respectively. When the opening price is below the closing prices, body is customarily filled in green; in the opposite scenario, known as the body, it is filled in red. Additionally, candlesticks can reveal the demand and supply for a given period.

Body

The body of a candlestick refers to the rectangular region bounded by the open and closed prices at a given moment. When the close value exceeds the open value, the body is conventionally[128] filled or depicted in green or white; when the open is greater than the close, the body is devoid of any representation, shaded, crimson, or black in color. [1]

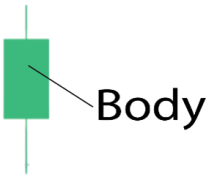


Figure 5.3: CandleStick Body

[1]

Wicks

Wicks are the slender lines that snake above and below the body. The greater Wick Indicates the maximum price attained. During that period. The reduced Wick. Indicates the minimum price attained within that specified time interval.

Figure 5.4: CandleStick Wicks

Closed and opening prices

The launch price denotes the initial price exchanged within a given period. The last-traded price during the specified time period constitutes the closing price.

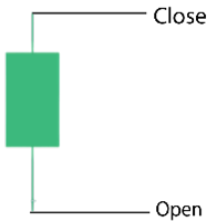


Figure 5.5: CandleStick Close and Open Price

5.4 Patterns in the candlestick

5.4.1 Positive Patterns

Bullish Engulfing

A larger bullish candlestick engulfs the preceding smaller bearish candlestick. [2]

[2] https://www.dailyfx.com/education/candlestick-patterns/bullish-engulfing.html

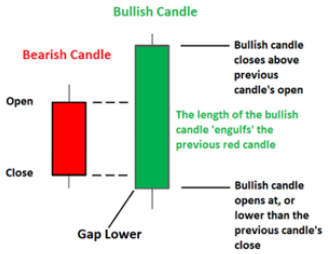


Figure 5.6: Bullish Engulfing

Hammer

A small body with a long lower fringe at the peak, which suggests the possibility of a reversal following a downtrend.

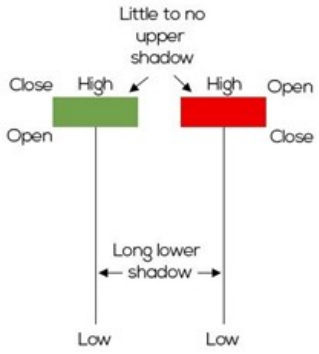


Figure 5.7: Hammer

[3]

5.4.2 The Bearish Sequence

Bearish Engulfing

A larger bearish candlestick engulfs the preceding smaller bullish candlestick.

Shooting Star

A small body with an extended upper wick at the bottom of an uptrend, indicating a possible reversal. [4]

[3]https://www.thinkmarkets.com/en/learn-to-trade/indicators-and-patterns/general-patterns/hammer-candlestick-pattern/

[4]https://fxopen.com/blog/en/shooting-star-pattern/

Figure 5.8: Shooting Star

5.4.3 Indecision or Reversal of Trend

Doji

A doji is formed whenever the opening and closing prices are identical, indicating market indecision. [5]

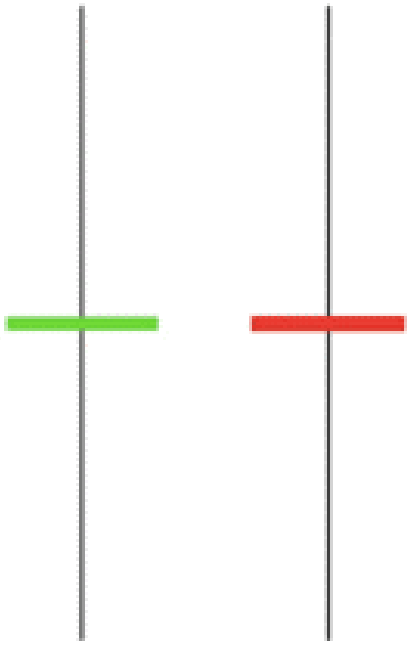


Figure 5.9: Doji

spinning

A spinning top is a small body with upper and lower wicks that spin, signifying a tug-of-war between sellers and purchasers. [6]

[5] https://www.learnstockmarket.in/technical/doji-candle-stick/
[6] https://www.dailyfx.com/education/candlestick-patterns/spinning-top-candle.html

Figure 5.10: spinning

5.4.4 Continuation Structure

The absence of wicks on a marabou candlestick indicates robust momentum in the desired trend direction. Candlestick charts offer a graphical and instinctive method for assessing market sentiment, price trends, and reversals. Analysts and traders make informed judgments regarding the entry and exit of positions in financial markets by analyzing candlestick patterns.

Figure 5.11: Continuation Structure

5.5 Trends

In financial markets, a trend refers to the general direction in which the prices of assets are moving over time. Recognizing and understanding trends is fundamental to technical analysis, a method traders and analysts use to make predictions about future prices movements based on historical data.

5.5.1 Types of Trends

Uptrend

A series of higher high and higher low characterize an uptrend. Prices generally move upward, indicating bullish market sentiment. Investors often seek opportunities to buy during uptrends. [7]

Figure 5.12: Uptrend

Downtrend

A sequence of lower high and lower low marks a downtrend. Prices generally move downward, indicating a bearish market sentiment. Investors may consider selling or shortening during downtrends. [8]

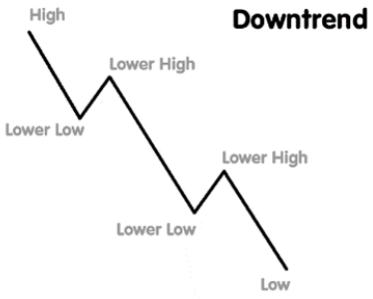


Figure 5.13: Downtrend

Sideways or Range-bound Trend

Prices move within a horizontal range with no clear upward or downward direction. This trend is also known as consolidation, and it often occurs when the market is indecisive or lacks a dominant trend. [9]

[7] https://www.babypips.com/forexpedia/uptrend
[8] https://www.babypips.com/forexpedia/downtrend
[9] https://www.learnstockmarket.in/term/sideways/

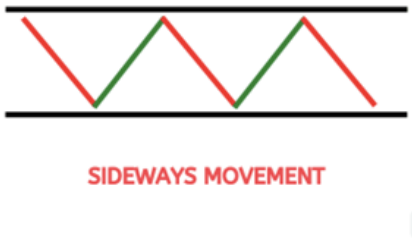


Figure 5.14: Sideways or Range-bound Trend

5.6 Market Sentiment

Market sentiment, also referred to as investor sentiment, is a substantial gauge of investors' overall perspective and disposition[129] regarding a particular security or the wider financial market. This attitude, reflected in pricing trends, significantly affects supply and demand and, thus, price fluctuations. When market sentiment is bullish, as evidenced by an increase in prices[130], and bearish, as prices decrease, traders employ sentiment indicators in conjunction with other analysis techniques to optimize entry and exit signals. The objective is to achieve the highest possible returns by precisely and promptly assessing and responding to market sentiments. The Behavioral Finance Theory, as proposed by Kahneman and Tversky, examines "irrationality" among investors that is psychologically grounded. Investor behavior is substantially influenced by cognitive and affective biases, including but not limited to overconfidence, reliance on previous performance, or difficulty adjusting viewpoints to new information. Investors' decisions, which are frequently influenced by perceptions and the decisions of other investors[131] rather than rigorously adhering to fundamental principles, impact market dynamics. [10]

Figure 5.15: Market Sentiment

John Maynard Keynes' Animal Spirit Hypothesis posits that in situations of uncertainty, individuals succumb to cognitive biases and allow their emotions to guide their decisions. Connected to market sentiment[132], herd behavior causes investors to migrate to thriving markets with impractical expectations, thereby contributing to market downturn excesses and inefficient prices. Diverse market sentiment-driven trading strategies exist. Long-term investors anticipate capital gains and rewards during periods of favorable sentiment by the prevailing sentiment. In contrast, contrarian investors oppose the dominant sentiments by identifying opportunities in undervalued equities with solid fundamentals. They do so during periods of extreme pessimism to capitalize on market sentiments when they reverse. Market sentiment is of the utmost importance to day dealers and technical analysts, as it dictates short-term price fluctuations based on the psychology of the masses. By trading in opposition to the prevailing trend, contrarian investors can profit from market sentiment when they trade against the consensus. The denotations "bullish" and "bearish" pertain to market sentiment and signify price movements in an upward or downward direction[133]. It is critical to observe that fundamental market adjustments and sentiment do not invariably coincide. Sentiments frequently influence the stock market, and price fluctuations may transpire for motives foreign to fundamental analysis. The basic value is based on actual business performance, whereas sentiment in the marketplace reflects broad issues, expectations, or emotions. Investors must consider both elements to attain a comprehensive understanding of the market.

5.7 Depth of Market

Depth of Market (DOM), alternatively known as the depth of the market or the order book, is an essential notion within financial markets. It furnishes an all-encompassing perspective on the interplay between demand and supply for a specific asset[134]. In addition to presenting the highest bid and offer prices, it furnishes an exhaustive inventory of all outstanding sale and purchase orders across different price tiers. This data is of the utmost importance for speculators attempting to comprehend the intricacies of the market and possible fluctuations in prices. Market depth is represented graphically through a dynamic computerized list that is perpetually updated in real time to mirror modifications in the purchased book. The system comprises a sequence of price levels, each representing the aggregate quantity of orders to buy or sell. Traders can evaluate the liquidity of an asset and forecast its price reaction to different trading activities by analyzing this data. A depth

[10]https://alternative.me/crypto/fear-and-greed-index/

of market exhibit comprises the price threshold, number of orders, and market actors as its principal components. The price level denotes the precise cost at which orders are transmitted, while the order quantity discloses the overall quantity of contracts or shares purchased at that price level. Each buyer (bids) and vendor (asks) identified as market participants contributes to the aggregate market sentiment.

Figure 5.16: Depth of Market (DOM)

Comprehending market sentiment via DOM is crucial to making well-informed trading decisions. To forecast possible price fluctuations, traders frequently examine the order book for anomalies, patterns, or imbalances. As an illustration,[135] a substantial accumulation of purchase orders at a particular price level might suggest robust support, whereas an abrupt upswing in sell orders might indicate resistance. In volatile and fast-paced markets, where immediate details can mean a substantial difference, DOM data is especially valuable. Market depth is a common tool utilized by algorithm traders, day traders, or institutional investors to optimize how they trade and implement orders more efficiently. Specific sophisticated trading platforms provide previous market depth analysis and present current market conditions. This functionality enables traders[136] to examine historical order book data, which assists them in recognizing recurring patterns and enhancing their predictive capabilities. In its entirety, depth of market is a potent instrument that provides traders with discernment into the complex strata of supply and demand, thereby facilitating better-informed decision-making and augmented risk management within the ever-changing realm of financial markets.

5.8 Bear and Bull

The financial media discourse surrounding bear and bull markets underscores their importance to investors and the economy. Inadequate academic research and the absence of precise definitions and consensus regarding turning points, on the other hand, contribute[137] to their ambiguity. In contrast to the established turning points of the business cycle, as reported by NBER, bear and bull markets do not share a comparable foundation. The lack of scholarly attention in previous decades can be attributed to the erroneous belief that stock market prices fluctuate randomly. The current study challenges this notion by proposing that changes in fear of risk or other aspects of the economy may cause expected returns to fluctuate over time. [11] They are

Figure 5.17: Bear and Bull

introducing an official definition of bear and bull markets and implementing it using two empirical turning point detection procedures. The definition centers on the consistent increase in share prices, drawing inspiration from techniques employed in business cycle analysis[138]. Bull markets are distinguished by two criteria: first, local peaks and lows, defined by sufficiently persistent gains; second, they are periods of returns that exceed the norm. The analysis examines the consistency and longevity of the bull and decline market phases over a two-hundred-year period, thereby shedding light on their enduring characteristics. Consistency among the two approaches is validated, and the outcomes shed light on bull and bear market characteristics. Consistency in a model of time-sensitive mean return parameter movements

[11] https://www.vectorstock.com/royalty-free-vector/bull-vs-bear-symbol-stock-market-trend-vector-34478761

is suggested by in-sample tests, which calls into question the validity of a random-walk framework with constant drift. The implications of the results extend to the development and evaluation of models based on capital asset pricing theory, underscoring the significance of share index turning points as early warning indicators for market timing. This paper enhances comprehension of the cyclical behavior of the stock market, thereby stimulating further investigation in this field[139]. [12] A bull market in the financial

Figure 5.18: Bear Market

markets is characterized by an extended duration of price appreciation and a positive investor outlook. A general sense of assurance exists regarding the market's upward trajectory during this stage, propelled by robust corporate earnings, positive economic indicators, and an overall optimistic economic forecast.[140] Bull market investors are likelier to purchase securities, contributing to increased demand, price appreciation, and market prosperity. A bear market, conversely, is characterized by a prolonged duration of declining prices and a negative outlook among investors. A bear market is characterized by pervasive apprehension regarding the state of the economy, which discourages the purchase of securities [141]on the expectation of further declines. Bear markets are characterized by diminished corporate earnings and unfavorable economic indicators, contributing to heightened selling pressure, reduced prices, and increased trading volumes as investors attempt to liquidate their positions. Investors rely heavily on the terms "bull marketplace" and "bear market" to assess market conditions and formulate well-informed decisions based on prevailing sentiments.

[12]https://www.investopedia.com/terms/b/bear-market-rally.asp

5.9 Advanced Insights into Spot, Margin, and Futures Trading in Cryptocurrency Markets

Spot trading is a rudimentary form of commerce in which assets, including cryptocurrencies, are exchanged for and against one another at the prevailing market value. Comparable to conventional commerce, it provides instant ownership. Nonetheless, this method excludes leverage, requiring traders to utilize solely their current assets without procuring additional funds. Moreover, the settlement of the entire transaction occurs in real time, establishing a palpable sense of ownership.

As an extension of Spot Trading, Spot Margin Trading presents the notion of leverage[142]. Traders can also borrow funds from the platform to augment their trading positions, facilitating the purchase or sale of assets more than their present holdings. Although ownership remains unchanged, the increased dependence on borrowed capital raises the possibility of liquidation, particularly if market fluctuations render the loan-to-value ratio unfavorable.

Futures trading is an intricate [143]financial instrument in which participants engage in contractual agreements constructed around an underlying asset, commonly cryptocurrencies. Futures contract purchases and sales, in contrast to spot trading, do not result in the transfer of physical ownership of the assets involved. Rather, they entail an obligation to purchase or sell the underlying asset at a pre-arranged price on a future date. Leverage is a crucial instrument that empowers traders to manage larger positions with a reduced margin. Futures contracts are differentiated from perpetual contracts by expiration dates, necessitating their settlement. These financial instruments function for speculative and risk-hedging objectives within the ever-changing cryptocurrency market, offering a flexible strategy for maneuvering through market fluctuations.

5.10 Technical Analysis

Technical analysis is an all-encompassing trading discipline that operates under the premise that market activity mirrors every piece of information accessible. Technical analysis, which Charles Dow and the Dow Theory initially formulated in the late 1800s, has since been substantially enhanced and broadened by eminent scholars, including John Magee, William P. Hamilton, Robert Rhea, and Edson Gould. The evolution serves to emphasize its lasting significance throughout the centuries.

Technical analysis is predicated on the comprehension of how the actions of market participants influence the prices of securities, as opposed to evaluating their intrinsic value. Traders operating in historical markets, such as those handling rice in Japan or corn in England, devised basic charting techniques like the contemporary "Point and figure" chart to evaluate the equilibrium between supply and demand and facilitate subsequent transactions.
Fundamentally, technical analysis examines volume trends, price fluctuations, and psychological determinants that impact participants in the market. It uses various instruments such as mathematical computations, statistical tools, and chart pattern analysis to interpret the dynamics of supply and demand. In contrast to fundamental analysis, which assesses the value of a security by considering economic factors, technical analysis is predicated that historical price fluctuations and trading activity provide valuable indications of forthcoming trends. Over the course of its development, the field has assimilated countless patterns and signals that have been extracted from thorough research. By examining the effect of supply and demand on price, volume, and implied volatility, technical analysis tools produce trading signals for the near future. In addition to forecasting price fluctuations, these tools assist analysts in assessing security's relative strength or vulnerability in the wider market. Technical analysis, which applies to a wide range of securities with a trading history encompassing futures, equities, commodities, fixed income, and currencies, gains considerable traction in the forex and commodities markets, where minute price fluctuations hold the utmost significance. Technical analysis continually updates its trading strategies to incorporate emerging signals and patterns as markets progress. This ensures that it remains a valuable tool for retail and professional investors. Due to its continuous adaptability, historical origins, and versatility, technical analysis is indispensable to market analysis and trading decisions.

5.10.1 Technical analysis Trends

Horizontal congestion with a Double Bottom pattern

A peak distinguishes Horizontal Congestion with a Double Bottom pattern consisting of two consecutive troughs. Support levels are typically formed by rounded or pointed troughs at or near the same price level. A consolidation phase is denoted by the fluctuation in price between the support level established by the troughs and the peak. The price must ultimately surpass the midpoint to validate the pattern, indicating the possibility of a trend reversal or continuation. Traders frequently seek after this pattern because it may indicate a change in market sentiment and present advantageous circumstances for strategic entry or exit points. [13]

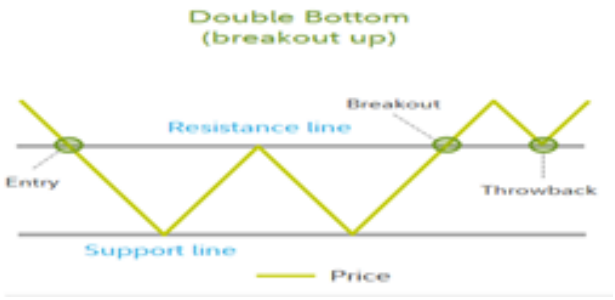


Figure 5.19: Horizontal congestion with a Double Bottom pattern

Horizontal Congestion with a Dual Top

Two consecutive peaks separated by an opposing reversal point comprise the "Horizontal Congestion with a Dual Top" technical analysis pattern. These rounded or pointed peaks are commonly encountered at or near the same price and constitute resistance. The pattern indicates a phase of consolidation or gridlock in price movement. To validate the pattern, prices must ultimately surpass the intermediate reversal point, which would signify the possibility of either a trend reversal or its continuation. By studying the reversal point and the formation of two sequential peaks, investors and traders can discern possible resistance areas and formulate well-informed decisions anticipating future price fluctuations.

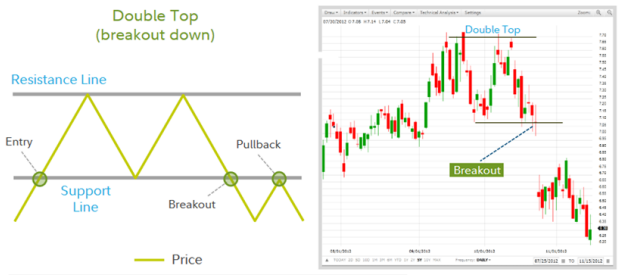


Figure 5.20: Horizontal Congestion with a Dual Top

Horizontal Congestion Triple Top

A technical analysis formation, the "Horizontal Congestion with a Triple Top" pattern is distinguished by two intermittent troughs that separate three distinct peaks at approximately the same price level. Traders frequently regard this pattern as a possible indication of a reversal in the price movement,

[13] $https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Idenitfying-Chart-Patterns.pdf$

as it signifies a phase of congestion or consolidation. A breakout occurs when the price surpasses the extreme of an intermittent trough or a trend line that connects those two points. The price surpassing a resistance level signifies a potential shift in market sentiment, which qualifies this breakout as significant. Examining the configuration of three consecutive peaks interspersed with sporadic troughs enables traders to discern critical levels and render well-informed judgments regarding prospective shifts in trends.

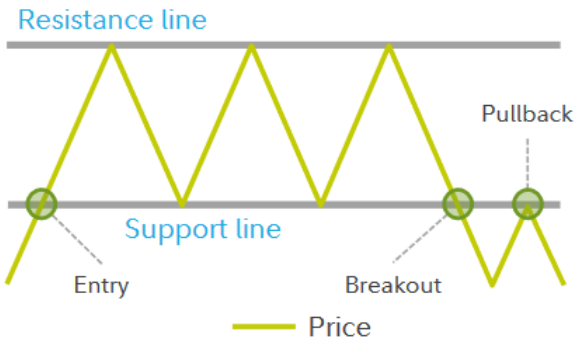


Figure 5.21: Horizontal Congestion Triple Top

Horizontal Congestion Triple Bottom

Three distinctive troughs at approximately the same price level, divided by intermittent summits at any level, define a Triple Bottom. The breakout is deemed confirmed when the price surpasses the highest or lowest point of the intermittent peaks or a line of trend that connects those points. Traders must closely monitor market conditions and assess potential failings before basing decisions on this formation, as its optimal performance may occur after a prolonged decline. Knowing the unique characteristics that define the Triple Bottom pattern empowers traders to discern possible reversals of trends and execute well-informed trading decisions.

Horizontal Congestion Rectangles

Price action is constrained by a trading range of support and resistance levels during a broadening formation. With a modest incline, it resembles a horizontal channel. Nevertheless, it frequently encounters numerous erroneous breakouts, requiring substantiation of a breakout before formulating trading strategies. The "shortfall," which may indicate the eventual breakout trajectory, ought to be considered by traders. This pattern might be most effectively executed by a bottom bursting upward. Analyzing broadening

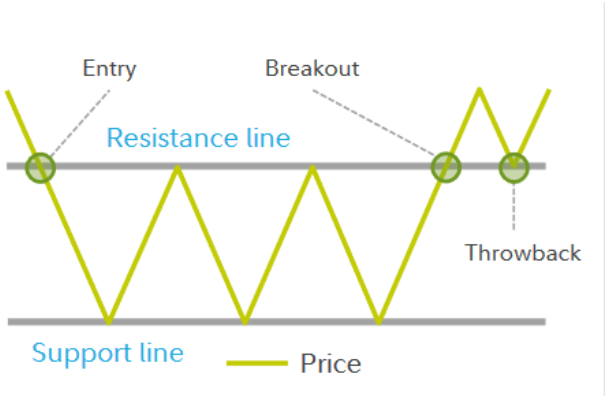


Figure 5.22: Horizontal Congestion Triple Bottom

formations entails evaluating these attributes to predict possible alterations in market dynamics. [14]

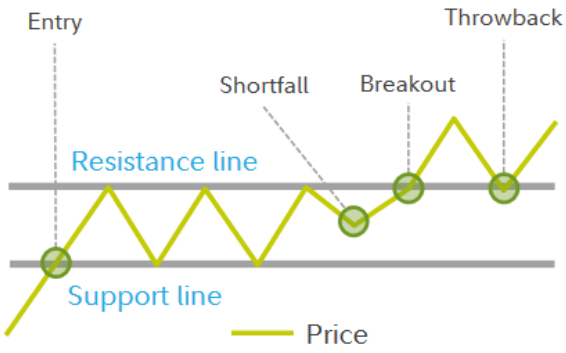


Figure 5.23: Horizontal Congestion Rectangles

Triangle Symmetrical

A sloping downward upper trend line or an upward-sloping lower trend line comprises the descending triangle pattern, which forms a triangle. At least two price touches are required at each boundary within this pattern, and numerous false breaches are possible. Although performance has been moderately successful, traders must verify a breakout before reaching any conclusions. As it is regarded as above average among all patterns, an upward breakout could be the most favorable outcome for this pattern. Traders frequently examine descending triangles in search of forward-looking price movement indicators. [15]

[14] $https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Idenitfying-Chart-Patterns.pdf$

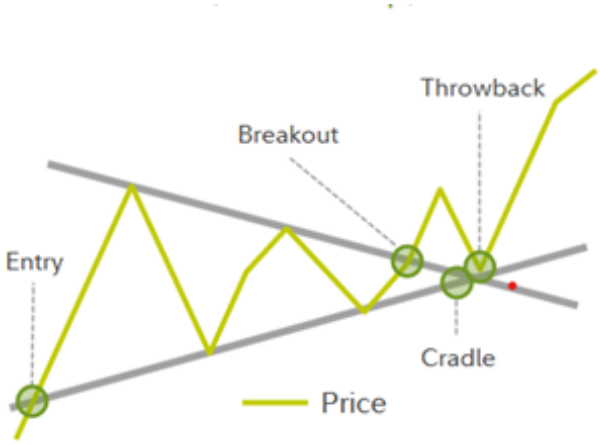


Figure 5.24: Triangle Symmetrical

Triangle Ascending

The ascending triangle pattern is distinguished by a lower trend line that slopes upwards and an upper trend line that is horizontal in orientation. The horizontal trend line functions as a support, and the ascending trend line functions as a resistance, forming a triangular configuration. Although prices may break in either direction during this pattern, an upward breakout is more frequent. Breakouts frequently transpire within the observed pattern, and although failure rates are approximately average, numerous minor false breakouts may transpire. On the upside, post-breakout performance is regarded as average, while on the downside, it is regarded as above average. Traders look for ascending triangles as possible indicators of forthcoming price fluctuations.

Triangle Descending

The descending pyramid pattern is delineated by two trend lines, wherein the lower line exhibits a horizontal inclination while the upper line descends. Typically, this pattern signifies a phase of market consolidation in which prices are contained within both trend lines. Although prices can break in either direction, downward breakouts are the norm. When an upside break occurs, the descending triangle typically demonstrates above-average performance despite the possibility of retracements. Traders frequently scrutinize this pattern due to its insights regarding possible price movements after a breakout. [16]

[15] $https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Identifying-Chart-Patterns.pdf$

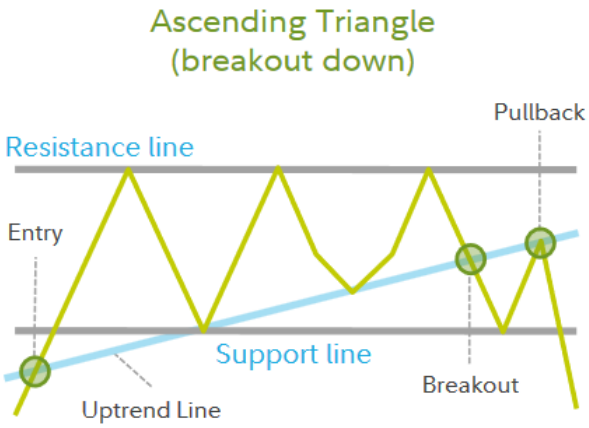


Figure 5.25: Triangle Ascending

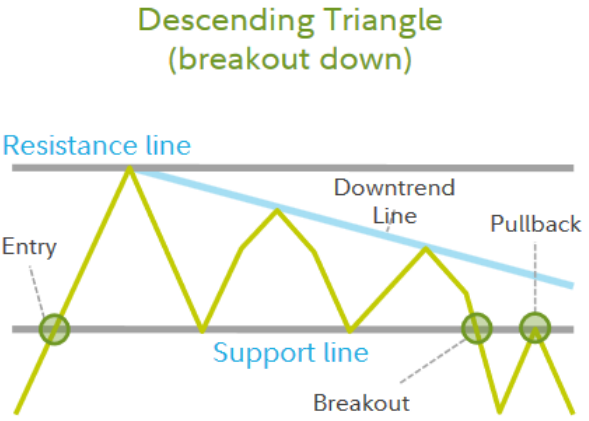


Figure 5.26: Triangle Descending

Head and Shoulders Top

The triple top pattern is distinguished by three peaks, the highest at the center. The cranium of the pattern is elevated, while the shoulders are roughly at the same level. A crucial component is the "neckline," denoted by a line that joins the two troughs between the peaks. Completeness of the pattern is indicated by a breach occurring below the neckline. Analysts frequently project targets for future price movement. These targets are calculated by extending the distance from the top of the head toward the neckline and then projecting that distance downwards from the neckline. Significantly, this triple peak pattern is renowned for its comparatively low failure rates, rendering it a conventional and dependable pattern to discern prospective market peaks.

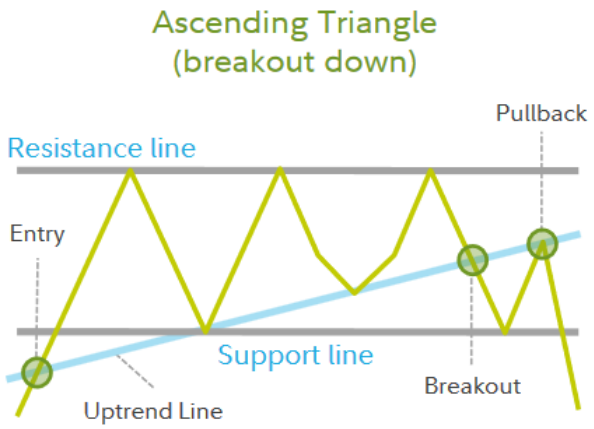


Figure 5.27: Head and Shoulders Top

Head and Shoulders Top

The triple top pattern is distinguished by three peaks, the highest at the center. The cranium of the pattern is elevated, while the shoulders are roughly at the same level. A crucial component is the "neckline," denoted by a line that joins the two troughs between the peaks. Completeness of the pattern is indicated by a breach occurring below the neckline. Analysts frequently project targets for future price movement. These targets are calculated by extending the distance from the top of the head toward the neckline and then projecting that distance downwards from the neckline. Significantly, this triple peak pattern is renowned for its comparatively low failure rates,

[16] $https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Identifying-Chart-Patterns.pdf$

rendering it a conventional and dependable pattern to discern prospective market peaks.

Figure 5.28: Head and Shoulders Top

Head and Shoulders Bottom (Inverse)

The examined pattern exhibits an inverted structure similar to a top pattern while retaining its overall characteristics. Nevertheless, notwithstanding the resemblances, this inverted pattern generally yields lower profits than its counterpart, the top pattern. These unique characteristics are considered by traders and analysts when evaluating market conditions and formulating decisions predicated on observed patterns. [17]

Cup and Handle

The subject pattern is distinguished by its rounded bottom (as opposed to a "V" bottom), two conspicuous "lips" situated at each end, and a unique "handle" that bears resemblance to a flag pattern and extends from the base. The pattern is concluded with an eruption occurring above both lips. Notable is the fact that these patterns frequently undergo a recall, during which the price re-enters the area of the outbreak. This pattern's overall performance is on par with the norm for bottom patterns. Traders and analysts meticulously scrutinize these elements to assess possible price fluctuations and formulate well-informed decisions predicated on the discerned pattern [18]

[17] $https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Idenitfying-Chart-Patterns.pdf$

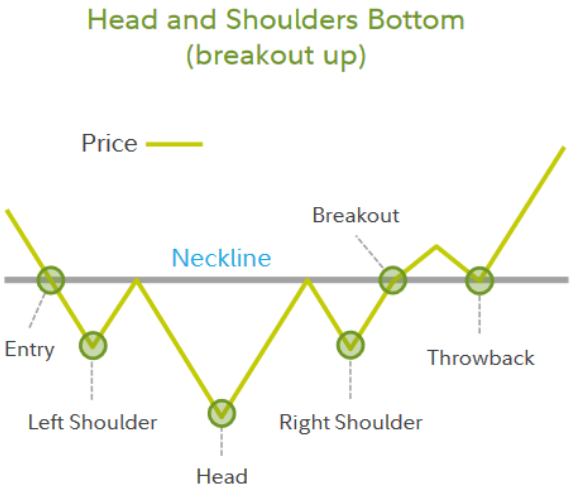


Figure 5.29: Head and Shoulders Bottom (Inverse)

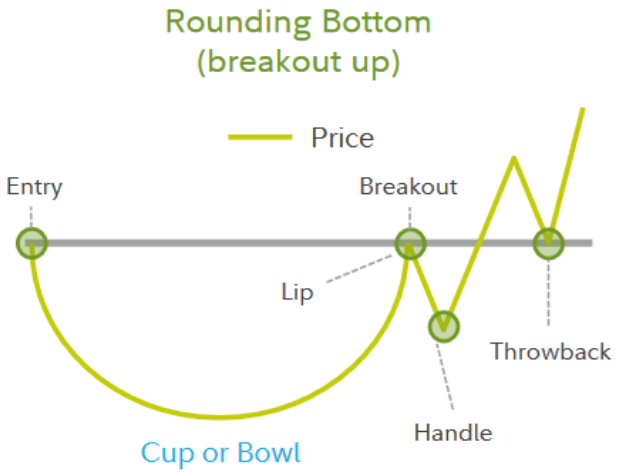


Figure 5.30: Cup and Handle

5.11 Support and Resistance

Technical analysis is predicated on the foundational principles of support and resistance, which serve as the skeleton for perceptive price chart interpretation. These concepts are predicated on the intricate interplay between supply and demand, with prices reacting to changes in these conditions. In essence, prices increase when supply exceeds demand and decrease when demand is insufficient supply. Sideways movement signifies a balanced supply and demand state, providing a more intricate perspective on market sentiment.
While not an exact science, technical analysis is a navigational tool for traders striving to make well-informed decisions within the financial domain. A crucial principle identified in downtrends is that support becomes tangible when prices decline due to an oversupply. A reduction in prices attracts prospective purchasers, thereby stimulating heightened demand. When supply and demand converge to a particular threshold, a support level is formed, halting prices' downward momentum. This support may take the form of a distinct price level or a wider zone, signifying a region where purchasers are enthusiastic about engaging. The ability to distinguish these levels gives traders the anticipation necessary to forecast possible reversals in the current price trend. On the contrary, resistance functions as the antithesis of support. As a result of increased demand, ascending price movements encounter an inflection point at which selling pressure exceeds buying interest. The graphical representation of this resistance level on price charts indicates the critical point at which supply surpasses demand. Resistance, akin to support, may take the form of a distinct level or a more expansive zone. Traders must possess the critical ability to recognize and understand these levels, as they function as strategic entry or departure points in their trading deliberations. By utilizing support and resistance zones, traders can effectively anticipate and navigate price fluctuations. The reaction of prices to these levels provides traders with information regarding possible continuations or reversals of trends. Returning from these levels would indicate a likely continuance of the current trajectory, whereas a breach would indicate a reversal and potential financial losses for speculators. Technical analysis, which integrates scientific and artistic principles, entails a rigorous evaluation of past support and resistance levels to extract predictions regarding potential future market responses. The adaptability of support and resistance is demonstrated across various periods, including daily, weekly, and monthly charts. The importance of support or resistance is frequently emphasized over extended time periods. Traders astutely discern past levels and predict the reverberation of analogous responses in subsequent periods. Although not flawless,

[18] $https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Idenitfying-Chart-Patterns.pdf$

this methodology takes advantage of the combined behaviors of market participants, thereby converting these levels into dynamic zones that impact trading choices.
When used in conjunction with static barriers, trendlines are essential elements in the analysis of support and resistance. As prices approach the trendline during an uptrend, support is identified as a barrier against substantial declines. On the contrary, traders identify declining peaks during a downtrend and establish a connection between them and a trendline, expecting resistance as prices near this line. A minimum of three touches is necessary to affirm a trendline as legitimate. Support and resistance gain greater significance as the frequency of historical rebounds from these levels increases. Traders employ these levels strategically as entrance and exit points, recognizing the significant influence they have on the course of an asset. Although not without difficulties, technical analysis is an essential instrument for traders to navigate the complex landscape of financial markets. Proficiency in these principles necessitates years of diligent application, emphasizing the critical significance of experience in efficiently capitalizing on support and resistance in all-encompassing trading strategies.

5.12 Fibonacci series

Honoring the Italian mathematician Leonardo Pisano Bogollo, who was subsequently referred to as Fibonacci, the Fibonacci series (sum) consists of the Fibonacci numbers represented as Fn. The Fibonacci series numbers are 0, 1, 1, 2, 3, and 5,... All terms in a Fibonacci series are calculated as the sum of the two terms that came before it, with the first and second terms being 0 and 1, respectively. There are older[144] references that may omit the letter '0'. Mathematicians remain captivated by the series, and its captivating properties continue to be the subject of investigation and study. Natural phenomena exhibit implementations of the Fibonacci series. It is present in biological environments, such as tree branching and flower petal patterns. The following sections will examine the Fibonacci series formula, properties, and applications. The sequence of numbers called the Fibonacci series (or Fibonacci numbers) consists of each number being the product of the two antecedent numbers, with the initial two elements consisting[145] of the digits '0' and '1'. There may be prior iterations of the series in which the letter '0' is omitted. Thus, the following constitutes a Fibonacci series: 0, 1, 1, 2, 3, 5, 8, 11, ... Thus, it is evident that each term can be determined through the process of aggregating the two terms preceding it. Using the values '0' and '1' for the first and second terms, F0 and F1, respectively, the third term F2 = 0 + 1 = 1. In a similar vein,

$$F3 = 1 + 1 = 2$$
$$F4 = 2 + 1 = 3$$
$$F5 = 2 + 3 = 5$$
$$F6 = 3 + 5 = 8$$
$$F7 = 5 \text{ plus } 8 \text{ equals } 13, \text{ plus}$$

Hence, the corresponding expression is for any term (n+1) in this series. Consequently, a Fibonacci series can be visually depicted in the subsequent image. In mathematics, the formula for the Fibonacci series can be utilized to locate the absent elements within the Fibonacci series. Utilizing the recursive formula, the (n+1) element in the sequence can be determined as , where = 0 and = 1.

To create the Fibonacci series spiral, a logarithmic spiral, the corners of squares whose side lengths correspond to the Fibonacci numbers in the sequence are joined. The spiral is observed in various natural phenomena, including the leaf arrangement on a stem, the carapace of a nautilus, and the spiral arms of galaxies, among others. The Fibonacci series spiral, renowned for its symmetrical and aesthetically appealing appearance, has been the subject of extensive mathematical study. The subsequent rectangle featuring the spiral of the Fibonacci series is a golden rectangle. Thus, the "Golden Ratio" characterizes its dimensions.

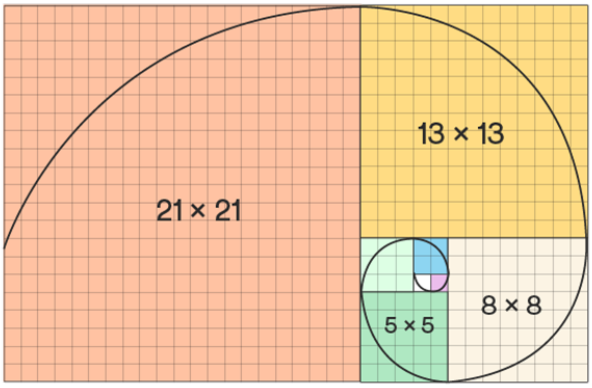


Figure 5.31: Fibonacci series spiral

[19] Fibonacci retracement levels, which are horizontal lines denoting possible support and resistance areas, are derived from the Fibonacci sequence[146]. On a chart, these percentage-based levels—23.6%, 38.2%, 50%, 61.8%, and 78.6%—are delineated between two significant price factors. The identification of Fibonacci retracement levels is facilitated by their static nature, which enables traders to anticipate and react to price levels

[19]https://www.cuemath.com/numbers/fibonacci-series/

effectively. Traders employ Fibonacci retracement levels for many objectives, including but not limited to establishing price targets, placing entry orders, and determining stop-loss levels. In addition to these levels, other technical analysis techniques, such as Elliott Wave theory and Gartley patterns, are incorporated. Nevertheless, discerning which of the numerous Fibonacci retracement levels will exert the most significant influence at any given moment presents a formidable obstacle. In technical analysis, Fibonacci retracements are indispensable because they mark critical regions where a stock may reverse or halt. Mathematically derived from the Fibonacci sequence, these ratios (23.6 percent, 38.1 percent, 50 percent, 61.8 percent, and 78.6 percent) provide traders with [147]insights into potential price movements. Using Fibonacci retracement levels entails delineating trend lines between pivotal data points on a chart, enhancing the informed Ness of trading strategy decisions. Notwithstanding the unpredictability of price responses at such levels, integrating Fibonacci retracements alongside additional indicators augments the precision of trend evaluations and trading judgments.

The Fibonacci sequence and its corresponding ratios hold considerable importance in the domain of technical analysis as it pertains to financial markets. Traders frequently employ Fibonacci tools to discern critical levels of resistance, support, and prospective market reversals. The following ratios result from the sequence, which begins with 0 and 1 and proceeds with each number being the sum of its two preceding numbers: 23.6%, 38.2%, 50%, 61.8%, and 78.6%. Based on these ratios, numerous Fibonacci applications are constructed. In price charts, Fibonacci retracement levels link pivotal moments, including market lows and highs. The following levels—23.6%, 38.2%, 50%, 61.8%, and 78.6% — signify potential support or resistance zones for the price of an asset, thereby presenting opportunities for reversals or consolidation. Furthermore, when applied to substantial moves in the direction of the trend, Fibonacci extensions forecast forthcoming price levels; typical extension levels consist of 161.8%, 261.8%, and 423.6%.

Traders also utilize Fibonacci ratios to determine critical levels of support and resistance. The 61.8% threshold, commonly known as the "Golden Ratio," possesses notable importance as a prospective point of reversal. Fibonacci fans and arcs serve the purpose of identifying trendlines and drawing attention to possible regions of reversal. Time zones, an additional Fibonacci instrument, are vertically applied to price charts to indicate potential periods of trend change for the market according to intervals derived from Fibonacci. In addition, Fibonacci ratios play a crucial role in Elliott Wave Theory, a technical analysis approach that discerns discrete waves within market fluctuations. The recurrent adherence of these waves to Fibonacci relationships enhances the predictive capability of the theory. Although not without its limitations, utilizing Fibonacci tools in financial markets grants

analysts and traders significant knowledge regarding possible price levels, trends, and junctures of reversal, thereby enhancing the repertoire of technical analysis methodologies.

5.13 Order Book

An order book is an essential element within financial markets; it denotes a scrupulously organized electronic compendium of orders to buy or sell a particular security or investment vehicle[148], categorized by price level. As this configuration is commonly known, market depth depicts the number of shares being offered or bid on at each price level. Additionally, the identities of the market players who placed these orders are revealed, although some may choose to remain anonymous. Order books enhance market transparency by providing traders with crucial trading information that facilitates formulating well-informed decisions.

Figure 5.32: Order Book

Order books are widely employed on many exchanges, covering many assets, including equities, bonds, currencies, securities, and cryptocurrencies such as Bitcoin. Although the information they contain is generally identical in both manual and electronic formats, the way the orders are presented[148] may differ depending on the source. The purchase and sell information could be displayed on the screen in a left-and-right or top-and-bottom orientation. A dynamic and perpetually updated real-time instrument, an order book functions as such throughout the course of the trading day. It is referred to as the "continuous book" by exchanges like Nasdaq; for orders requiring execution at market open or closure, distinct records are kept and are called the "opening (order) book" and "closing (order) book," respectively[149]. At

the opening and closing of the Nasdaq market, these accounts are consolidated to obtain a singular opening and closing price. Generally, an order book comprises three essential elements: order history, purchase orders, and sell orders. Purchase orders exhibit pertinent consumer details, such as the quantities they intend to purchase for each proposal. Similar to purchase orders, sell orders include all asking prices or proposals at which individuals are willing to sell. The market order history provides a comprehensive log of all previous transactions. The highest offer and lowest ask prices are displayed at the head of the book, providing traders with an indication and analysis of the current market conditions and the prices that are essential for carrying out orders[150]. Traders utilize the information contained in the order book to improve their decision-making. Through an analysis of the brokerages involved in the purchase and sale of equities, one can ascertain whether institutional or retail investors are the primary drivers of market fluctuations. In addition to disclosing order imbalances, the order book offers immediate insights into the potential trajectory of a given stock. For example, a substantial imbalance in favor of buy orders could indicate that purchasing pressure exerts upward momentum. Although the order book strives to ensure transparency, specific components, such as "dark pools," remain undisclosed. These are hidden bundles of orders maintained by major market participants who wish to conceal their trading intentions. As price devaluation is typically the result of public disclosure of significant transactions before execution, dark pools aid in maintaining price stability. While the existence of dark pools diminishes the complete transparency of the order book, it emphasizes the intricate nature of market dynamics.

5.14 Wyckoff Theory

Traders may adopt a variety of trading strategies; however, only some of them are effective. Indeed, many of these methodologies have demonstrated their ineffectiveness as time has passed. However, Wyckoff's Theory is one of those schools of philosophy that has withstood the test of time. Wyckoff Theory was established in the 1930s by Richard Wyckoff, and merchants continue to employ it to examine the financial markets at present[151]. A set of principles known as Wyckoff Theory enables traders to comprehend institutional actors. It enables traders to identify Smart Money's footprints and trade with those with the greatest market influence. By doing so, the trader's profitability and accuracy are enhanced. Richard D. Wyckoff emerged as an early adopter of a technical framework within financial markets. He occupied a prominent position within the realm of technical analysis. Wyckoff endeavored to discern latent patterns or rationales that underpin bond, commodity, and stock market fluctuations. Wyckoff obtained visibility into the machina-

tions executed by Smart Money. He developed the "Richard Wyckoff Theory of Acquisition and Distribution," which he called "The Composite Man" at the time, based on his expertise in the financial markets. Wyckoff's research revealed that several prosperous market participants and high-performing equities of that era possessed comparable characteristics. He examined the "trading range," the optimal price range for purchasing or selling a stock, in detail. He discovered through additional research that it was feasible to forecast forthcoming price fluctuations. Per Wyckoff, institutional investors and Smart Money operators construct their orders within these trading ranges. Nevertheless, their strategies are manifested in prices through the imprint they leave, which can be utilized to forecast future prices by analyzing these trading ranges. Wyckoff observed that these manipulations consistently uncovered retail dealers. After that, he devoted himself to educating the public about the "Wyckoff Theory" that evolved from the "Rules of the Game" utilized by major institutions. Collectively, Wyckoff referred to these institutions or major actors as the "Composite Man." Wyckoff proposed the notion of the "Composite Man" to delineate how an individual operating in the background could manipulate a trading asset for the greater good. However, it is possible to characterize the Composite Man as the Central Bank, given its authority over the valuation of every traded asset. The Central Bank's price manipulation can result in losses for merchants. However, those who comprehend the Composite Man's strategy and liquidity can profitably remain in the game. (Within the realm of cryptocurrencies, the Composite Man is frequently referred to as a "crypto whale.") In general, the Composite Man purchases in consolidation before price movement, keeping in mind the market. Wyckoff referred to this phase as the Accumulation Stage. It is the accumulation of orders by major institutional merchants before the occurrence of any price movement. Therefore, traders must analyze prices during these accumulation phases to identify optimal entry opportunities. The optimal time to exit would be after a price movement, as the Composite Man aims to exit by aligning their orders with those of other merchants inclined to purchase at elevated prices. Wyckoff posited that by comprehending the Motivations and Conduct of the Composite Man, it would be possible to foresee numerous lucrative trading opportunities in advance[151]. To analyze schematics utilizing Wyckoff's Theory, a trader must possess knowledge of three laws. These regulations offer a comprehensive outline of the cost, aiding the merchant in locating it. In which direction is the pricing inclined to move? "A direction that is most likely to occur." Which asset is the most probable mover? "The asset which is most susceptible to a movement." At what time is it optimal to exchange it? "During the most lucrative time to trade."

5.14.1 The Law of Demand and Supply

This determines the trajectory of prices; it is a fundamental tenet of Wyckoff's theory. Whenever demand exceeds supply, prices tend to increase because of the increased volume of purchases. Price decreases are a common consequence of increased selling activity of a dealing asset when supply exceeds demand. The condition of "Equilibrium" arises when the interplay between supply and demand maintains asset prices in a consolidated range within a narrow distance. When either demand or supply changes, an imbalance ensues, resulting in a price movement accompanied by significant momentum. If you are interested in learning about supply and demand, consult the article " Demand and Supply In the foreign exchange market: Strategies to 10X Your Results.

5.14.2 The Law of Causes and Effect: Distribution and Accumulation

It is generally acknowledged that changes cannot occur spontaneously but require an established cause. This cause is typically established by transitioning from uninformed to knowledgeable Smart Money. This commences during sideways movement or as the market is in a ranging phase. This phenomenon becomes apparent on charts during periods of narrow ranges or minimal volatility. As the duration of this process prolongs, the anticipated resultant transfer effect grows in magnitude. A period for consolidation is followed by an expansion and explosive price movement characterized by large ranges. Wyckoff designates these narrow intervals as the "Accumulation Phase" (Cause), during which the price would experience an upward movement (Effect) and establish a bullish trend, or the "Distribution Phase" (Cause), during which the price would experience a downward movement (Effect) and develop a bearish trend. To identify occasions for sideways movement or trade range causes and project the most probable target areas (Effect), Wyckoff himself employed point-and-figure charts. On contemporary markets, however, traders favor employing instruments for target projection such as Fibonacci, Elliott Waves, Harmonic Patterns, or Demand and Supply Zones.
[20]

5.14.3 The Law of Results and Effort

This law signals a potential change in trend that may occur shortly. The "Effort" or quantity of orders exchanged in support of a specific price movement

[20]https://citytradersimperium.com/wyckoff-theory-forex

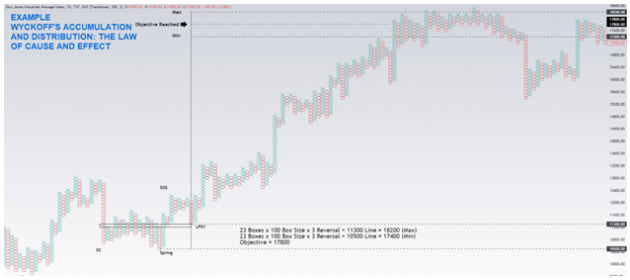


Figure 5.33: The Law of Causes and Effect

is denoted by volume. The volume determines the "Result" of the price movement. The interest of major institutions in each asset will manifest through a volume fluctuation. Frequently, the divergence between volume and price fluctuations serves as an early indicator of a trend reversal. However, bear in mind that since stocks and futures are traded on centralized exchanges, volume divergence with pricing applies to these instruments. Every transaction that takes place traverses a Central Exchange. The volume measurements are, therefore, precise. This law is more difficult to ascertain in Forex than in Wyckoff due to the decentralized nature of the trading activity; every trading system represents the trade volume of a specific broker rather than the entire industry. However, we have identified additional methods of divergence that may be of assistance to you with this concept. Refer to our article on Forex divergence for further details. The initial step is to utilize Wyckoff's Laws of Analyses to survey the financial markets to find the assets most likely to be offered in profitable transactions. Try to comprehend the state of liquidity, the game in which the Composite Man is engaged, and his probable next course of action.

Figure 5.34: The Law of Results and Effort

[21]

5.14.4 Market Cycle Theory of Wyckoff

By applying his knowledge of demand and supply, Wyckoff defined markets and demonstrated that they operate in a perpetual cycle consisting of only

[21] https://citytradersimperium.com/wyckoff-theory-forex

four phases at any given time. Traders and investors can surmise the market's probable trajectory by comprehending its current position in this cycle.

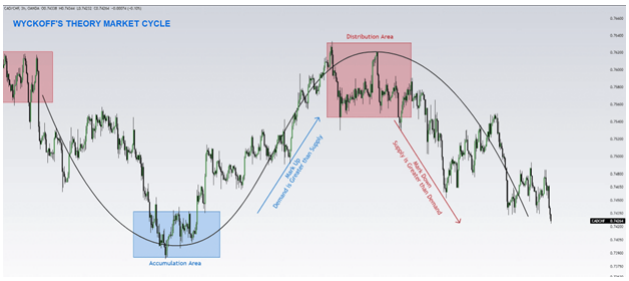


Figure 5.35: Market Cycle Theory of Wyckoff

1. Phase of Accumulation
2. Markup
3. Phase of Distribution
4. Markdown

By conducting additional research in accumulation and distribution, the buyer or seller would be well-positioned to enter into a harmonious transaction with the composite man. Accumulation and Distribution Zones, which are trading ranges, are regions in which supply and demand reach equilibrium. In anticipation of the subsequent markup or markdown, major institutional actors either accumulate or distribute their positions in buy or sell areas, respectively.

5.14.5 The Wyckoff Schematics company

By conducting an in-depth analysis of these trade ranges, Wyckoff acquired the knowledge and skills necessary to discern the underlying storyline of each range and thus improve his ability to forecast future events. The Wyckoff Schematics were initially categorized as Accumulation Schemas or Distribution Schemas, and subsequently, each schema is subdivided into phases, with distinct events occurring within each phase. Although the fundamental schematics provided do not encompass every conceivable variation that a range of trades might exhibit, the underlying principles remain applicable. Therefore, phases and occurrences must be described conceptually. [22]

[22]https://citytradersimperium.com/wyckoff-theory-forex

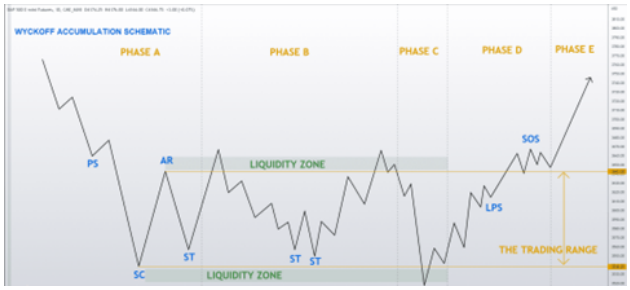


Figure 5.36: The Wyckoff Schematics company

In the Accumulation Phases

Phase A signifies the conclusion of a declining trend: Until now, supply has exceeded demand, and prices are declining. A surge in trading volume typically accompanies this phase. Supply depletion is indicated by the implementation of Preliminary Support (PS). Then, a period of high volatility and intensive selling activity due to panic selling constitutes a Selling Climax (SC). After selling pressure has been worn out, the Automated Rally (AR) ensues in the form of a rebound, which signifies the closure of short positions and a surge in institutional demand. The Secondary Test (ST), which typically consists of a narrowing spread and lower volume, is conducted to assess the downside. Initial trading range boundaries are established by the Selling Climax (SC) lows and the Automated Rally (AR) highs. By establishing boundaries in these regions, the trader can concentrate more intently on the market's behavior near those areas of interest.

Phase B – Cause Development: As mentioned, the legal concept of effect and cause has been examined in this context. We are laying the groundwork for initiating a new rise during phase B. Traders representing institutions or the composite man amass positions at a discount, expecting the price to increase during the markup phase. During this phase, the price remains in reorganization, but it may attempt to breach the trading range and generate phony breakouts. Multiple secondary tests (ST) may also be administered during phase B. The objective of the Composite Man is to obtain the maximum amount of available liquidity. As phase B progresses, the magnitude of the fluctuations diminishes and is accompanied by a reduction in volume. When the entire supply of floating liquidity is consumed, there is a high probability that a sufficient cause exists, and the price is prepared to transition into Phase C.

In Phase C, spring arrives: During phase C, the property is subjected to a supply assessment procedure. "Spring" refers to when the price accumu-

lates any remaining liquidity listed below the Offering Climax (SC). Traders are typically stopped out (via "Fake Breakout") by the spring before any genuine upward movement. The spring-like structure in the Accumulation Schematic is frequently favored over other diagrams due to its greater clarity regarding the residual supply's complete absorption and the run-on liquidity's conclusion. Nevertheless, in instances where the trend is robust, the Accumulation Schematic remains applicable even in the absence of spring.

Phase D: Analysis Confirmation: The transition from Phase C (Cause) to Phase E (Effect) occurs during this phase. A consistent rise in demand, as evidenced by a rise in trade volume and volatility, ought to ensue. Volatility, as well as trading volume increases, are clear indications that institutions have indeed intervened. The price encounters one or more elevated troughs at the Last Point of Support (LPS). Typically, the LPS occurs before the actual outbreak. The LPS may occasionally undergo a brief consolidation before a breakthrough. Subsequently, The Signs of Strength (SOSs) manifest as voluminous momentum flames. The price will at least ascend to the upper limit of the trading range throughout Phase D.

Figure 5.37: Analysis Confirmation

E. Departure from the Trading Range: The last stage in the Accumulation Schematic is denoted as Phase E. Demand is in complete control, the price exits the trading range, and it is evident that the markup has begun. Pullbacks are typically transient in nature.

5.14.6 Wycoff Distribution Schematic

Phase A of distribution signifies the conclusion of an uptrend. Contrary to accumulation, phase A denotes the waning of the uptrend in a distribution trading range. Additionally, there is an increase in transaction volume during this phase. Up until this juncture, price increases have been driven by demand. However, the Buying Climax (BC) and Preliminary Supply (PSY)

indicate that supply is now entering the market. After purchasing pressure has been depleted, the Automatic Rally (AR) manifests as a rebound, signifying the closure of buy positions and an increase in institutional supply. The Secondary Test (ST), which typically consists of a narrowing spread and lower volume, is conducted to assess the upside. The Buying Climax (SC) high and the Automatic Rally (AR) low establish the trading range's initial boundaries. By drawing lines in these regions, the trader can concentrate more intently on the market's behavior near these areas of interest.

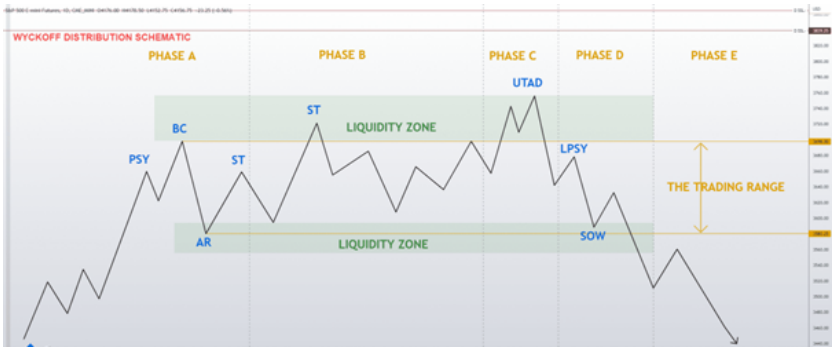


Figure 5.38: Wycoff Distribution Schematic

[23]

Phase B – Cause Development:

As with the previous accumulation, the objective of Phase B is to construct a cause; nevertheless, it is in anticipation of a new downtrend. We are laying the groundwork for initiating a new downtrend during phase B. In the Markdown Phase, institutional traders, the composite man, or traders who have sold their long positions, disseminate their holdings at a premium in anticipation of a price decline. The price remains consolidated during this phase, but it may attempt to breach the trading range and generate phony breakouts. Additionally, multiple secondary tests (ST) may be administered during phase B. Phase B serves the identical objective to accumulation and distribution, with the additional intention of maximizing the absorption of the remaining demand. The objective of the Composite Man is to obtain the maximum amount of available liquidity. As phase B progresses, the magnitude of the fluctuations diminishes and is accompanied by a reduction in volume. When the floating supply of liquidity completely absorbs the floating demand, a sufficient cause has probably existed, and the price is prepared to transition into Phase C.

[23]https://citytradersimperium.com/wyckoff-theory-forex

Phase C: Evaluation:

During phase C, the asset is subjected to a demand assessment procedure. "Upthrust" refers to the residual liquidity the price accumulates above the Buying Climax (SC). Traders are typically halted by the upthrust (Fake Breakout) before the occurrence of a genuine decline. As it is more evident that the run-on liquidity has concluded and the remaining demand has been completely absorbed, the upthrust in the distribution schema is frequently favored. However, if the trend is feeble, the upthrust may not always occur; however, the Distribution Schematic.

Figure 5.39: Phase C: Evaluation

[24]

Phase D: Confirms Analysis indicates:

The transition from Phase C (Cause) to Phase E (Effect) occurs during this phase. A consistent increase in supply, as indicated by a rise in trading volume and volatility, ought to ensue. Volatility and trading volume increases are clear indications that institutions have indeed intervened. At the Last Point of Supply (LPSY), the price experiences one or more successive peak points. Typically, the LPSY occurs before the actual outbreak. The LPSY may occasionally undergo a brief consolidation before a breakthrough. Subsequently, The Signs of Strength (SOW) manifest as voluminous momentum flames. The price will, at minimum, descend to the bottom of the trading range during Phase D.

E. Departure from the Trading Range:

The final phase of the Distribution Schematic is denoted as Phase E. The price exits the trading range, the supply is completely under control, and

[24]https://citytradersimperium.com/wyckoff-theory-forex

the markdown has begun inexplicably. Pullbacks are typically transient in nature.

5.14.7 Determining the Finest Trades

In addition to the Fundamental Laws of Analysis, Wyckoff devised a five-step procedure for determining the most profitable transactions. By fulfilling the subsequent five stages, this checklist enhances the likelihood and financial gain of any position. Determine the current market environment and the most likely future trend as the initial step. Is the market trending or consolidating? Does your analysis of supply and demand align with this perspective? What is the most probable trajectory that lies ahead? When the supply exceeds its demand, prices will decrease; conversely, prices will increase when the supply falls short of the request. Comprehending supply and demand enables traders to make informed decisions regarding whether to pursue long or short positions or abstain from trading the asset entirely. Select the most outstanding among related assets in the second step. Select assets that are correlated to validate your analysis. For instance, if the EURUSD analysis concludes with the indication that the price is in the Accumulation Phase. Subsequently, it would be best to examine additional GBPUSD and DXY pairs to validate your EURUSD analysis. EURUSD and GBPUSD typically fluctuate in the contrary direction of DXY. EURUSD has probably entered a genuine accumulation phase, given that GBPUSD is presently in the distribution phase, and the DXY is also in the accumulation phase. As previously explained, this indicates a discrepancy between the quantity sold and the price. Select an entry whose cause is at least equivalent to your objective in the fourth step. Historically, this stage necessitated a trader's knowledge of point-and-figure charts to discern targets. Nonetheless, target projections are feasible with an awareness of the market narrative over a longer time and regions of high liquidity. Determine whether a property is ready to be moved in Step 5. In this phase of the Wyckoff Theory, a nine-step enumeration is utilized to ascertain whether an asset's price is poised for an upward or downward movement. These tests determine whether the market has witnessed an influx of sufficient supply or demand to justify establishing an unsustainable short or long-term position. Additionally, it identifies whether the money has been completely absorbed. The Nine Selling and Buying Tests have been more detailed in adhering to principles that guide trade entry. The tests determine when markup and markdown are imminent, and the trading range is approaching its conclusion. The trader can determine the shortest distance path with the aid of the nine evaluations. [25]

[25] https://citytradersimperium.com/wyckoff-theory-forex

Figure 5.40: The Law of Causes and Effect

Events of Wyckoff Accumulation

1. The price target for the downside has been attained.
2. The Basic Support, Sell Climax, and Secondary Test are all present.
3. Rising volumes characterize bullish price movement during rallies and falling volumes during reactions.
4. The downward substructure has been breached (the downtrend or supply line has penetrated).
5. The existence of spring.
6. The price is experiencing increasing lows and highs.
7. The asset is outperforming the market (during rallies, the price exhibits greater volatility, whereas during retracements, it moves more slowly).
8. The formation of the Accumulation Trading Range (Base).
9. Should the initial stop-loss be triggered, the projected upward potential target would be at least three times the loss.

Distribution Events by Wyckoff

1. The price target to the upside has been accomplished.
2. The Basic Supply, Sell Climax, and Supplemental Test are all present.
3. Bearish in nature, the price movement is characterized by volume increases during declines and decreases during reactions.
4. The upward structure has been breached (the uptrend or support line penetrated).
5. The upthrust is in the vicinity.

6. The price is experiencing decreasing highs and lows.

7. The asset is less robust than the market, as evidenced by its price fluctuating more rapidly during declines and more slowly during retracements.

Assemblage of the Distribution Trade Range (Base)

Should the initial stop-loss be triggered, the projected downside target would be at least three times the magnitude of the loss.

Redistribution and Re accumulating

Re accumulating as Distributing and Redistribution as Accumulating are frequently misconstrued as accumulation and redistribution; accumulating functions similarly to distribution. An essential distinction is that accumulation halts a downward trend. Re accumulating occurs when a sustained upward movement continues, enabling the major participants to establish additional positions. Conversely, it applies to redistribution. Contrary to the conclusion of an ascending trend observed in Distribution, Redistribution persists as an extended downward trend. Although these market fluctuations resemble accumulation and distribution. They frequently need clarification with accumulation or distribution by traders. Gaining comprehension of the phases and the current price position of the market facilitates the trader's ability to discern between trend-ending and trend-continuation. Wyckoff Diagrams. As an illustration, following an ascending trend, a re accumulating commences, accompanied by a Selling Climax (SC) and Automated Reaction (AR), which momentarily halts the upward trajectory. Traders erroneously equate re accumulating with accumulation at this juncture.

5.15 Elliott Wave

In the 1930s, Ralph Nelson Elliott formulated the Elliott Wave Theory. He believed that stock markets, commonly perceived to exhibit a degree of disorder and capriciousness, followed repetitive trading patterns. Elliott identified fractal patterns in the movement of stock prices that he termed waves. Decades have passed since Elliott first proposed his theory. Elliott postulated that the prevailing psychology of investors dictates financial price trends. It was discovered that fluctuations in financial markets invariably manifested as recurring fractal patterns, or "waves," in mass psychology. Elliott's and Dow's theories are similar in that they both posit that stock prices exhibit cyclical patterns. However, by also acknowledging the "fractal" character of markets, Elliott could dissect and scrutinize them with considerably more

precision. Fractals are mathematical shapes that replicate themselves infinitely on an ever-smaller scale. Elliott saw that stock index pricing trends exhibited a consistent structure. Following this, he investigated the potential of these recurring patterns to serve as accurate indicators of forthcoming market movements[152]. Ralph Nelson Elliot identified the phenomenon he termed the "wave principle." It has been observed that social or mass behavior exhibits discernible patterns of trending and reversing. By employing stock market data as one of his research instruments, Elliott was able to discern thirteen recurring patterns or "waves" within the market for stocks data [26] He designated, elucidated, and depicted those patterns. The notion

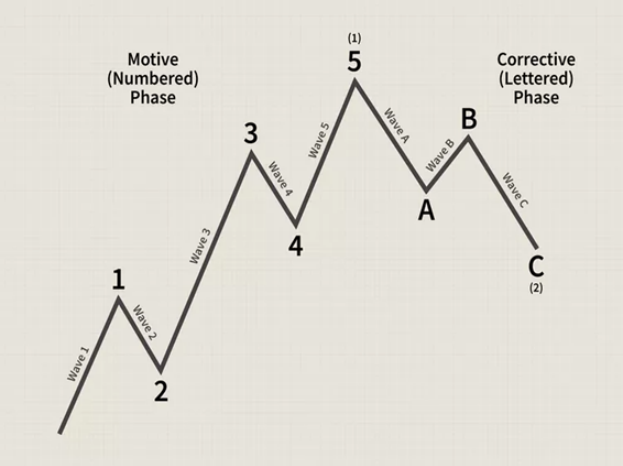


Figure 5.41: Elliott Wave

that patterns exist in market movements was extensively debated. However, recent scientific advancements have substantiated that pattern formation is an intrinsic component of intricate systems, such as financial markets. He discovered that different phases of growth or decline generate fractal-like patterns, which persist regardless of whether the time scope is zoomed in or out. There are two distinct categories of waves: corrective waves and impulse waves. Both have subsets, with the impulsive waves typically consisting of five waves preceding the three corrective waves. Ralph Nelson Elliot identified the phenomenon he termed the "wave principle." It has been observed that social or mass behavior exhibits discernible patterns of trending and reversing. By employing stock market data as one of his research instruments, Elliott was able to discern thirteen recurring patterns or "waves" within the market for stock data. He designated, elucidated, and depicted those patterns. The notion that patterns exist in market movements was extensively debated. However, recent scientific advancements have substantiated that

[26]https://www.investopedia.com/terms/e/elliottwavetheory.asp

pattern formation is an intrinsic component of intricate systems, such as financial markets. He discovered that different phases of growth or decline generate fractal-like patterns, which persist regardless of whether the time scope is zoomed in or out. There are two distinct categories of waves: corrective waves and impulse waves. Both have subsets, with the impulsive waves typically consisting of five waves preceding the three corrective waves. It can promote sales and result in reduced prices. In specific situations, sentiment exerts a greater impact on market behavior than economic principles and political developments. The pricing process and the market are intricate and challenging to grasp.

Chapter 6

Machine Learning in Financial Market

6.1 Introduction

According to Chapter 3, we achieve the knowledge and features using the feature importance concept, creating a new dataset. The next step is to pass this dataset of extra knowledge of features with machine learning and automated machine learning pipelines. In this chapter, we discuss all the concepts based on machine learning, automated machine learning, features, how the model selection is done, and model analysis with predictions. We will discuss basic concepts of machine learning and automated machine learning to gain knowledge about model evaluation and model predictions. A dynamic subfield of computational algorithms, machine learning, attempts to simulate human intelligence by acquiring environmental knowledge[7]. It allows machines to make decisions and recognize patterns without explicit programming. It incorporates many methodologies, such as reinforcement learning, unsupervised learning, and supervised learning. Supervised or controlled[153] learning models are trained on labelled datasets to predict outcomes. Unsupervised or uncontrolled learning entails identifying patterns in unlabeled data, whereas reinforcement learning entails the acquisition of knowledge by agents through feedback and interaction with their environment. In numerous domains, ML has become indispensable, thereby transforming industries. It facilitates disease diagnosis and individualized treatment programs in the healthcare industry. ML is advantageous for credit assessment, algorithmic trading, and fraud detection in the financial sector. For demand forecasting and recommendation systems, e-commerce is dependent on ML.[154] ML is utilized by autonomous vehicles for navigation and decision-making. ML is employed in Natural Language Processing (NLP) to perform language

translation and sentiment analysis, among other duties. The perpetual development of machine learning is propelling advancements and exerting an impact on diverse facets of technology and the processes by which decisions are made. The implementation of algorithms in financial marketplaces has increased[155]. By employing intricate mathematical models and statistical analysis, these algorithms discern patterns and trends within enormous quantities of financial data. These tools have many applications, such as trade execution, risk management, and market analysis. One method that algorithms are implemented in the financial markets is to assist in trading decisions. Traders can employ algorithms to formulate trading strategies that implement trades automatically in response to market conditions, including price fluctuations or specific events. The implementation of algorithms in financial marketplaces has increased. By employing intricate mathematical models and statistical analysis, these algorithms discern patterns and trends within enormous quantities of financial data. Trade execution, in addition to risk management and market analysis. One method that algorithms are implemented in the financial markets is to assist in trading decisions. An algorithm may initiate a purchase order, for instance, when a particular price is reached on a stock or when a particular news event transpires[156]. Algorithmic trading is a computer-based approach to trading in which trading decisions are generated using mathematical models and algorithms. Adopting this methodology eradicates emotional bias, and decisions are exclusively grounded in data and subsequent[157] analysis. Utilizing algorithms enables the exclusion of emotion from the decision-making process; however, this approach incurs a significant expense in algorithmic trading.

6.2 Automatic Machine Learning

Automatic Machine Learning (AutoML, also known) signifies a paradigm shift within machine learning, intending to optimize and mechanize[158] the conventionally intricate and labor-intensive procedure of constructing efficacious models. AutoML platforms automate critical components of the machine learning pipeline, such as feature or attribute engineering, data preprocessing, algorithm selection, model evaluation,[159] and hyperparameter tuning, by utilizing sophisticated algorithms and optimization techniques. The principal objective is to streamline the machine learning process so that individuals of diverse skill levels, from novices to seasoned data scientists, can utilize it by minimizing manual intervention requirements. Automatic Machine Learning (AutoML) tools frequently feature intuitive interfaces that facilitate rapid deployment and greatly speed up model instantiation.[160] This enables organizations to leverage the capabilities of machine learning for predictive analytics without the need for extensive expertise. It is critical

to acknowledge that although AutoML does provide efficiency improvements, human expertise is still necessary for tasks such as data comprehension, problem definition, and result interpretation. With the continuous progression of technology, AutoML remains a crucial component in democratizing machine learning, thereby expanding the applicability to a broader range of industries. Tpot, H2O, and Auto-sklearn are famous and mostly use automatic machine-learning models.

6.3 Implementation of Machine Learning on Finance data

By delivering insights, automating procedures, and improving decision-making, the application of machine learning (ML) to financial data has fundamentally transformed the operations of financial institutions. ML techniques are utilized in diverse domains of the finance industry to analyze data, optimize operations, and mitigate risks. Among the most prominent applications are Algorithmic trading, Risk Analysis, and other aspects of financial markets. As in Chapter 2, we have a financial dataset of BTC cryptocurrency, so all the analysis is based on financial market data. Implementing machine learning models' data and data sets is very important. For this purpose, we selected the Binance Exchange-Based BTC data. To use the machine learning model, we use the financial data set of BTC for the last 7 years. This data depends on different time intervals, e.g. 1-minute data sets, 3-minute data sets, 5-minute datasets, and 15-minute data sets to train such complex models. On a chart, a time-frame is a specified duration denoted by a solitary candlestick or bar. Every candlestick or bar represents the cryptocurrency's opening, closing, highest, and lowest prices[161] during the specified period. For instance, on a one-hour time-frame, the price action of the hour is represented by each candlestick. In this way, we get the mentioned time-frame datasets. The duration of long-term time-frames varies from days to weeks to months. These instruments are well-suited for investors with a longer-term position holding the horizon and are more concerned with the underlying fundamental factors that influence the cryptocurrency market. Long-term time-frames comprehensively depict the primary trends and patterns that characterize cryptocurrency. Noise and fluctuations brought about by short-term events and emotions are also diminished. For example, each candlestick symbolizes a single trading day. Among swing traders seeking medium-term signals and trends, this time-frame is prevalent. Timespans ranging from minutes to hours are considered short-term. They are well-suited for frequent traders with a greater interest in the technical variables influencing the cryptocurrency market.

Brief periods offer more trading opportunities and signals due to their ability to capture cryptocurrency's swift price fluctuations and transformations. Additionally, they enable the trader to swiftly modify their positions and capitalize on minor price fluctuations. The short time trading time-frame is highly volatile and difficult to handle. Many models predict best in long-term time-frames but fail in short-term prediction due to high risk and high volatility. As we use time frames, e.g. 1,3,5, and 15 minutes, which are short-term time-frame datasets and highly volatile, it's very risky and hard to predict the price of these time-frame levels. After getting the datasets, we know that When training machine learning models using Bitcoin (BTC) data, the initial data preparation steps are considerably simplified due to the absence of repetitive values and null or void values in the finance dataset. This distinctive attribute reduces the necessity for laborious data cleansing procedures frequently demanded in alternative fields. The purity of the dataset facilitates a smoother progression to critical stages of model development, including training, feature engineering, and time-series analysis. By utilizing a pristine dataset, one can concentrate on capitalizing on the intrinsic patterns and trends in the Bitcoin data, refining the feature selection process, and honing machine learning models to generate precise predictions and perceptive analyses. The efficient data preparation procedure establishes a strong groundwork for constructing resilient models capable of accurately representing the intricacies of the cryptocurrency market. When the dataset is cleaned, the next step is choosing the features for prediction of what we want to predict using machine learning models. The feature engineering strategy utilized to forecast the closing value of Bitcoin (BTC) over the past seven years is predicated on open, high, low, and close (OHLC) data. Incorporating temporal attributes such as the day of the week and month mitigates the influence of possible cyclic patterns. By utilizing lagged features of the closing price, one can discern recent market behavior by capturing past trends. Indicators of volatility, which are calculated using high and low prices as benchmarks, aid in comprehending the attributes of price fluctuations. This optimized methodology concentrates exclusively on the inherent patterns present in the primary OHLC data by avoiding the utilization of supplementary technical indicators. The objective is to develop a set of features that is both streamlined and efficient, thereby improving the model's capacity to forecast BTC closing values by leveraging past market dynamics. One thing to remember is that, as we know, each time-frame has unique datasets, so all these data cleansing, data wrangling and feature engineering steps must be done for these datasets. After we know that, we want to predict the price of BTC using historical datasets. But we first use standard scalar to scale the features, then use these scale features to evaluate machine learning and automatic machine learning models. The following ex-

plained models were used in our research. It's important to know the basics of each machine learning model to evaluate machine learning. That is why, first in this chapter, we discuss machine learning and terminologies used in this research. Then, how the model is selected and evaluated in this chapter will also be discussed after explaining the terminologies and machine learning models.

6.3.1 Machine learning Models

KNeighbors Regressor

The KNN algorithm applies to problems involving both classification and regression. 'Feature similarity' is utilized by the KNN algorithm to forecast the values of newly added data points. This implies that the value ascribed to the new point is determined by its similarity to the points in the [162]training set. Initially, the distance is computed between the newly introduced point and every training point. K data points in proximity are chosen (via distance). The final estimation for the new point(s) is calculated as the mean of these data points. [1] The initial phase is calculated by calculating the distance

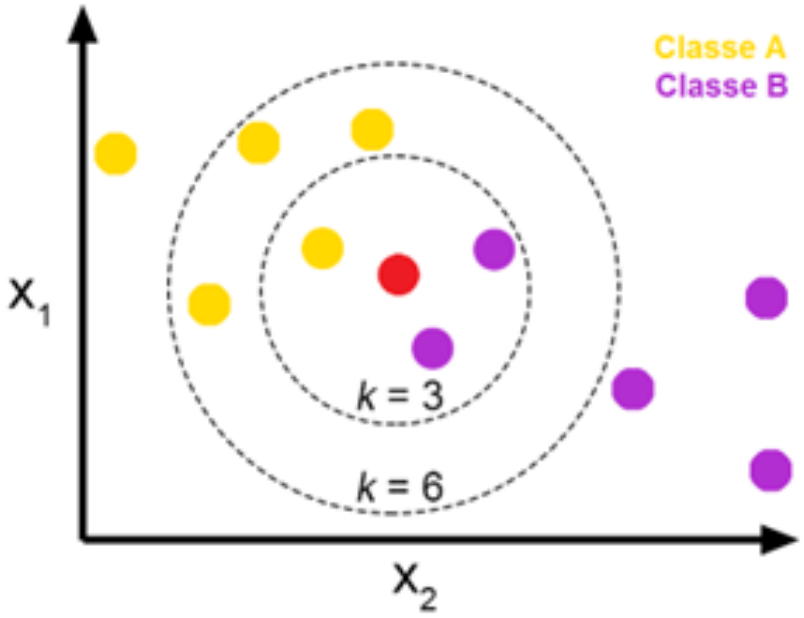


Figure 6.1: KNeighbors Regressor

between the new point and each training point. Multiple approaches exist to compute this distance, with the Euclidian, Manhattan (for continuous), and Hamming distance methods being the most widely recognized.

[1]https://miro.medium.com/v2/resize:fit:1100/format:webp

Euclidian Distance Calculation

The square root of the squared differences between a newly introduced point (x) and an already established point (y) yields the Euclidean.

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{6.1}$$

Manhattan Distance

It refers to the sum of the absolute variations/differences of two real vectors to determine their distance.

$$\text{Manhattan Distance} = \sum_{i=1}^{n}|x_i - y_i| \tag{6.2}$$

Hamming Distance

The Hamming Distance is a measure applied to categorical variables. The distance D equals zero when the values (x) and (y) are identical. D=1 otherwise

$$\text{Hamming Distance} = \sum_{i=1}^{n}|x_i - y_i| \tag{6.3}$$

$$\text{If } x = y, \text{ then } D = 0 \tag{6.4}$$

$$\text{If } x \neq y, \text{ then } D = 1 \tag{6.5}$$

The Hamming Distance is a measure applied to categorical variables. The distance D equals zero when the values (x) and (y) are identical. D=1 otherwise, based on these distances, the algorithm subsequently determines the k-nearest neighbors to be considered, where "k" represents the number of neighbors specified by the user. In the context of classification, KNN utilizes majority voting among the k-neighbors to ascertain the class of the input data point. Conversely, regression tasks require the output values of the k-neighbors to be averaged (or weighted averaged). The resulting computation is subsequently designated as the forecast for the given input data point. Prominent attributes of KNN, including its versatility in handling a wide range of data types, its responsiveness to the parameter "k," its embrace of passive learning principles, and its deliberations on scalability, all contribute to its efficacy in numerous fields and emphasize its importance within the machine learning domain.

6.3.2 Linear Regression

Linear regression is an extensively employed and foundational algorithm in supervised machine learning. It operates under the assumption of a linear association amidst one or more than one independent variable (predictors or features) and a dependent variable (also referred to as the objective). The principal aim of linear regression is finding the best-fitting linear equation that describes a relationship between the variables. Simple linear regression involves a single predictor, and the corresponding linear equation is denoted as

$$y = mx + c \tag{6.6}$$

y = x + b, in which y represents dependent variable, m signifies line's slope, x denotes independent variable b signifies y-intercept.

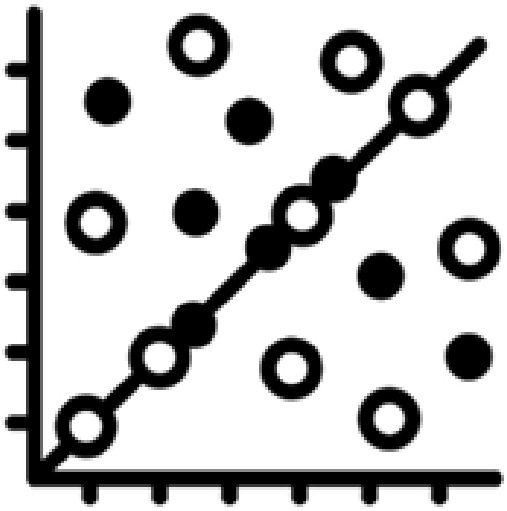


Figure 6.2: Linear Regression

[2] y = x + b, in which y represents the dependent variable, m signifies the line's slope, x denotes the independent variable, and b signifies the y-intercept. Understanding the relationship between variables and predicting numerical values (regression) are two applications in which linear regression is especially useful. The analysis is computationally efficient and interpretable and yields valuable insights regarding the magnitude and orientation of the relationships in the data. Conversely, if the underlying data displays non-linear patterns, linear regression's performance may be constrained due to

[2]https://cdn.vectorstock.com/i/1000x1000/02/39/linear-regression-vector-38490239.webp

its assumption of a linear relationship. One may consider[163] employing advanced or polynomial regression techniques in such situations. Notwithstanding its apparent simplicity, linear regression continues to be a fundamental instrument in statistics and machine learning, serving as a precursor to more intricate modelling methodologies, data analysis, and predictive modelling.

6.3.3 Decision Tree Regressor

A prevalent algorithm in machine learning, decision trees apply to regression tasks and classification. Their simplicity in implementation[164], interpretation, and comprehension renders them an optimal selection for novices embarking on the journey of machine learning. A decision tree handles classification and regression assignments, the non-parametric supervised learning algorithm. The system's structure is a hierarchical tree comprising leaf nodes, internal nodes, branches, and a root node. To facilitate classification and regression, decision trees are implemented as models that are simple to comprehend. In decision support, a decision tree is a hierarchical model representing decisions and their potential outcomes, considering random occurrences, resource costs, and utility. This algorithmic model[165] is non-parametric and supervised learning; it employs conditional control statements and applies to classification and regression tasks. Root nodes, branches, leaf nodes and internal nodes compose a tree structure resembling a hierarchical tree. It is a device that possesses a wide range of practical implementations. In addition to classification, decision trees may also be applied to regression issues. Its nomenclature indicates that it exploits a tree-shaped flowchart structure to represent predictions generated by an order of feature-founded divisions visually. It commences with some root node and infers with some leaf-based decision. [3]

Terminologies used in Decision Tree

Root Nodes: The root node is the primary node in the decision tree, from which the complete population or dataset is divided by various conditions or features.
Decision Nodes: Decision nodes are nodes that are produced when root nodes are split. The intermediate conditions or decisions are represented by these nodes within the tree.
Leaf nodes: Nodes that cannot be further divided and frequently represent the ultimate classification or result. Terminal nodes are an alternative name for the leaf nodes.
Sub-Tree: A subordinate part of the decision tree is denoted as a sub-tree,

[3]https://miro.medium.com/v2/resize:fit:1100/format:webp/0*IY07ftCHvVuuw35_*png*

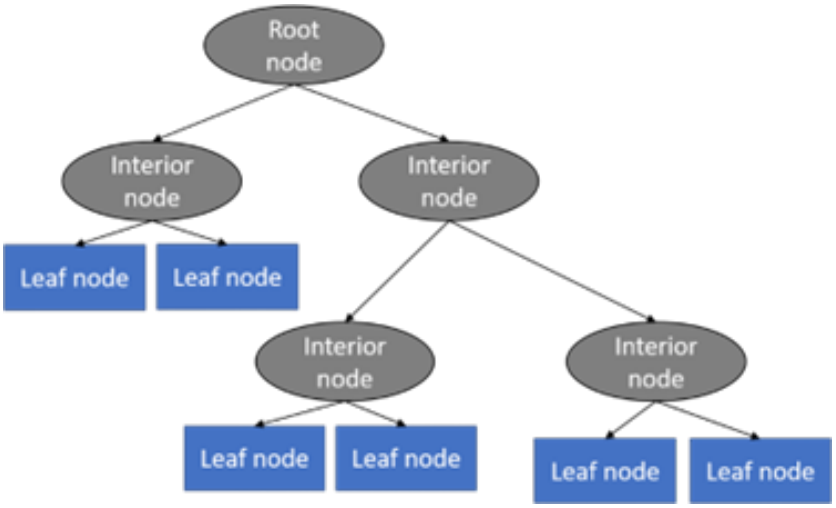


Figure 6.3: Decision Tree Regressor

analogous to how a graph subdivision is designated as a subgraph. It signifies a distinct segment of the decision tree.
Pruning: Pruning is a procedure by which particular nodes in a decision tree are eliminated or reduced in number to make the model more straightforward and prevent overfitting.
Branch or Sub-Tree: A subtree or branch denotes a subsection of the complete decision tree. It denotes a distinct trajectory of choices and results within the arboreal structure. **Parent and Child Node:** A parent node is a node in a decision tree subdivided into child nodes; the sub-nodes emanating from it are called child nodes. The parent node symbolizes a condition or decision, whereas the offspring nodes represent possible outcomes or subsequent decisions that may occur by that condition.
Inverting decision trees entails positioning the root at the top and dividing it into numerous nodes. For the uninitiated, decision trees consist of a collection of if-then statements. If the condition is satisfied, it advances to the subsequent node associated with that determination.

How it works

Commencing from the root: The algorithm commences at the apex, referred to as the "root node," symbolizes the complete dataset.
Employing Optimal Questioning: It seeks the feature or query that divides the data into the most distinct and significant clusters. Comparable to posing an inquiry at a crossroads in a tree.
Branching Out: It generates new branches by dividing the data into smaller subsets in response to the answer to that query. Each limb in the tree symbolizes a potential pathway. **Iterating through the Procedure:**

The algorithm proceeds to pose inquiries and partition the data at each branch until it arrives at the ultimate "leaf nodes," symbolizing the anticipated results or categorizations.
Certain assumptions inform the construction of decision trees and influence their performance. Frequently, binary divisions are performed by decision trees, which divide data into two subsets according to a single feature or condition. A recursive partitioning process is employed, whereby child nodes are generated until a halting criterion is satisfied. While it is a common assumption, feature independence may only sometimes be the case in practice. The objective is to generate homogeneous subgroups within each node, ensuring that samples within a node are maximally similar concerning the target variable. Using a greedy, top-down methodology, decision trees choose divisions to maximize information gain or minimize impurity. The model accommodates numeric and categorical attributes, operating under the assumptions of feature equality, absence of missing values, sensitivity to outliers, and sample size. Entropy, which quantifies disorder or uncertainty, is frequently employed when constructing decision trees to assess divides and improve model performance.

$$E(S) = -p_{(+)} \log(p_{(+)}) - p_{(-)} \log(p_{(-)}) \tag{6.7}$$

- p^+ denotes the probability of a positive class.
- p denotes the probability of a negative class.
- S represents the training example's subset.

$$\text{Information Gain} = E(Y) - E(|Y| \mid X) \tag{6.8}$$

6.3.4 Random Forest Regressor

Leo Bierman and Adele Cutler devised the widely used algorithm of machine learning Random Forest, which combines multiple decision tree outputs to produce a single result. It has been widely adopted due to its adaptability and user-friendliness, as it can be utilized to solve classification and regression issues. A Random Forest is comparable to a group decision-making team in machine learning. It improves prediction quality by combining the viewpoints of numerous "trees" (individual models), resulting in a more accurate and robust overall model. The Random Forest Algorithm has gained significant traction due to its adaptable and user-friendly design, allowing it to address classification and regression challenges efficiently. The algorithm demonstrates its prowess in managing intricate datasets and preventing overfitting, which renders it a valuable instrument for many predictive endeavors in machine learning.[166]

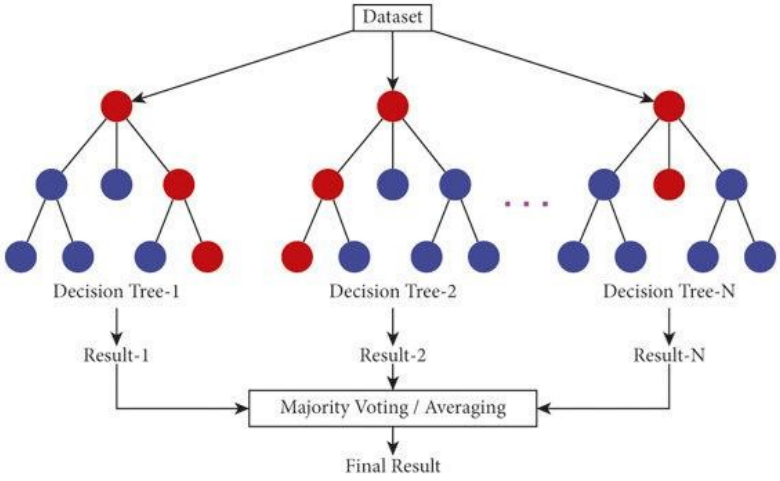


Figure 6.4: Random Forest Regressor

[4] One of the most crucial aspects of the Random Forest Algorithm is its competence to process datasets encompassing both continuous and categorical variables (just like in regression and classification correspondingly). It executes classification and regression tasks more efficiently. This tutorial aims to understand the random forest algorithm and demonstrate its application to a classification task.

Operation of the Random Forest Method

An examination of the ensemble learning technique is necessary before comprehending the random forest algorithm's operation in machine learning. To form an ensemble, one must merge multiple models. Predictions are generated by employing a compilation of models instead of a single model.

Begging: It engenders a discrete training detachment using replacement with the sample training data, plus the mainstream vote also determines the final output. As an illustration, Random Forest.
As the ensemble technique, Bagging, alternatively referred to as Bootstrap Accretion, is incorporated into the Random Forest algorithm. The following are the phases of Bagging:

- **Subset Selection:** The bagging process commences with selecting a subset, or random sample, from the complete dataset.
- **Bootstrap Sampling:** Each model is constructed using the Bootstrap Samples, instances of the original data substituted. The term for this method is row sampling.

[4]https://encrypted-tbn0.gstatic.com/images

- **Bootstrapping:** The process of row sampling with replacement is commonly known as bootstrapping.

- **Training of Independent Models:** Every model undergoes training in isolation using its respective Bootstrap Sample. The outcomes of this training procedure are produced for every model.

- **Majority Voting:** The ultimate output is ascertained by majority voting the combined results of all models. The outcome that is predicted most frequently across all models is chosen.

- **Aggregation:** Aggregation is the process of generating the ultimate output through majority voting and combining all the results.

Boosting: It improves weak learners into strong learners through the sequential construction of models with the utmost accuracy in the final iteration, such as the ADA BOOST and the XG BOOST. Boosting is among the methodologies that implement the ensemble learning concept. The final output of a boosting algorithm is generated by joining numerous elementary models (also denoted as weak learners/base estimators). It is achieved by constructing a model from feeble models connected in series.[167] Although numerous boosting algorithms exist, AdaBoost is the initial genuinely successful algorithm designed specifically for binary classification. Adaptive Boosting, abbreviated as AdaBoost, is a widely used boosting method that merges several "weak classifiers" into a solitary "strong classifier." The Random Forest algorithm proceeds in a sequential fashion, comprising discrete phases. At the outset, a laboriously chosen subset of characteristics and data points is utilized to construct each decision tree in the model. This process involves retrieving n random records and m features from larger datasets containing k records. Following this, distinct decision trees are constructed for each sample in the second phase. Each of these trees generates a unique output as it develops. In the fourth and final stage, the final prediction is ascertained using a prudent methodology: in the case of cataloguing, mainstream voting is applied; for regression, averaging is employed. Through its ensemble structure, the Random Forest can generate accurate and reliable predictions by amalgamating the merits of numerous decision trees. **Features of Random Forest:**

- **Diversity:** When constructing a single tree, not every attribute, variable, or feature is considered; thus, the individual tree is exclusive.

- **Contrary to the curse of dimensionality:** The feature space is diminished as every single tree does not account for every feature.

- **Parallelization:** Parallelization entails the creation of individual trees using distinct data and attributes. This allows the CPU to be utilized to its maximum capacity when constructing random forests.
- **Train-Test Split:** In the context of a random forest, the need to partition the data into distinct train and test sets is obviated, as the decision tree invariably excludes 30
- **Stability:** Stability is achieved when the outcome is determined through majority voting or averaging.

$$g(x) = f_0(x) + f_1(x) + f_2(x) + \cdots \tag{6.9}$$

6.3.5 Logistic Regression

Statistically, logistic regression is a prevalent and effective technique utilized in binary classification tasks that require estimating the likelihood that a given observation belongs to a specific class. In contrast to its nomenclature, its primary function is data classification into two[168] distinct categories rather than regression analysis. The logistic regression model utilises input features to forecast an event's probability. Fundamental to this model is the logistic function, which is alternatively called the sigmoid function. It converts the linear combination of input features and coefficients into a binary range from 0 to 1. The formula for logistic regression is defined as follows:

$$P(Y = 1) = \frac{1}{1 + e^{-(b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_n X_n)}} \tag{6.10}$$

Here $b_0, b_1, b_2, \ldots, b_n$ are the coefficients that are linked to the characteristics $x_0, x_1, x_2, \ldots, x_n$. By introducing non-linearity via the logistic function, the model [169]can capture intricate relationships between the outcome variable and the features.

[5] The process of training the logistic regression model entails the utilization of maximum likelihood estimation to modify the coefficients. The aim is to optimize the probability of encountering the specified results by utilizing the selected parameters. Typically, iterative optimization algorithms such as gradient descent are employed. 0.5 is designated as the decision boundary for logistic regression. The model assigns the observation to class 1 if the predicted probability exceeds this threshold; otherwise, it designates it for class 0. Due to this binary categorization, logistic regression is especially applicable when the outcome variable consists of two distinct groups.[170] Extensions of logistic regression exist to address multi-class classification

[5]https://www.shutterstock.com/image-vector/vector-image-logistic-regression-binary-260nw-2194696121.jpg

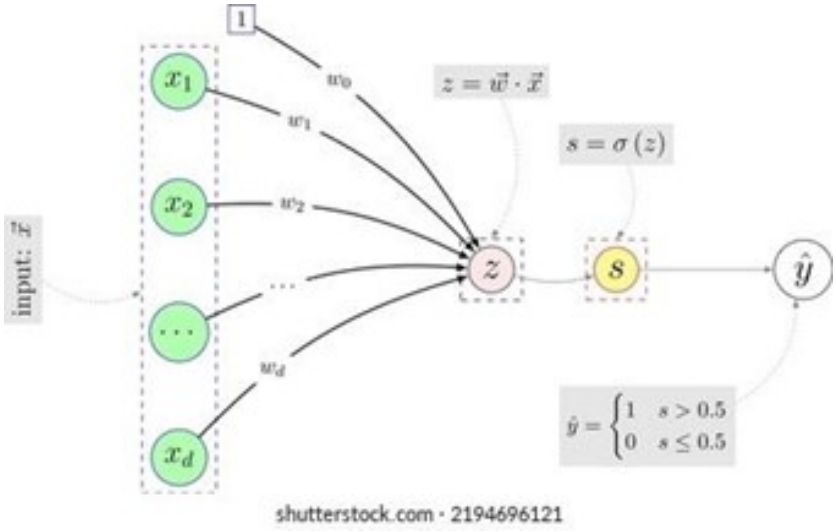


Figure 6.5: Logistic Regression

problems, including one-versus-one and one-versus-rest approaches. Due to its straightforwardness, interpretability, and effectiveness, logistic regression is widely utilized in numerous fields, e.g. finance, marketing, healthcare, and the social sciences. Nevertheless, it is critical to mention that logistic regression operates under the assumption of a model called linear[171] association model between the features and the log odds of outcome. Consequently, it might be less optimal when confronted with intricate data patterns.

6.3.6 Bayesian Ridge

Bayesian Ridge Regression is an approach to linear regression that integrates Bayesian principles into a probabilistic framework. In contrast to classical linear regression, Bayesian Ridge Regression incorporates regularization into the model, enabling the inclusion of prior beliefs concerning the coefficient[172] distribution. By incorporating a degree of zero-regression shrinkage for the estimated coefficients, this Bayesian method enables the management of multicollinearity and prevents overfitting.
The Bayesian Ridge Regression formula is mathematically represented as follows:

$$w_{BR} = \text{argmin}_w \left\{ \frac{1}{2n} \|y - x\omega\|_2^2 + \alpha\|\omega\| + \beta\|\omega\|_2^2 \right\} \tag{6.11}$$

where w_{BR}, y denotes the target variable and represents the weight vector estimated by Bayesian Ridge Regression. The design matrix of input features is denoted by X, the vector of coefficients is denoted by w, and the notation $\|.\|$ represents the Euclidean norm. The degree of regularization that is applied to the coefficients is determined by the regularization terms α, γ, and

β.

Bayesian Ridge Regression[173] is distinguished by its utilization of a prior distribution on the coefficients. The prior distribution is a Gaussian distribution with a precision of λ and a mean of zero. This prior distribution penalizes large coefficients and encourages a more reliable estimation by serving as a regularization term. The parameters are as follows: [6] The

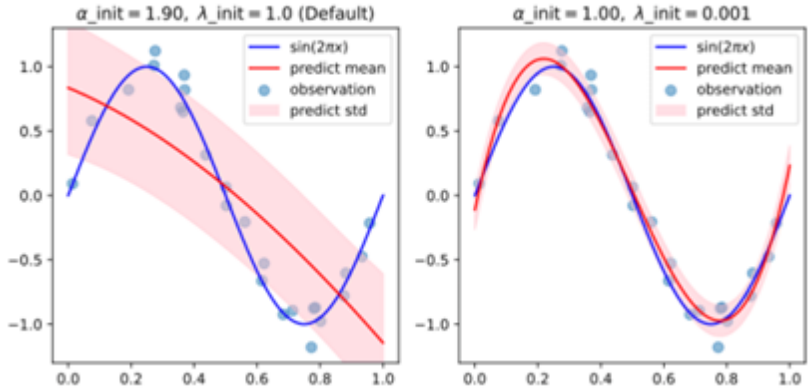


Figure 6.6: Bayesian Ridge

parameters γ and β within the equation govern the compromise between fitting the data and maintaining fidelity to the prior distribution. Typically, an iterative algorithm, such as Expectation-Maximization (EM) or Variational Inference, is employed to estimate the posterior distribution of the weights and solve the optimization problem. In addition to generating point estimates for the coefficients, Bayesian Ridge Regression also produces uncertainty estimates, rendering it well-suited for scenarios where uncertainty in the model parameters is paramount. When dealing with datasets containing high-dimensional feature spaces or when the assumptions of independence and constant variance of errors that govern classical linear regression are violated, this Bayesian approach to linear regression is especially useful. Bayesian Ridge Regression offers a versatile and principled structure for addressing various intricacies that may arise in regression inquiries.

6.3.7 LGBM Regressor

Gradient Boosting Decision Trees (GBDTs) are widely used machine learning algorithms implemented effectively in various systems, including XGBoost. Numerous optimization techniques have been derived from GBDTs. The model's efficacy and scalability are marginally compromised as several characteristics in the dataset increase. Each feature must tediously and

[6] https://miro.medium.com/v2/resize:fit:1400/1*ApxrpYULBMkGW9Sbd-e9gg.png

time-consumingly scan every instance of the data to estimate the number of possible division points, which is the primary cause of this behavior. The Light Gradient Boosting Model (LGBM) is implemented to address this issue. EFB, Exclusive Feature Bundling, and gradient-based side sampling, abbreviated as GOSS, are two techniques.[174] Using the remaining data, GOSS will eliminate a substantial proportion of the data set containing minor gradients and approximation of the overall information expansion. Data instances exhibiting substantial gradients hold more significance regarding information gain computation. GOSS can produce precise outcomes with substantial information gain using a significantly reduced dataset compared to alternative models. When utilizing the EFB to reduce the number of features, it infrequently accepts any non-zero value simultaneously while including mutually exclusive features. This influences the outcome regarding feature elimination effectiveness while maintaining the separation point's accuracy.[175]

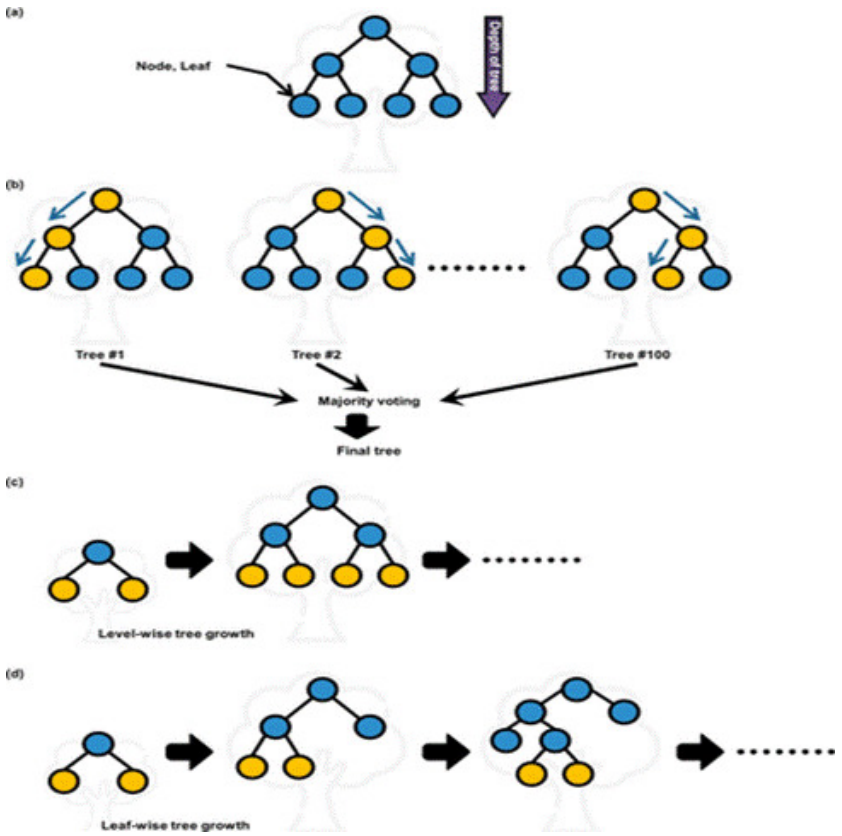


Figure 6.7: LGBM Regressor

[7] By integrating the two modifications, the training time of any algorithm can be increased by a factor of twenty. Consequently, LGBM can be concep-

[7] $https://pubs.acs.org/cms/10.1021/acs.langmuir.2c03465/asset/images/medium/la2c03465_0004.gif$

tualized as gradient-enhancing trees when EFB and GOSS are combined. The primary characteristics of the LGBM model are:

- Increased precision and accelerated training velocity.
- Low memory utilization.
- Superior accuracy compared to alternative boosting algorithms and effective management of overfitting when applied to smaller datasets.
- Parallel learning assistance.
- Compatibility with both large and small datasets.

It has become the algorithm of choice for machine learning oppositions involving regression and classification problems involving tabular data due to the benefits and characteristics of LGBM. LightGBM is an open-source gradient-boosting framework designed for machine learning that is both potent and efficient. It is optimized for managing sizable datasets and exhibits commendable performance regarding memory utilization and speed. LightGBM creates a robust predictive model by combining several weak novices (characteristically decision trees) using the gradient boosting technique.

Explanation of the Mathematical Foundations of LightGBM

A technique called verdict trees is employed to compress a function, for instance, from input space X to gradient space G. Assume a training set containing instances such as x1, x2, and xn, in which each element is a vector in space X with s dimensions. All the negative gradients of the loss function regarding output models are embodied in each restatement of gradient boosting as g1, g2, and gn. The decision tree divides each node according to the most illuminating characteristic, generating the greatest evidence gain. The variance after segregation can be used to indicate data development in this model type. The representation of it is as follows in a formula:

$$Y = \text{Tree}_1(X) - \text{lr} \times \text{Tree}_2(X) - \text{lr} \times \text{Tree}_3(X) \tag{6.12}$$

In clarification, Let suppose O represents a training dataset utilized on the fixed node of the decision tree. Moreover, the variance gain for a node is calculated by dividing measure j at a point d.

$$V_{j|0}(d) = \frac{1}{n_0}\left(\left(\sum_{x_i \in 0: x_{ij} \leqslant d} g_i\right)^2 /(n^j_{l|0}) + \left(\sum_{x_i \cdot \in 0: x_{ij} > d} g_i\right)^2 /(n^j_{r|0})\right) \tag{6.13}$$

GOSS, or Gradient One-Sided Sampling, employs haphazard sampling on instances with minor gradients while utilizing each occurrence with larger gradients. The designation "O" represents the training dataset for each specific node in the decision tree. The following expression represents the variance gain of j or dividing the measure at a point d for node

$$\tilde{V}_{j|0}(d) = \frac{1}{n_0}\left(\left(\sum_{x_i \in A_l}\left(g_i + \frac{(1-a)}{b}g_i\right)\right)^2\right)^{\sim} /(n^j_{l|0}) + \left(\sum_{x_i \in A_r}\left(g_i + \frac{(1-a)}{b}g_i\right)\right)^2 /(n^j_{r|0}) \tag{6.14}$$

6.3.8 AdaBoost Regressor

AdaBoost, which stands for Adaptive Boosting, is an ensemble learning technique that constructs a robust and precise predictive model by combining the forecasts of numerous feeble learners[176]. AdaBoost operates on the principle of assigning greater significance to observations that weak learners incorrectly classify. This enables subsequent weak learners to concentrate on the errors committed by their predecessors. The AdaBoost algorithm is an abbreviation for the Adaptive Boosting. It is a Boosting method implemented in machine learning as a collaborative method. It is called "Adaptive Boosting" because weights are reassigned for each instance, with incorrectly classified instances receiving greater weights.

$$\omega(x_i, y) = \frac{1}{N}, i = 1, 2, 3, \ldots, n \tag{6.15}$$

This algorithm constructs a model for assigning each data point an equal weight. It subsequently allocates greater weights to incorrectly classified points. Subsequently, in the subsequent model, greater emphasis is placed on all elements with higher weights. It will continue to train models until an error rate decreases.

[8] In iterations (t = 1 to T, where T represents the total number of feeble learners), as follows

1. Develop a Weak Learner: Train a weak learner (typically a decision tree) using the present instance weights on the training data. The performance of the weak learner ought to be marginally superior to that of random guesswork.

2. Determine Weighted Error: Determine the weighted error (ε_t) of the trainee set's feeble learner.

$$\varepsilon_t = \frac{\sum_{i=1}^{n} \omega_i I(y_i \neq h_t(X_i))}{\sum_{i=1}^{n} \omega_i}$$

[8] https://www.researchgate.net

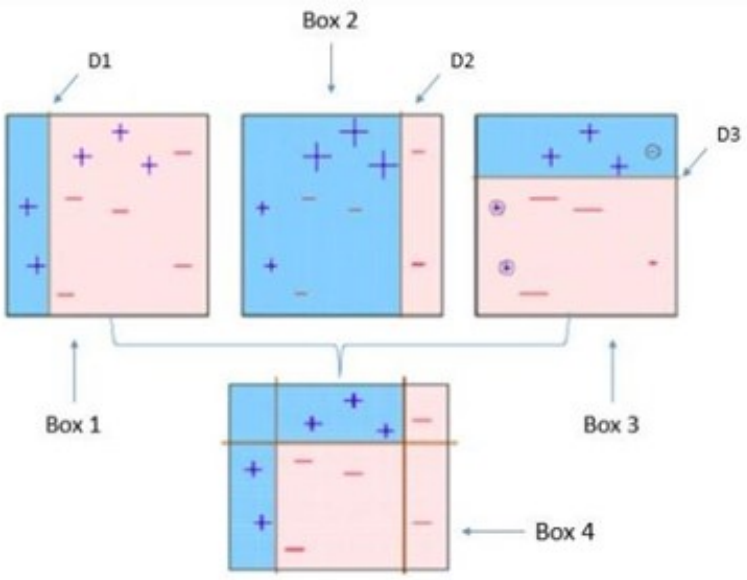


Figure 6.8: AdaBoost Regressor

where I is the Indicator Function, y_i is the true label, and $h_t(x_i)$ is the prediction related to weak learner indicating the instance i.

3. Learner Weight: Determine the weight (α_t) with respect to the feeble learner's performance.

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$$

The quantity α_t is proportional to the feeble learner's performance, with greater weights assigned to learners who demonstrate greater accuracy.

4. Revision of Instance Weights: Adjust weights of the training instances to assign greater significance to misclassified instances. This causes the weights of correctly classified instances to diminish and the weights of misclassified instances to increase.

$$\omega_i = \omega_i \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(x_i))$$

5. Consensus Prediction: Final prediction of the model is a weighted sum of the predictions made by the feeble learners.

$$F(x) = \sum_{t=1}^{T} \alpha_t \cdot h_t(x)$$

Where F(x) is the predicted target value
The final AdaBoost model is constructed by combining the weak learners into a strong model, with the accuracy of each learner influencing its contribution. The algorithm prioritizes instances that[177] present difficulties in classification in an iterative manner, thereby gradually enhancing the model's performance. Weighted voting guarantees that poor learners with higher accuracy exert a more substantial impact on the ultimate prediction.

6.3.9 XGB Regressor

XGBoost is a machine learning algorithm that falls under the classification of ensemble learning algorithms, specifically within the gradient boosting framework. The model's generalizability is enhanced by incorporating regularization methods and utilizing decision trees as foundational learners. XGBoost is widely utilized in ranking, classification, and regression because of its stellar performance in computational efficiency, feature importance analysis, and missing value management.
Hence, CERN's designation of it as the most efficacious approach for classifying signals discharged from the Large Hadron Collider is not unexpected. A unique and specific challenge posed by CERN was the need for a scalable solution capable of accurately differentiating an exceedingly rare signal from background sounds during a complex physical process and processing 3 petabytes of data annually.[178] It was concluded that XGBoost was the most pragmatic, straightforward, and robust solution.
Extreme Gradient Boosting, or XGBoost, is an algorithm for machine learning that utilizes ensemble learning. There is a trend towards supervised learning tasks, which include classification and regression. XGBoost employs an iterative aggregation process to build a prognostic model from the predictions of many individual models, often decision trees. The algorithm functions by progressively incorporating weak learners into the ensemble; each succeeding learner focuses on correcting the errors made by the antecedent ones[179]. Gradient descent optimization minimizes a predefined loss function during the training procedure. [9] The XGBoost Algorithm is distinguished by its ability to handle intricate data relationships, its implementation of parallel processing to enhance computation efficiency, and its utilization of regularization techniques to prevent overfitting. XGBoost is an exceptionally versatile algorithm that is implemented in a wide range of domains.

[9] https://www.researchgate.net

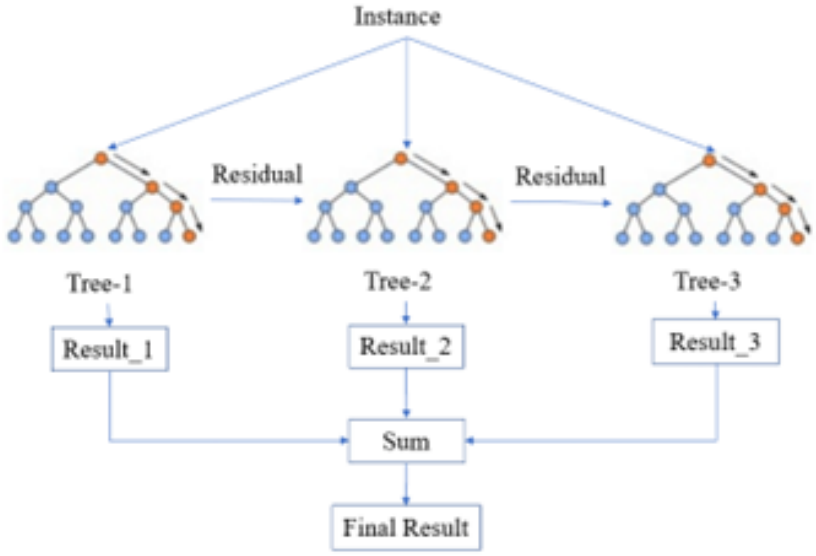


Figure 6.9: XGB Regressor

Ensemble Learning

XGBoost is an ensemble learning methodology. In certain circumstances, it may need to be more adequate to rely exclusively on the output of the solitary machine learning models. Ensemble learning offers a systematic methodology for coalescing the predictive aptitudes of many learners. A unified model is produced by combining the outputs of numerous models. The ensemble consists of base learners, models that different learning algorithms or the same algorithm may have generated. Bagging and enhancement are two widely used techniques in ensemble learning. Although both approaches can be applied to an extensive range of statistical models, decision trees have been the most frequently utilized implementation. It is crucial to begin with a succinct analysis of Bagging before exploring the complexities of gradient enhancement.

Bagging

Despite being considered among the most intuitive models, the behavior of decision trees is remarkably unpredictable. Let us contemplate a singular training dataset arbitrarily partitioned into two separate segments. We shall train a decision tree by incorporating each component to generate two models. When both models were fitted, their results would be dissimilar. High variance is associated with the behavior described in decision trees. The implementation of Bagging or boosting aggregation reduces the variance of each learner. The bundling technique utilizes a collection of parallel-generated

decision trees as its foundational learners. For training, these learners are furnished with replacement-sampled data. Each learner's cumulative results constitute the final prediction.

Increasing

Boosting is a process that entails the consecutive construction of trees in a manner that attempts to reduce the defects introduced by the previous tree. The residual errors are revised as each tree acquires knowledge from its antecedents through learning. Consequently, knowledge will be acquired by the subsequent tree in the sequence through a revised iteration of the residuals. Base learners in the boosting are weak learners characterized by a significant degree of bias; their ability to predict outcomes is only marginally more effective than that of arbitrary conjecture. Each feeble learner contributes vital information that facilitates prediction. By effectively combining these weak learners, boosting techniques can produce strong learners. The learner with the highest level of proficiency minimizes both variance and bias. Different forming techniques resembling random forests that optimize tree growth boost utilize trees with fewer divisions. Despite their relatively modest height, the trees in question exhibit a significant level of interpretability. One can attain the most effective parameter selection, including tree depth, gradient boosting learning rate, and several trees or iterations, through corroboration methods, for instance, k, k-fold cross validation. An overabundance of trees has the potential to induce overfitting. Hence, it is critical to determine the limiting criteria for augmentation carefully.
Three straightforward phases comprise the ensemble technique for gradient boosting:

1. Define an initial model F_0 for the purpose of predicting the target variable x. A residual $(y - F_0)$ will be associated with this model.

2. By fitting the residuals from the prior step to a new model h_1.

3. F_1 is the result of combining h_1 and F_0; it is the enhanced version of F_0. Mean squared error associated with F_1 will be less than that of F_0:

$$F_1(x) \leftarrow F_0(x) + h_1(x)$$

4. To enhance the performance of F_1, a new model F_2 could be constructed using the residuals of F_1.

$$F_2(x) \leftarrow F_1(x) + h_2(x)$$

5. 'm' iterations of this process are possible until residuals are minimized to the greatest extent conceivable:

$$F_m(x) \leftarrow F_m(x) + h_m(x)$$

The additive learners remain undisturbed by the functions that were established in the preceding stages. Conversely, they disseminate their own information in an effort to reduce the errors.

Features of XGBOOST

XGBoost is a widely used gradient-boosting implementation. Consider a few characteristics of XGBoost that contribute to its intrigue:

- **Regularization:** It can be applied to complex models using the L1 and L2 regularization options provided by XGBoost. The function of regularization is to prevent overfitting.
- **Sparse data management:** Sparse data can arise from data processing stages, including one-hot encoding or missing values. XGBoost integrates a sparsity-aware partition-finding algorithm to accommodate diverse sparsity patterns in data. As the quantile sketch algorithm demonstrates, most established tree-based algorithms can identify division points when the weights of the data points are equivalent. However, their capacity to handle weighted data is constrained. XGBoost's distributed weighted quantile sketch algorithm facilitates the effective administration of weighted data.
- **Block structure for parallel learning:** XGBoost can maximize processing performance using multiple CPU processors. Implementing a block structure in the system's design facilitates this process. In memory-stored and sorted data, elements are known as blocks. In contrast to alternative algorithms, this one allows subsequent iterations to leverage the data layout instead of performing a recalculation. This function is practicable in addition to split finding and column subsampling requirements.
- **Cache consciousness:** To retrieve gradient statistics via row index, XGBoost requires non-continuous memory access. As a result, the design of XGBoost prioritizes the effectiveness of hardware utilization. This is achieved through the allocation of internal buffers for the storage of gradient statistics in each thread.
- **Discretionary computing:** This feature improves the efficacy of disc space utilization in situations involving massive datasets that exceed the memory capacity.

These are the machine learning models. Now, we discuss the automatic machine learning model, how it works, and how it helps to evaluate the best-predicting models. How automatic machine learning model pipelines work.

6.4 Automatic Machine Learning Algorithms

The concept of "AutoML" refers to the development of tools and techniques that optimize the process of constructing and optimizing machine learning models. Automobile's objective is to democratize artificial intelligence capabilities by providing machine learning tools that are accessible to users with diverse levels of expertise. The primary aim of automatic machine learning models is to streamline and streamline the complex and often time-consuming procedures that are involved in the construction, training, and refinement of machine learning models.
Important characteristics of models for automatic machine learning include the following:
Automated Selection of Models: AutoML systems aid in the identification of the most appropriate machine learning algorithm for a given task. To determine the most effective algorithm for a provided dataset, they examine numerous alternatives, such as support vector machines, neural networks, decision trees, & others. The efficacy of a machine learning model is significantly impacted by the hyperparameters it is tuned to. AutoML technologies automate the process of hyperparameter tuning, which entails determining the optimal combination to boost the model's accuracy or decrease the loss function.

- **The engineering of features:** A feature engineering process is employed to heighten the model's performance by identifying and modifying relevant features existent in the input data. AutoML systems possess the capability to autonomously select, extract, and modify features, thereby augmenting the model's capacity to extrapolate insights from the data.

- **Preprocessing of Data:** Categorical variable encoding, normalization, and missing value imputation are a few of the duties that automatic machine learning models assist with. By following this procedure, it is ensured that the data is in a suitable format for the purpose of training machine learning models.

- **Validation and Evaluation of Models:** To assess the efficacy of the models that have been trained, AutoML systems often incorporate automatic model evaluation methods, including cross-validation. Implementing this approach decreases the likelihood of overfitting and produces a more precise assessment of the model's ability to generalize.

- **Ensemble methodologies:** To enhance the overall accuracy of forecasts, a number of AutoML frameworks employ ensemble approaches to combine multiple models. Automatically applying ensemble methods

to combine the strengths of multiple models and mitigate their flaws is feasible.

- **Scalability and Efficiency:** The implementation of machine learning on complex problems and large datasets is made possible by the scalability and efficiency of autonomous machine learning models. This is notably advantageous in practical scenarios where the quantity of data may be substantial.

It is critical to bear in mind that although Automatic Machine Learning models have substantially simplified the workflow, they do not serve as universally applicable solutions. While these tools aid in accelerating and democratizing the machine learning process, optimal results still necessitate a comprehensive understanding of the subject matter and domain expertise. The models of automatic machine learning models are discussed.

6.5 Models of AutoML

6.5.1 TPOT

Despite numerous AutoML tools designed to accelerate machine learning through model selection, my insatiable desire remains filled with Tree-based Pipeline Optimization Tool (TPOT), an early AutoML package. An open-source Python script, this extraordinarily intelligent tool can automate the most laborious aspect of machine learning—identifying the optimal pipeline for your data—by traversing hundreds or thousands of potential pipelines.[180] Obtaining classification accuracy on par with the competition is an absolute steal with this AutoML tool. It possesses unparalleled value. Additionally, this technique can identify artificial feature constructors that can substantially enhance classification accuracy by discovering distinct pipeline operators. The enchantment of TPOT is generated in supervised learning via genetic programming. Despite my characterization of it as a form of "magic," it cannot substitute machine learning engineers or data scientists. This instrument identifies the most significant attributes by systematically examining the data. Moreover, it proposes pipelines. The ability to export these pipelines and integrate their domain expertise with fit visualization is within the user's prerogative.

What should you consider before attempting this with a larger dataset?

- **Time Consumption:** Their quest could take several hours or even days to complete.

- **Multiple Approaches:** It could suggest various approaches for the same dataset.

Utilizing TPOT AutoML has several advantages

- **Ease of Use:** By automating many tedious and repetitive operations, TPOT streamlines the machine learning process.
- **Superior Pipelines:** TPOT uses various methods to create superior pipelines, including genetic programming and hyperparameter tweaking.
- **High Configurability:** TPOT lets users choose a range of hyperparameters and limitations to create pipelines tailored to their requirements.
- **Scalability:** TPOT is appropriate for big data applications since it can grow to massive datasets and distributed contexts.
- **Open-source:** Many individuals and organizations can utilize TPOT since it is free to use and open-source.

Driven by Darwin's theory of natural selection, Tpot uses genetic programming to create an optimized search space. Genetic programming takes advantage of the following properties:

- **Selection:** The fitness function is assessed at this point for each individual, and its values are normalized so that each has a value between 0 and 1, and their total is 1. Next, some random numbers, say R amid 0 and 1, are chosen. We now retain the people whose fitness function value is greater than or equal to R.
- **Crossover:** To create a new population, we can choose the most fit individuals from the above group and conduct crossover between them.

Several significant Tpot functions

The automated learning module for the supervised classification challenge is called TpotClassifier. Some significant arguments it makes are listed below:

- **Generations:** the number of pipeline process iterations (by default, 100).
- **Population size:** the number of individuals (by default, 100) to be kept in the population for genetic programming per generation.

- **Offspring size:** the quantity of offspring produced throughout each round of genetic programming (100 by default).
- **Mutation rate:** between [0,1] mutation rate (0.9 by default).
- **Crossover rate:** between [0, 1] crossover rate (0.1 by default) crossover rate + mutation rate <= 1.
- **Scoring:** measures to assess the pipeline's quality. Here, scoring considers factors like F1 score, accuracy, etc.
- **Cv:** cross-validation method; in K-Fold cross-validation, the value will be K if it is an integer.
- **n_job**: the maximum number of concurrent processes that can operate (by default, 1).
- **max_time_mins**: Tpot's maximum allotted time for pipeline optimization (default: None).
- **max_eval_time_mins**: The maximum number of minutes that TPOT has to assess a single pipeline is max_eval_time_mins (default: None).
- **Verbosity**: The amount of data that TPOT shows on the screen at any time.
    1. Nothing
    2. little details
    3. additional details and a progress meter
    4. everything (default: 0)

TpotRegressor

The TpotRegression module automatically carries out deep learning tasks for regression. Most of the arguments used to designate the TpotClassifier above are similar. The scoring is the only variable[180] that is different in this case. To assess the regression using TpotRegression, we employ parameters like "r2," "neg_median_absolute_error," "neg_mean_absolute_error," and "neg_mean_squared_error."

- **fit(features, target):** Utilize the provided data to run the TPOT optimization pipeline.
- **predict(features):** To forecast target values of an example or examples of a feature set, use the optimized pipeline.

- **score(test features, test target):** This function assesses the model using test data and yields the best possible score.
- **export(output file name):** The optimized pipeline can be exported as Python code.

6.5.2 H20 Regressor

Open-source H2O is an AI and machine learning platform. Compatibility with the Flow web interface has been established. H2O Flow enables the creation of a diverse range of machine learning models without coding. Pipelines for machine learning can be constructed with a simple point-and-click operation. Furthermore, H2O provides a web-based graphical user interface (GUI) that employs these methods via JSON. Deploying H2O AutoML-trained models on AWS Spark platforms is a straightforward process. One of the main advantages of H2O AutoML is that it enables developers to allocate their efforts towards other responsibilities, such as feature engineering, data collection, and model deployment. This is achieved by automating various processes, including fundamental data processing, tuning and model training, collaborative construction, and model piling to optimize their performance.

The features of H2O AutoML

- H2O AutoML offers essential data processing functionalities. Every H2O algorithm incorporates these as well.
- Trains a random grid of algorithms (e.g., GBMs, DNNs, and GLMs) by carefully selecting a hyperparameter space.
- Cross-validation is used to fine-tune each model.
- They train two stacked ensembles. Optimized for model performance, one ensemble includes every model, while the other includes only high-performing models from each class/family of algorithm (augmented for production use).
- Yields an all-model sorted "Leaderboard."
- Exporting any model to production is simple.

Buildings

H2O architecture is used in H2O AutoML. Different APIs will be found at the top of the H2O architecture, while the H2O JVM will be at the bottom. Using socket connections, H2O offers REST API patrons for Excel,

R, Python, Tableau, and Flow Web UI. Several parts that will operate on the H2O JVM process are contained in the bottom layer.

- A cluster of H2O is made up of one or more nodes. There is just one JVM process per node. Three layers comprise each JVM process: language, core infrastructure, and algorithms.
- The language layer is the first layer at the bottom. It comprises the Scala layer and a manifestation assessment engine for R.
- The algorithm layer is the second layer. It contains algorithms like XGBoost, GBM, Random Forest, K-Means, and others previously included in H2O.
- The core infrastructure or third layer handles resource management, including memory and CPU management.

We can make many arguments like these:

- **Threads:** The number of CPU cores that the H2O server can employ; automatically, it uses all of them.
- **Ip:** The host's IP address on which the H2O server will operate. It makes use of a local host by default.
- **Port:** The port that H2O server will operate on.
- **max_mem_size:** A string of characters that tells H2O how big the memory allocation pool can get in bytes. This number needs to be more than 2MB and a multiple of 1024. To signify megabytes, affix the letter M, and append the letter G to indicate gigabytes.

The machine learning models, and automatic machine learning models are explained above. Now, these are the terminologies and concepts for finding the best models for evaluation. For example, R_2_Score and other terminologies are also discussed below. Every model will be differentiated using these terminologies. So, it's important to know these terminologies.

6.6 Terminologies used for Results of models

6.6.1 R_2_Score

A statistical measure, the R-squared (R2) score, is alternatively known as the coefficient of the determination. It enumerates the degree to which independent variables, in a regression model, account for variability perceived

independent variables. Alternatively stated, it evaluates the degree to which the regression model adequately accounts for variability in observed data, thus indicating its goodness of fit.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{6.16}$$

- R^2 represents the R-squared value.
- The sum of squared residuals SS_{res} signifies inconsistency between the observed values of the dependent variable and those prophesied by the model.
- The total sum of squares SS_{tot} represents the entire variance of the dependent variable.

The sum of all squares SS_{tot}

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \overline{y_i})^2 \tag{6.17}$$

n Denotes the number of observations. y_i Embodies the actual value of a dependent variable. At the time of observation $\overline{y_i}$, the mean of the dependent variable.

The sum of residuals squared SS_{res}

$$SS_{res} = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \tag{6.18}$$

Where $\widehat{y_i}$ signifies the observed value of the dependent variable as predicted. For the regression model.

R_2 **Calculation**

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{6.19}$$

The range of the R_2 score is 0-1. A value of 0 indicates that the model nosedives account for any erraticism witnessed in the dependent variable, although a value of 1 signpost that the model sufficiently accounts for all variability.

6.6.2 MSE or Mean Square Error

Mean Squared Error, abbreviated as MSE, is a recurrently used metric in the valuation of some regression model's enactment. The metric analyses the average squared deviation between predicted and actual values. The MSE formula is expressed as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{6.20}$$

- **MSE** stands for "Mean Squared Error."
- The value of n epitomizes the number of observations in the dataset.
- $y_i$represents the true value of the dependent variable at the time of observation i.
- Where i denotes the anticipated value of the dependent variable for observation $\widehat{y}_i$ per the regression model.

The square root of residuals

$$(y_i - \widehat{y}_i)^2 \tag{6.21}$$

For each observation, determine the squared deviation from the true value y_i in conjunction with the anticipated value $\widehat{y}_i$.

The product of the squared residuals

$$\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{6.22}$$

The sum of the residuals squared for each observation.

Residuals Squared Average

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2 \tag{6.23}$$

The average squared difference can be calculated by dividing the sum of squared residuals by the observations. The mean square error (MSE) quantifies the degree of agreement between the predictions generated by a regression model and the observed values. The reduced mean square error (MSE) signifies enhanced performance, as it designates that the model's predictions approach the actual values on average.

- The MSE of a flawless model would be zero, signifying the absence of any incongruity between predicted and observed values.
- An increase in the mean square error (MSE) shows that, on average, the model's prophecies diverge more from true values.
- Although MSE is a frequently employed metric, it is susceptible to outliers. When differences are squared, the impact of larger errors is amplified. When outliers can substantially impact a model's performance, alternative metrics such as Huber loss or Mean Absolute Error or MAE may be considered.

MSE is a prevalent and uncomplicated metric utilized to evaluate the precision of regression models. It does so by quantifying the average squared discrepancy between values predicted and actual outcomes. It indicates the comprehensive model fit and may prove beneficial when contrasting various models or fine-tuning model parameters.

6.6.3 RMSE (Root Mean Squared Error)

The Root Mean Squared Error is a metric commonly employed to assess a regression model's effectiveness, especially when imposing a greater penalty on larger errors relative to smaller ones is advantageous. RMSE is obtained by dividing the average discrepancies between the prophesied and genuine values by the square root of the mean squared error (MSE). The RMSE formula is expressed as follows:

$$\text{RMSE} = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6.24}$$

- RMSE stands for "Root Mean Square."
- MSE stands for "Mean Squared Error."
- The value n symbolizes the number of observations in the dataset.
- y_i represents the true value of a dependent variable at the time of observation i.
- i denotes the anticipated value of the dependent variable for observation $\hat{y}_i$ by the regression model.

The square root of residuals

$$(y_i - \hat{y}_i)^2 \tag{6.25}$$

For each observation,y_i determine the squared deviation from the true value in conjunction with the anticipated value $\hat{y}_i$.

The product of the squared residuals

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{6.26}$$

The average squared difference can be calculated by dividing the sum of squared residuals by the number of observations.

RMSE: root for the MSE

$$\text{RMSE} = \sqrt{MSE} \tag{6.27}$$

RMSE is calculated using the square root of MSE. This phase is incorporated to restore the error metric to the same scale as the initial dependent variable. RMSE is more interpretable because it is expressed as the same unit as the dependent variable. A reduced RMSE indicates enhanced model performance, as it signifies that the model's predictions approach the actual values on average. Due to the squaring of residuals, bigger errors have a more pronounced effect on the RMSE, which is sensitive to outliers. When comparing the accuracy of various regression models or determining the effect of parameter modifications, RMSE is a useful metric. Nevertheless, it is critical to consider the problem's context, as specific applications may give precedence to categories of defects. Furthermore, in the case of distorted distributions or outliers, it may be prudent to contemplate alternative metrics like Mean Absolute Percentage Error abbreviated as MAPE or Mean Absolute Error/MAE.

The next concept is Optuna. Optuna is just for the best hyperparameter optimization. Optuna Is fast and gives better hyperparameters than other hyperparameter optimization methods, e.g., grid search. These models are trained with hyperparameter tuning. In parameter optimization, Grid Search is frequently employed as a starting point. By employing techniques such as cross-validation, this approach meticulously investigates a predetermined grid of hyperparameter parameters and assesses the performance of the simulation for every possible combination. Although Grid Search may appear simple and straightforward to execute, it does possess certain limitations, most notably in the pace and efficiency of computation, especially when confronted with an extensive space of hyperparameters. An enhanced and versatile methodology could involve the integration of Optuna into one's hyperparameter optimization strategy. Bayesian optimization is incorporated into the sequential model-based optimization (SMBO) methodology utilized by Optuna. In contrast to Grid Search, Optuna dynamically adjusts its search algorithm in response to the outcomes of preceding trials.

The ability to adapt enhances Optuna' s efficiency and enables it to navigate a complex hyperparameter space with greater effectiveness. The framework gradually optimizes the model's efficacy by systematically sampling new hyperparameter values. Although Grid Search is easier to use due to its simplicity, its potential sluggishness should be considered, particularly in larger hyperparameter spaces. When time can be significant, and freedom is important, contemplate switching to Optuna, a sophisticated methodology that will notably expedite the hyperparameter optimization procedure. So, for hyperparameter selection, the best choice is Optuna. More information about Optuna will be discussed next. How it works, its features and other information about Optuna will be discussed below.

6.7 Optional

For machine learning models, Optuna is an open source hyperparameter optimization framework. It is a user-friendly and effective platform designed by Preferred Networks, Inc. that facilitates the adjustment of hyperparameters to optimize the functionality of machine learning algorithms. A critical stage in the machine learning pipeline, hyperparameter optimization entails locating the optimal set of hyperparameter values that produce the most effective model.

6.7.1 Features of Optuna

Objective Function

Defining an objective function that assesses the proficiency of the machine learning model with a specified set of hyperparameters is central to the optimization procedure in Optuna. The aim is to determine whether this objective function is minimized or maximized.

Search Spaces

Optuna provides extensive support for various search spaces, encompassing continuous, discrete, and categorical variables regarding hyperparameters. The adaptability facilitates an exhaustive investigation of the hyperparameter space.

Trials

Within Optuna, a trial denotes an individual assessment of the objective function utilizing a predetermined configuration of hyperparameter values.

To investigate the hyperparameter space efficiently, Optuna conducts iterative trials.

Study

A study is a compilation of trials, and Optuna searches the hyperparameter space utilizing various algorithms. The research maintains a log of the optimal hyperparameters identified throughout the optimization procedure.

Optimization Algorithms

Optuna offers a diverse range of optimization algorithms, including random search and TPE,which govern the exploration of the hyperparameter search space.

Integration with Libraries for Machine Learning

Optuna exhibits a high degree of compatibility with widely used machine learning libraries, including scikit-learn, TensorFlow, PyTorch, and XGBoost. This facilitates the integration of hyperparameter optimization into users' pre-existing machine-learning workflows.

Distributed and Parallel Optimization

Optuna facilitates parallel and distributed optimization, allowing users to execute multiple trials concurrently to accelerate the hyperparameter search procedure.

Visualization

To analyze the optimization results, Optuna provides visualization tools. Users can acquire valuable insights into hyperparameter tuning by examining statistics, convergence graphs, and optimization history.

Pruning

Pruning is an Optuna feature that enables the premature termination of trials with a low probability of producing optimal results. This facilitates the acceleration of the optimization process and conserves computational resources.

Storage Back-ends

Optuna supports various storage back-ends, enabling users to store optimization results persistently. This facilitates the resumption of optimization of experiments and the sharing of results. As every Machine learning model selector knows when selecting an appropriate machine learning algorithm for various applications, execution time is a crucial determinant. Data processing speed directly affects scalability, efficacy, and resource utilization. Scalability is enhanced in algorithms with reduced execution times, enabling them to effectively manage substantial datasets and real-time processing demands without substantial interruptions. Furthermore, increased execution speeds improve the overall system efficiency, resulting in financial savings through more effective utilization, especially in cloud computing. Accelerating the innovation cycle during the model development phase, speedier execution permits rapid iteration and experimentation. Minimizing exe minimization in production environments is critical to adhering to service-level agreements and ensuring optimal system availability. Moreover, rapid execution permits flexibility in response to dynamic environments by facilitating prompt model retraining to account for shifting data distributions or requirements. Hence, it is imperative to take execution time into account in conjunction with other performance metrics when choosing machine learning algorithms that achieve an optimal trade-off between computational efficiency and predictive performance.

6.8 Execution Time

Jupyter Notebook Extensions (nbextensions) are supplementary components designed to improve the visual as well as functional aspects of Jupyter Notebooks. One such extension is the Execute Time nbextension, which presents the execution time of individual code cells within the interface of the Jupyter Notebook. This can be advantageous for monitoring the execution time of specific cells while coding. Utilizing the Utilizing Time nb extension is as follows:

6.8.1 Proceed with installation

To utilize the time nb extension within your Jupyter Notebook environment, installation is required. This can be accomplished with the following command.

```
jupyter nb extension install execute_time/ExecuteTime
```

6.8.2 To activate the nbextension

It is necessary to enable the Execution Time nb extension following installation. Implement the subsequent command.

```
jupyter nbextension enable execute_time/ExecuteTime
```

6.8.3 Usage

- After the nb extension has been successfully installed and enabled, the Jupyter Notebook can be executed ordinarily.
- The extension will exhibit the execution time in the upper-right corner of the code cell when it is executed.

The Execute Time nb extension offers a practical method for monitoring the execution time of individual code cells. This is advantageous in performance analysis and optimization, extending the temporal complexity of distinct code segments. It is important to note that nb extensions are context-dependent and may require installation and activation for every Jupyter Notebook session or environment in which they are intended to be utilized. The Execute Time extension is primarily useful for rapidly determining the execution time of code cells without requiring additional code annotations or special commands. Now we know each model and Each terminology used for developing and evaluating the model to predict the price of each timeframe dataset. For this dashboard, we have scaled features. The next step is to evaluate each model in each timeframe dataset.

6.9 Socket Io

Websites previously required a refresh each time a resource was requested. This resulted in unwarranted delays that prolonged the average wait time. Users frequently had to wait minutes for a page or file to load. In contrast, real-time applications (e.g., push notifications, instant messaging, and online gaming) operate during a specified time interval to provide users with an immediate and current resource version. These applications strive to maintain the lowest possible latency to provide a consistent and seamless user experience. Socket.IO was developed by Guillermo Rauch and introduced to the public in 2010. Introducing an event-driven, real-time paradigm overcomes the shortcomings of conventional HTTP communication. Presently, many Internet applications are built upon a Client-Server architecture. A client is an individual who requests a server. In response to the request, a server provides suitable outcomes. Because of the duties that they each execute,

these two entities are similar. The browser serves as an exemplary client application. Typically, browser clients and servers exchange data via HTTP requests and responses. This communication is flawed in that it only permits the transmission of a request or a response at any given time. To enhance comprehension, envision it as a half-duplex connection. Furthermore, HTTP headers are replete with superfluous data that becomes inconsequential once a connection between the client and server is established. In contrast, sockets operate at the transport layer of the network stack. Reducing the number of redundant fields enhances the efficacy of web-based information transfer. Using the same underlying concept, Socket.IO enables bidirectional communication between web clients and servers. To manage them separately and efficiently

- the client library operates on browsers.
- server

6.10 Server Sent Event

Server-Sent Events (SSE) is a web technology that permits servers to deliver up-to-the-minute updates to consumers via a solitary HTTP connection. In contrast to conventional HTTP requests, which require the client to commence communication, SSE enables the server to instigate correspondence and promptly update the client with newly acquired data. SSE operates on the principle of unidirectional data transmission and does not rely on UDP–it utilizes a persistent HTTP connection.

6.10.1 Principal SSE Characteristics

Unidirectional Communication

SSE enables communication exclusively between the host and the client. This unidirectional data transmission is appropriate when the server must deliver notifications or updates to connected clients.

Format based on text

SSE messages are transmitted in a text-based format, facilitating client-side processing and parsing. The information is commonly conveyed as unformatted text and is amenable to processing via JavaScript.

Event-driven design principles

SSE is constructed with an event-driven framework. The server transmits events to the client, which can be received and responded to by the client-side JavaScript. Therefore, it is highly suitable for applications that necessitate real-time updates.

Individual HTTP Connection

A solitary, persistent HTTP connection between the client and the server is upheld by SSE in the interim. This approach enhances efficiency beyond specific alternatives by repeatedly eliminating the need to establish and terminate connections for every update.

Reconnection by Opposition

SSE incorporates mechanisms to reconnect automatically. The client will endeavor to reconnect automatically in the event of a connection failure, thereby guaranteeing an uninterrupted stream of updates.

6.11 AI base Prediction Dashboard

The terminologies, techniques, and technologies mentioned above are used to develop a dashboard that predicts the price of BTC time intervals: 1-minute, 3-minute, 5-minute, and 15-minute.

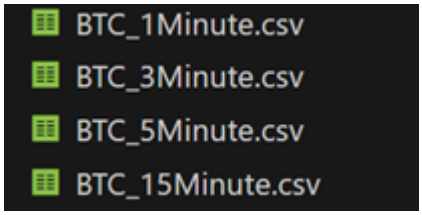


Figure 6.10: CSV Data Files

The major work of this dashboard is to train the model, which gives more accurate values or predictions. For this, the explained models are used with different concepts.
After the hyperparameter optimization is finished using Optuna, the next phase entails utilizing hyoptimized values to conduct comprehensive training of the machine learning models. This stage is critical, as it allows the models to adjust to the optimized hyoptimizedters, assuring that they accurately represent the complexities of the latent patterns within the data. Providing the algorithm with instructional data facilitates acquiring knowledge and

refining internal parameters to optimize optimization. Within this context, the models undergo training using discrete time intervals recorded by the Kline: one minute, three minutes, five minutes, and fifteen minutes. Every model is customized with specific attributes and fluctuations that exist during the corresponding period.

Figure 6.11: AI base Prediction Dashboard

To ensure the accuracy of the dataset, the training set to each time interval has been subjected to rigorous cleansing and manipulation processes to remove any values that are absent or outliers. Including preprocessing features, such as technical indicators like moving averages and a relative strength index (the RSI indicator), enhances the models' capacity to identify significant patterns within the designated time intervals. During the training process of the models on their corresponding datasets, it is critical to evaluate their performance by employing essential evaluation metrics. Indicating the extent to which the model accounts for the variability observed in the dependent variable, the R2 score provides valuable insights regarding the overall fit. In contrast, the root mean squared error (RMSE) and mean squared deviation (MSE) measure the precision of predictions quantitatively by calculating the average and root mean of the squared differences among predicted and actual values. The performance metrics function as a guiding principle when selecting models. Models that possess greater R2 scores and reduced MSE and RMSE values exhibit enhanced predictive capabilities during the specified time intervals. By thoroughly assessing these metrics, it is possible to determine which model achieves the most advantageous trade-off between variance and bias, thereby guaranteeing reliable extrapolation to novel, unobserved data. Thorough evaluation of these metrics holds particular importance in domains where exactitude and reliability are paramount, such as healthcare prognostications or financial forecasting. Hence, this assessment

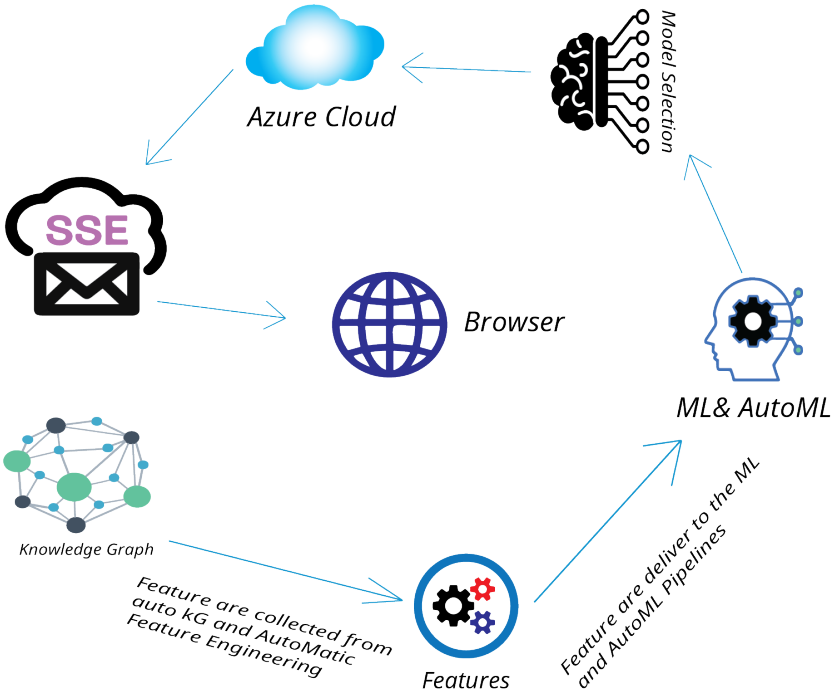


Figure 6.12: Algorithm of Dashboard

following hyperparameter tuning, which includes training on discrete time intervals, serves as a crucial milestone, enabling professionals to select models that derive advantages from optimized hyperparameters and demonstrate the utmost predictive precision for their periods. By adopting this exhaustive methodology, the ultimate models chosen to serve as evidence of the complex interaction that occurs when optimizing hyperparameters, assessing performance thoroughly, and comprehending the subtleties of various time intervals to attain optimal predictive modelling. TPOT and H2O are essential instruments in the ever-changing domain of automated machine learning for effectively managing the intricacies associated with model selection and tuning of hyperparameters. The main objective of this advanced application is to utilize machine learning models for prediction purposes across different time intervals, with a specific emphasis on the data extracted using Kline at one minute, three minutes, five minutes, and fifteen minutes. Prominent among these are TPOT, which is widely recognized programming, and H2O, which utilizes to investigate and assess a wide variety of algorithmic alternatives and the hyperparameter configurations in an efficient manner. As the frameworks for autonomous machine learning progress, they navigate a vast terrain of algorithmic and hyperparameter spaces, dynamically adjusting their approaches to capture the distinct temporal patterns that define each interval of time. H2O methodically investigates and capitalizes capitalization

parameter space, whereas TPOT systematically employs genetic programming to generate a population of potential models. The result is a collection of varied models, each meticulously adjusted to capture the intricate dynamics of its assigned time. Following training, TPOT and H2O surpass simple model evaluation. The organization's extensive collection of critical metrics comprises, among others, mean squared error (MSE), root mean squared error (RMSE), precision, recall, and F1 score. These metrics function as numerical assessments of the model's predictive precision, explanatory capability, and overall efficacy. By providing a comprehensive assessment of the models' performance, these metrics provide valuable insights regarding the models' capacity to generalize generalized observed data and identify immediate and enduring trends in the cryptocurrency market. Additionally, both frameworks identify the model with the highest performance during each time interval and produce comprehensive Python scripts that contain the ideal configuration and hyperparameters. The Python files function as concrete artefacts, representing the refined insights acquired via the automated machine-learning procedure. The specifications encompass the model itself and comprehensive particulars regarding the data preprocessing and feature engineering procedures distinct to the 1-minute, 3-minute, 5-minute, and 15-minute time intervals. Furthermore, H2O furnishes a leaderboard with exhaustive data regarding the trained models, enabling professionals to assess and contrast the efficacy of various configurations and algorithms. By distinguishing the highest-performing models, the leaderboard facilitates a more comprehensive comprehension of their merits and demerits with the given data.

6.11.1 Implementation of Diverse Models

Training machine learning models with the preprocessed data constitutes the final stage. One may utilize neural networks, decision trees, linear regression, and random forests, which are even more sophisticated. In addition, modifying the hyperparameters is essential for optimizing the model. Hyperparameters are configuration settings that influence the learning process and can substantially affect the accuracy of a model when the optimal combination is discovered.

By integrating historical data, utilizing techniques, integrating feature engineering to improve model input, applying standard scaling to ensure feature scales are consistent, and employing a variety of machine learning models with hyperparameters that have been meticulously tuned, this dashboard forecasts BTC prices at various time intervals. This all-encompassing strategy aims to construct a resilient and precise forecasting model for fluctuations in cryptocurrency prices.

The table below shows the results of all the models evaluated. As this table

shows, the R2_Score is nearly the same as all the models. So, to distinguish the models, we use MSE and RMSE terminologies. According to these terminologies, the model can best predict if it has MSE and RMSE. Also, this table has an execution time column using the Jupyter extension. These are the model training execution times.

Table 6.1: Model Results On 1 Minute dataset

Time Interval	Model	R2_Score	MSE	RMSE	Execution Time
1 minute	KNeighborsRegressor	0.999473229	133513.0196	365.3943343	4m 49s
	LinearRegression	0.999999372	159.2866986	12.62088343	14.8s
	DecisionTreeRegressor	0.999998615	351.0960265	18.73755658	22m 56s
	RandomForestRegressor	0.999998599	355.1692133	18.8459336	22m 44s
	LogisticRegression	0.999998612	351.6906129	18.75341603	22m 48s
	BayesianRidge	0.999999372	159.2866886	12.62088304	22.1s
	LGBMRegressor	0.999939654	15298.26534	123.6861566	20m 35s
	AdaBoostRegressor	0.996572854	868628.725	932.0025349	2h 31m 14s
	XGBRegressor	0.99994187	14736.51317	121.3940409	29m 13s

Table 6.2: Model Results On 3 Minute dataset

Time Interval	Model	R2_Score	MSE	RMSE	Execution Time
3 minute	KNeighborsRegressor	0.999098043	228839.2656	478.3714724	1m 23.7s
	LinearRegression	0.999998184	460.8235087	21.46680015	5.57s
	DecisionTreeRegressor	0.999996039	1004.851771	31.69939701	7m 19s
	RandomForestRegressor	0.999996033	1006.477941	31.7250365	7m 24s
	LogisticRegression	0.999996033	1006.451818	31.72462479	7m 20s
	BayesianRidge	0.999998184	460.8236354	21.4668031	7.63s
	LGBMRegressor	0.999943941	14211.83916	119.2134185	7m 15s
	AdaBoostRegressor	0.997036243	751947.1901	867.1488858	44m 34s
	XGBRegressor	0.999943371	14356.39885	119.8181908	23m 4s

Table 6.3: Model Results On 5 Minute dataset

Time Interval	Model	R2_Score	MSE	RMSE	Execution Time
5 minute	KNeighborsRegressor	0.998677385	336198.8614	579.8265787	41.8s
	LinearRegression	0.999996847	801.3455725	28.30804784	2.06s
	DecisionTreeRegressor	0.999992913	1801.438627	42.44335786	4m 6s
	RandomForestRegressor	0.999992797	1830.843599	42.78835821	4m 4s
	LogisticRegression	0.999992766	1838.879218	42.88215501	4m 6s
	BayesianRidge	0.999996847	801.3455601	28.30804762	2.81s
	LGBMRegressor	0.999943241	14397.15616	119.9881501	10m 48s
	AdaBoostRegressor	0.996944182	776766.2043	881.3434088	25m 15s
	XGBRegressor	0.999944831	13993.86997	118.2956887	19m 30s

Table 6.4: Model Results On 15 Minute dataset

Time Interval	Model	R2_Score	MSE	RMSE	Execution Time
15 minute	KNeighborsRegressor	0.998118524	482009.3282	694.2689163	10.6s
	LinearRegression	0.999991442	2192.563511	46.82481726	623ms
	DecisionTreeRegressor	0.999979078	5359.831053	73.21086704	15.1s
	RandomForestRegressor	0.999978715	5452.863629	73.8435077	15.7s
	LogisticRegression	0.999979091	5356.612895	73.18888505	1m 15.0s
	BayesianRidge	0.999991442	2192.566759	46.82485194	855ms
	LGBMRegressor	0.999941839	14618.19245	120.9057172	40.1s
	AdaBoostRegressor	0.996754226	831524.1818	911.8794777	6m 50s
	XGBRegressor	0.99994366	14160.64962	118.9985278	11m 27s

6.12 Automatic Machine Learning Results

As we mentioned above, the machine learning model results. The table below shows the results of all automatic machine learning models, e.g. TPOT and H20, with all the terminologies used above a column name Description have the best model given by automatic machine learning models.

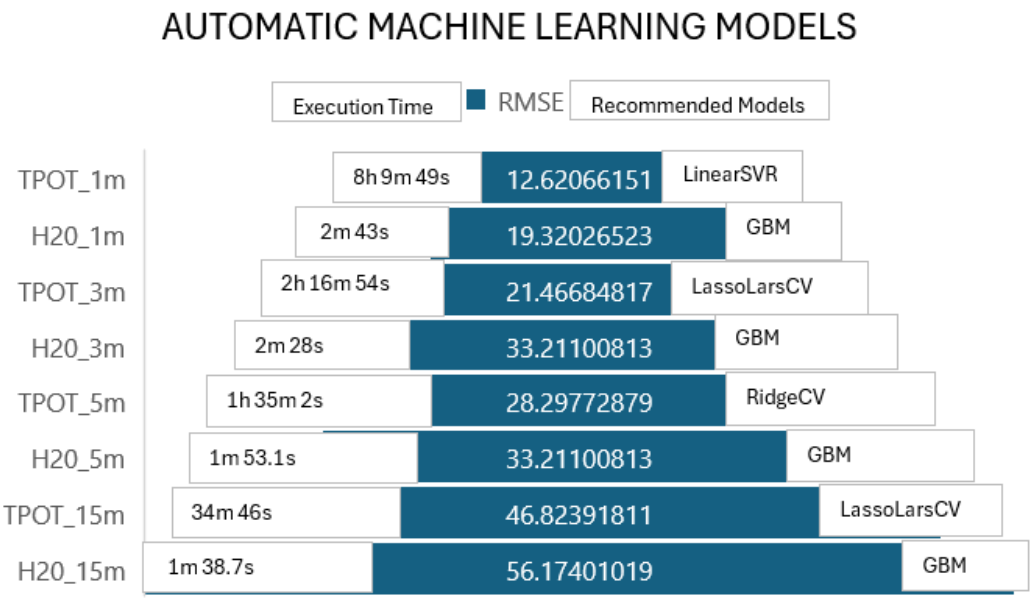


Figure 6.13: AutoML Models

Table 6.5: Automatic Machine Learning Results

Time Interval	Model	R2_Score	MSE	RMSE	Execution Time	Resulted Models
1 minute	TPOT_1m	0.999999372	159.2810969	12.62066151	8h 9m 49s	LinearSVR
	H20_1m	0.999999999	373.2726486	19.32026523	2m 43s	GBM
3 minute	TPOT_3m	0.999998184	460.8255703	21.46684817	2h 16m 54s	LassoLarsCV
	H20_3m	0.999999999	1102.971061	33.21100813	2m 28s	GBM
5 minute	TPOT_5m	0.99999685	800.7614545	28.29772879	1h 35m 2s	RidgeCV
	H20_5m	0.999999999	1102.971061	33.21100813	1m 53.1s	GBM
15 minute	TPOT_15m	0.999991442	2192.479307	46.82391811	34m 46s	LassoLarsCV
	H20_15m	0.999999999	3155.519421	56.17401019	1m 38.7s	GBM

The pipeline for model development is sped up by utilizing a data-rich methodology, but the chosen models are also guaranteed to be resilient and adaptable to the distinct temporal attributes of various intervals. The generated Python files are exhaustive blueprints to ensure reproducibility, enabling professionals to effortlessly duplicate, enhance, and expand their models. Using H2O in this data-intensive procedure, data scientists and analysts can comprehensively comprehend market dynamics. This enables them to develop precise and adaptable predictive models for various time intervals within the cryptocurrency market. Under its automated and iterative characteristics, this process facilitates scalability, precision, and efficiency, allowing professionals to maintain a leading edge in predictive modelling amidst the swiftly changing financial environment.

6.12.1 Model Selection

After the model selection procedure, XGBoost and LightGBM emerged as the leading models capable of forecasting cryptocurrency prices for various time intervals (Kline-extracted data: 1-minute, 3-minute, 5-minute, and 15-minute). Both models are well-known for their gradient-boosting frameworks, which provide an extensive collection of tools for predictive modelling. The comprehensive assessment procedure considered essential performance indicators, such as mean squared error (MSE), R2 score, and root mean squared error (RMSE), which provided insightful comparisons. XGBoost, a model renowned for its outstanding predictive capabilities, demonstrated competitive outcomes, attesting to its aptitude for identifying complex patterns within the cryptocurrency market. Nevertheless, the comparatively sluggish training process for XGBoost gave rise to concerns in situations where computational efficacy is of the utmost importance. LightGBM's performance was exceptional throughout all assessed periods due to its distributed and efficient gradient-boosting framework. In addition to rapid training durations, the model consistently demonstrated reduced root mean square error (RMSE) values compared to XGBoost and alternative automatic machine learning models. The rapidity of the training process is especially remarkable in contexts requiring timely predictions, such as real-time applications. The selection of LightGBM as the ultimate model was further supported by its remarkable computational efficiency, which rendered it highly suitable for processing high-frequency financial data. The model's capacity to attain reduced root mean square error (RMSE) values throughout all periods provided additional evidence of its superiority. Although XGBoost demonstrates exceptional predictive accuracy, LightGBM's advantage lies in situations that demand immediate model deployment and real-time prediction capabilities, as evidenced by its speedier training periods. LightGBM is deemed the optimal model for forecasting cryptocurrency prices because of

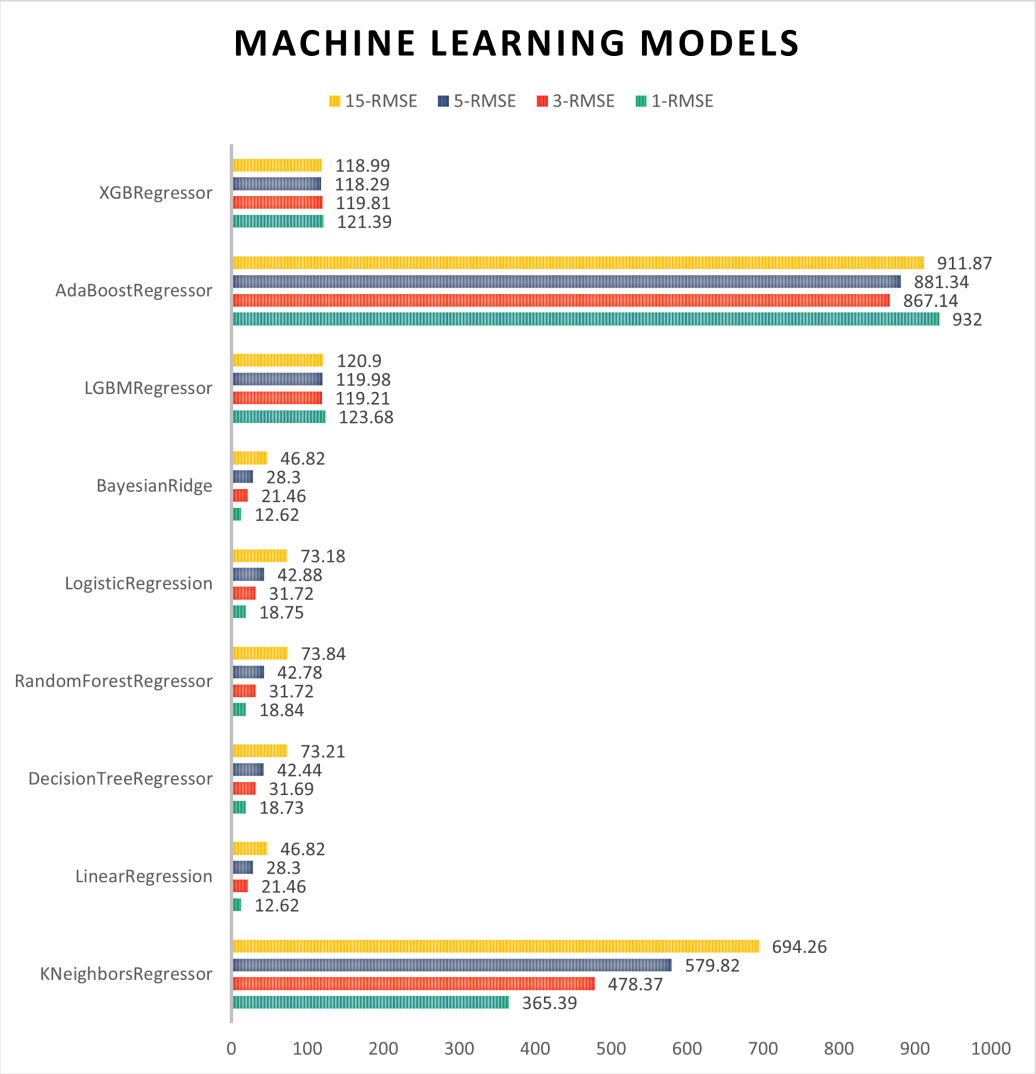


Figure 6.14: Machine learning Models

its remarkable amalgamation of precise predictions, streamlined training process, and appropriateness for handling high-frequency financial data. Considering computational efficiency and model performance, the sophisticated decision concludes that LightGBM is the most suitable option for precise and punctual forecasts in the ever-changing cryptocurrency market.

```
LGBM
def objective(trial):
    params = {
        'learning_rate': trial.suggest_float('learning_rate',
        0.01, 0.2),
        'max_depth': trial.suggest_int('max_depth',
        3, 10),
        'n_estimators': trial.suggest_int('n_estimators',
        50, 200),
        'subsample': trial.suggest_float('subsample',
        0.8, 1.0),
        'colsample_bytree':trial.suggest_float('colsample_bytree',
        0.8, 1.0),
        'num_leaves': trial.suggest_int('num_leaves',
        10, 100),
    }

    lgbm_model = LGBMRegressor(**params)
    lgbm_model.fit(X_train_scaled, y_train)
    y_pred = lgbm_model.predict(X_test_scaled)

    mse = mean_squared_error(y_test, y_pred)
    return mse

study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=50)

best_params = study.best_params
best_mse = study.best_value
final_model = LGBMRegressor(**best_params)
final_model.fit(X_train_scaled, y_train)
y_pred = final_model.predict(X_test_scaled)
```

```
XGB
def objective(trial):
    params = {
        'learning_rate': trial.suggest_float('learning_rate',
        0.01, 0.2),
        'max_depth': trial.suggest_int('max_depth',
        3, 10),
        'n_estimators': trial.suggest_int('n_estimators',
        50, 200),
        'subsample': trial.suggest_float('subsample',
        0.8, 1.0),
        'colsample_bytree':trial.suggest_float('colsample_bytree',
        0.8, 1.0),
    }

    xgb_model = XGBRegressor(**params)
    xgb_model.fit(X_train_scaled, y_train)
    y_pred = xgb_model.predict(X_test_scaled)

    mse = mean_squared_error(y_test, y_pred)
    return mse

study = optuna.create_study(direction='minimize')
study.optimize(objective, n_trials=50)
best_params = study.best_params

best_mse = study.best_value
final_model = XGBRegressor(**best_params)
final_model.fit(X_train_scaled, y_train)
y_pred = final_model.predict(X_test_scaled)
```

6.12.2 Best Models for Implementing

After determining that LightGBM was the most effective predictive model, the following stage was devoted to improving the deployment strategy to increase storage efficiency. Joblib utilized the pickle module during initial serialization to address concerns regarding the files' size. These issues were resolved by implementing joblib with gzip compression, and the storage and retrieval procedures were optimized, and optimized in a more agile and resource-efficient system.

The investigation advanced to Flask, an adaptable and diminutive web

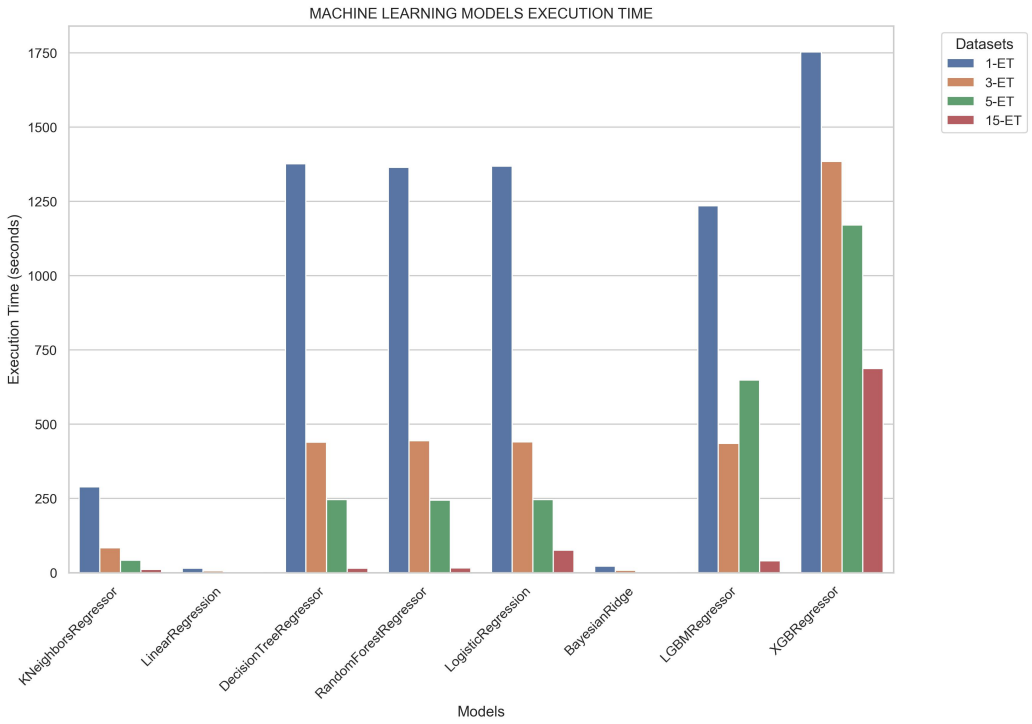


Figure 6.15: Machine learning Models Execution Time

framework that functioned as the critical element in establishing a robust server infrastructure. The Flask application was carefully designed to retrieve real-time data from the Kline in synchronized time intervals: 1-minute, 3-minute, 5-minute, and 15-minute intervals per the model's training schedule. The successful establishment of a consistent and one-way communication channel between the server and consumers was facilitated by deploying the Server-Sent Events (SSE) protocol. SSE surfaced as a sophisticated resolution optimized for sustaining persistent connections. By implementing this design decision, the server could effectively transmit real-time forecasts to connected clients, thereby obviating the necessity for frequent request-response cycles. The implementation of SSE demonstrated a dedication to

optimizing or optimizing, especially in situations where timely updates are required at precise moments. The decision to migrate from Socket.IO to SSE was made strategically, with simplicity and efficiency in managing persistent connections as primary motivators. SSE, a characterized directional data stream, exhibited seamless alignment with the model's predictions, thereby furnishing a dependable and expandable real-time communication solution. The intentional choice was made to incorporate Next.js into the framework for the interface. Following its reputation for adaptability and user-friendliness, Next.js was an ideal complement to Flask. By incorporating Next.js, a responsive and dynamic user interface was achieved, enabling users to interact with real-time cryptocurrency price predictions intuitively. Comprised of LightGBM for precision, Flask for server-side efficiency, SSE for streamlined communication, and Next.js for an engaging interface, this comprehensive account describes how a sophisticated and resilient predictive analytics platform is assembled. In addition to furnishing precise forecasts that are synchronized and synchronized time intervals, this platform offers users a seamlessly incorporated interface, enabling them to interact with cryptocurrency price data in real time. Every element's careful and precise engineering signifies a dedication to effectiveness, precision, and user satisfaction within the ever-changing domain of financial forecasts.

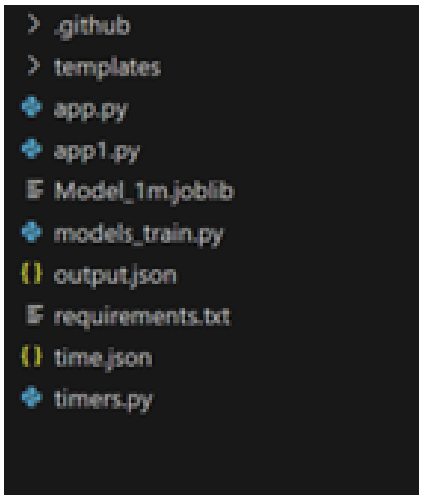


Figure 6.16: folders with joblib files

After the comprehensive implementation of the predictive model had been completed, the subsequent course of action entailed deploying both the front-end and back-end components onto the Microsoft Azure cloud platform. The back-end component was implemented on the Web App service, whereas the UI component was hosted on an Azure Virtual Machine (VM) instance. The robust and dependable hosting was ensured by utilizing Azure's utility and infrastructure capabilities in this distributed deployment architecture. The

Flask application and the LightGBM-based predictive model comprised the back end, deployed via the Azure Web App service. This platform provides a completely managed environment that facilitates the smooth implementation and expansion of web applications. By utilizing, integration with other Azure services, automatic scaling, and streamlined administration were all made possible. Regarding the front end, the decision was reached to implement it on an Azure-hosted virtual machine. The virtual machine functioned as the hosting environment for the Next.js application, granting it the ability to modify and regulate the front-end's dependencies and configuration. A Nginx server was implemented on the virtual machine to bolster the front end's security. Nginx is a highly resilient and efficient web server renowned for its capability to manage concurrent connections. Acting as an intermediary between the user's web browser and the Next.js application, it processed incoming queries and delivered the appropriate responses efficiently. Additionally, an SSL certificate was deployed utilizing a tool to ensure the security of user-frontend communication. Widely utilized for utilizing and managing SSL/TLS certificates, Certbot ensures the encryption and security of data transmitted between the user's browser and the front end. This aspect assumes heightened significance when handling sensitive financial information or user engagements. Using a reverse proxy configuration, the Nginx server directed requests to the Next.js application operating on the virtual machine. Obtaining an SSL certificate via Certbot added security by encrypting all data between the user's browser and the virtual machine.
Let's discuss which part of this dashboard. The first thing is to discuss the image below each box and its usage.

Let us discuss the first box. This box shows the live UTC. Because every trader and financial analyst knows that the financial market works on UTC at a standard time, this yellow color box shows the live time of UTC. In the

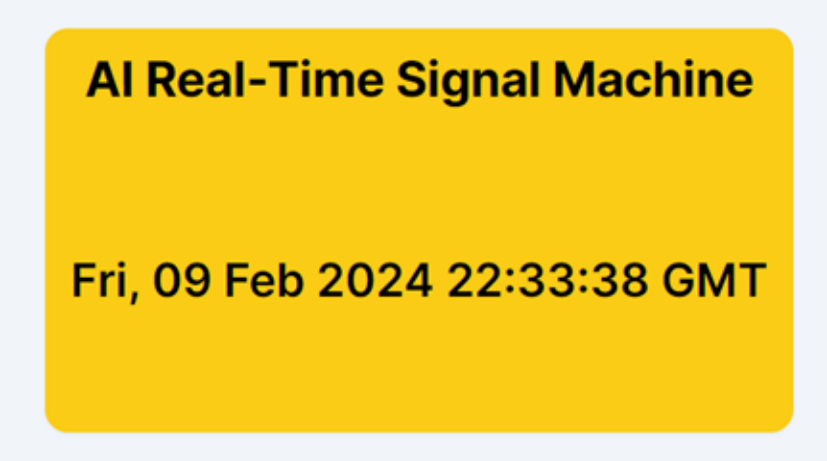


Figure 6.17: UTC Time

next box, the live BTC price according to the Binance API of the live stream. Why do we do this? The reason is that if you do not know the current price of the BTC, how will you know whether the prediction is correct or not? there is a small concept also present in this fig: the volume. It shows the volume of traders coming into BTC cryptocurrency, according to Binance.

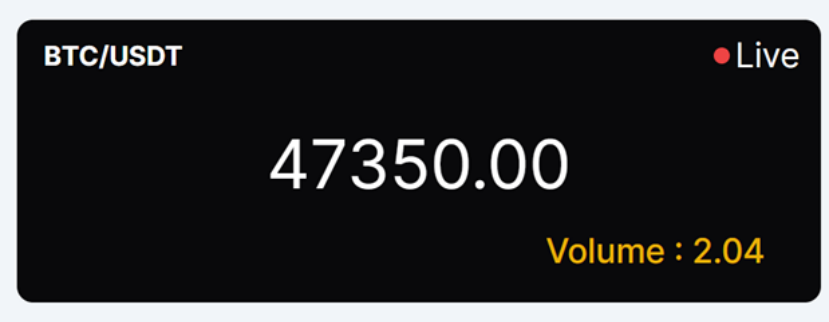


Figure 6.18: BTC LIVE PRICE WITH VOLUME

The figure below shows the prediction of the one-minute LGBM model according to the timeframe of 1 Minute. There is a timer on the right side of this box. This timer is synchronizing time. And when this timer is zero. The next candle stick prediction comes to the screen. Like the upper figure, the

Figure 6.19: 1-Minute Prediction

other timeframes are working the same, and each has its own timer according to the timeframe. Each prediction changed according to their timeframes based on UTC synchronized. [10]

Figure 6.20: Different timeframes Predictions

[10]https://predictor.pgot.net

Chapter 7

Algorithmic Time Machine BOT

Ever since the early 2000s, algorithms have experienced a profound and paradigm-shifting evolution within the financial sector, which has had a profound impact on trading methodologies. Algorithms had achieved a significant market presence by 2005, with a noteworthy 25% of the market being occupied. In the twenty years that followed, their prevalence increased exponentially, culminating in an astounding 85% market share by the end of the previous decade. In the realm of modern financial markets, algorithms have evolved into essential instruments, facilitating the effortless navigation of enormous quantities of data by means of sophisticated mathematical models and statistical analysis. This pervasive application transcends numerous financial responsibilities, including risk management, comprehensive market analysis, and trading decision-making.

An essential application of algorithms in the realm of financial markets is to assist traders in the formulation of strategic decisions. By developing algorithmic trading strategies, traders can mechanize transactions in accordance with ever-changing market circumstances, including price volatility and the progression of particular events. For example, an algorithm may initiate a purchase order in response to breaking news incidents or when a stock reaches a predetermined price. The rapid implementation of algorithmic trading serves as evidence of its effectiveness, emphasizing its capacity to influence and mould contemporary market dynamics profoundly. Algorithmic trading is predicated on mathematical models that underpin computer-based strategies; this guarantees that trading decisions are objective and devoid of emotional biases. Through the deliberate exclusion of visceral responses, decision-making is reduced to a thorough examination of data, thereby fostering a methodical and structured approach.

Nevertheless, this strategic advantage is full of corresponding expenses. Ensuring the continued efficacy of algorithmic trading requires a significant financial commitment towards state-of-the-art technology, which in turn re-

quires continuous maintenance, enhancements, and technological advancements. The expenditures related to information access and market data acquisition, in addition to transaction costs, collectively increase the overall financial burden imposed by algorithmic trading strategies. Notwithstanding these financial factors, the benefits of algorithmic trading are considered substantial, particularly in contrast to the inherent drawbacks associated with irrational decision-making that traders frequently confront. With the ongoing dynamic evolution of financial markets, algorithmic trading is positioned to maintain its prominent and influential status. It offers a sophisticated and data-driven method for navigating the intricacies of contemporary market environments. The ongoing enhancement and incorporation of algorithms into financial systems highlights the lasting influence and flexibility of this revolutionary technology in the domain of investment and trading. In previous chapters, we already knew about the financial market and its core concepts. In this chapter, we will discuss how algorithms work in the financial market and how algorithmic trading provides benefits in the financial market. For this, we already know in Chapters 1 and 2 that we have knowledge about features, and in this chapter, these features are called indicators. We will discuss in detail these features and how they are used in algorithmic bots.
Traders rely heavily on indicators because they provide them with indispensable instruments for assessing market dynamics and generating well-informed judgements. Trend identification is a critical component in trading, wherein traders rely on indicators such as moving averages to discern the general trajectory of market prices. Furthermore, these metrics validate the robustness of patterns, as illustrated by the MACD or RSI, enabling investors to assess momentum and formulate strategic judgements regarding trade entry and exit. Volatility indicators, including Bollinger Bands and ATR, serve the purpose of quantifying market fluctuations and provide traders with guidance regarding potential opportunities or phases of consolidation. Potential reversal points are indicated by oscillators such as the RSI and stochastic oscillator, which assist in identifying overbought or oversold conditions. Indicators produce entry and exit signals, and actionable insights are obtained through the utilization of moving average crossovers and other techniques. Indicators play a critical role in risk management by enabling the establishment of stop-loss orders, which serve to restrict potential losses.
Additionally, a comprehensive trading strategy is enhanced by the capability to tailor indicators to various timeframes and divergence analyses. Although not without their limitations, indicators provide traders with methodical frameworks for deciphering market data and navigating the intricate realm of financial markets. Traders frequently formulate strategies that are in accordance with their trading preferences and objectives by utilizing a combination of indicators.

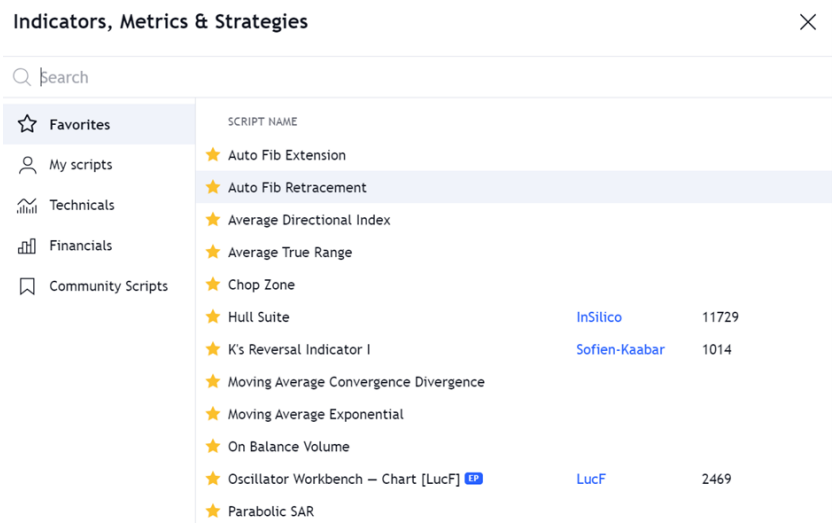


Figure 7.1: Indicators

7.1 Indicators Used in Algorithmic Trading Bot

Chapter 5 delves into an examination of the operational dynamics of our state-of-the-art trading bot. Within this complex realm, careful integration of critical indicators and features elevates the bot's capacity for effective decision-making. The strategic incorporation of these metrics is crucial, as it provides our algorithm with the flexibility required to navigate the complex environment of financial markets. The Relative Strength Index (RSI) is an essential component of our bot's repertoire, serving as a critical indicator of market momentum. The oscillator, which is constructed using recent price fluctuations as its basis, functions as a governing principle, aiding the algorithm in detecting periods of market depletion or expansion. The algorithm is guided by its insights to determine the most advantageous times to enter and depart, avoiding overbought situations and capitalizing on opportunities that arise in oversold conditions. When used in conjunction with the RSI, the Moving Average Convergence Divergence (MACD) enhances the analytical capabilities of our program. MACD provides nuanced insights into the intensity of trends and possible reversals through the monitoring of convergence and divergence for two moving averages. The provision of such detailed information enables the algorithm to dynamically adjust its strategies, thereby ensuring that it maintains a favourable position to exploit changing market conditions. The analytical capabilities of our algorithm are enhanced even further by the exponential moving average (EMA), which smooths a representation of recent price fluctuations. This functionality empowers the

automaton to accurately detect and track trends, thereby cultivating an intricate comprehension of market dynamics. Equipped with this comprehension, our automated system adeptly reacts to fluctuations in sentiment and pricing patterns, thereby guaranteeing a proactive stance towards market advancements. The oscillator functionalities of Williams%R augment the level of complexity incorporated into our bot's strategy. Through the assessment of overbought or depressed conditions, Williams%R aids in the identification of prospective market turning points. By strategically utilizing this information, our algorithm refines its approach to optimize transaction entry and exit points, thereby improving the overall performance of the portfolio. Our trading algorithm also utilizes stochastic oscillators to detect potential trend reversals and Exponential Moving Averages (EMAs), which identify trends and manage risk in addition to the indicators. The amalgamation of these attributes results in an all-encompassing and dynamic algorithmic approach. As our narrative progresses, we shall further explore the complexities inherent in the algorithmic strategies employed by our trading machine. We shall observe the astute integration of these indicators and characteristics into a trading strategy that is both dynamic and adaptable to navigate the perpetually shifting market environment effectively. By adopting this deliberate and strategic methodology, we not only demonstrate the technical prowess of our algorithm but also emphasize its ability to withstand the intrinsic intricacies of financial markets. Our trading algorithm will be put into practice in the following chapters, utilizing this powerful amalgamation of indicators and functionalities to carry out transactions, mitigate risk, and surpass industry standards. Our forthcoming presentation will showcase the practical implementation of our bot's mathematical capabilities. It will precisely, nimbly, and strategically navigate the financial markets. Prepare for an enlightening voyage.
These are the major features used in algorithmic bots for automation. Most of the calculations based on these features are very essential for bots.

7.1.1 RSI

The Relative Strength Index (the RSI), which J. Welles Wilder originally developed, is a critical momentum oscillator that is extensively utilized in the financial sector to evaluate the magnitude and speed of price fluctuations. Its scale, which spans from zero to one hundred, provides traders and analysts with invaluable insights. In conventional wisdom, [181]an RSI value exceeding 70 indicates conditions of overbought Ness, which may present favourable circumstances for selling, whereas an RSI value falling below 30 indicates conditions of overboldness, which may offer favourable circumstances for purchasing. The versatility of the RSI framework in detecting future trend reversals is enhanced by the fact that signal generation within it en-

tails the identification of convergence and failure movements. In addition to its traditional application, the RSI works as a dynamic instrument to interpret more extensive market patterns.
Furthermore, this flexibility encompasses the capacity to modify conventional overbought and overpriced thresholds (between 70 and 30) to correspond with the distinctive attributes of a particular security more precisely. Significantly,[182] the RSI may exhibit prolonged lingering in overbought or oversold regions during robust trends, serving as an indicator of the market's sustained strength or weakness. In addition to providing numerical insights, the RSI frequently unveils chart patterns, including trendlines and double peaks and bottoms, that may take time to be evident on the main price data. By identifying levels of resistance or support on the RSI, its analytical depth is further enhanced. The RSI fluctuates between 40 and 90 during a rising trend or bullish market, with the 40-45 zone serving as support. In a bear market or downtrend, the RSI fluctuates predominantly between 10 and 60, alongside resistance between 50 and 60. These ranges fluctuate in accordance with RSI configurations and the fundamental momentum of the security or market. The application of the RSI transcends the straightforward analysis of overbought and depressed conditions. The identification of prospective reversals is facilitated by the identification of divergences that arise between RSI readings and underlying prices. The occurrence of divergence between fresh highs or lows and the RSI indicates the possibility of a price reversal. Significantly, the predictive capabilities of the RSI are further enhanced through the identification of Top Swinging Failures and Bottom Swing Mistakes, which are distinguished by the RSI reaching lower highs or higher lows after price movements in the other direction. The RSI calculation formula is represented as:

$$\text{RSI} = 100 - \left[100 \div \left(1 + \left(\frac{\text{Average of Upward Price Change}}{\text{Average of Downward Price Change}}\right)\right)\right] \quad (7.1)$$

Calculates the mean profits and losses during a designated time frame, yielding a percentage that signifies the proportional magnitude of recent upward and downward price fluctuations. The true profundity of this formula is achieved via practical application and meaning, as illustrated most effectively in Wilder's book through real-world examples. Fundamentally, the RSI is an adaptable and potent tool that provides intricate observations regarding trends, reversals, and chart configurations; as such, it is an essential element of technical analysis of financial markets.

7.1.2 MACD

The Moving Average Convergence/Divergence (MACD), or MAC-D, reveals the relationship between the two Exponentially Moving Averages

Figure 7.2: RSI

(EMAs)[183] that represent the price of a security. It is a powerful momentum indicator that follows trends. Subtracting the 12-period EMA from the 26-period EMA yields the MACD line. Central to the MACD indicator is this computation, which depicts the momentum characteristics of a given security. A signal line is generated in conjunction with the MACD line by superimposing the nine-day moving average (EMA) of the MACD line onto the MACD display. Catalyzing purchase or sell orders[184], the signal line influences traders' decisions significantly. Traders who implement the MACD frequently base their decisions on crossovers that occur between the line of the MACD and the signal line. A bullish signal is generated when the MACD line crosses above the signal line, indicating the presence of prospective purchasing opportunities. On the contrary, a bearish signal is generated when the line formed by the MACD crosses below the signal line, suggesting possible opportunities for selling or entering short positions. In the field of securities trading, well-informed decisions are predicated on

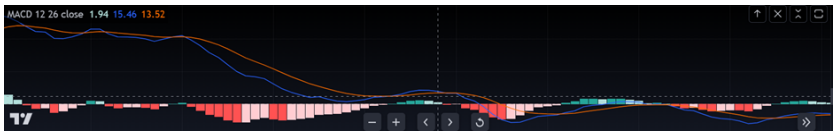


Figure 7.3: MACD

the constantly changing relationship between these two lines. MACD indicators offer a plethora of interpretive opportunities, where the most prevalent signals consist of crossovers, divergences, and rapid fluctuations. Signal-line crossovers between the MACD line, as well as an MACD line, indicate momentum shifts and possible trend reversals. Divergences are indicative of a trend that is waning when the MACD and price exhibit contrasting movement directions. The MACD line's rapid ascents and descents serve to emphasize abrupt shifts in momentum, providing traders with guidance on detecting possible volatility or acceleration of trends. The MACD, through the interaction of the MACD line or signal line, offers traders a comprehensive instrument to evaluate the momentum and possible trend trajectories of a given security. The versatility of this tool in analyzing crossovers, divergences, and variations contributes to its efficacy in guiding trading decisions in the ever-changing realm of financial markets.

$$\text{MACD} = \text{12-Period EMA} - \text{26-Period EMA} \tag{7.2}$$

7.1.3 EMA

A variant of the Moving Average (MA), the Exponential Moving Average (EMA), gives more importance and urgency to recent price fluctuations than the simple moving average (SMA), which gives an equal amount of weight to every observation during the period. A crucial parameter in EMA calculations, the smoothing factor establishes the significance attributed to recent observations; it is frequently configured to 2 to give precedence to the most recent data. [185]By augmenting the filtering factor, the impact of recent observations is magnified, thereby enhancing the responsiveness of the EMA. The procedure of determining the EMA entails waiting until the designated number of observations (e.g., twenty days) is accumulated, which is one more than the SMA. Determining the SMA is a simple computation that requires dividing the aggregate of closing prices during the designated period by the total number of observations. The formula for the filtering multiplier of the EMA is [2 ÷ (number of observations + 1)]. As an illustration, the multiplier of a 20-day moving average is [2/ (20+1)] = 0.0952.

$$\text{EMA} = \text{Closing price} \times \text{multiplier} + \text{EMA (previous day)} \times (1 - \text{multiplier}) \tag{7.3}$$

The significance of each data point is determined by this multiplier, which gives greater weight to more recent prices than to older ones. EMAs are adaptable; in place of closing prices, one may substitute open, high, low, or median values. The 12-day and 26-day moving average EMAs serve as foundational elements for technical analysis indicators such as the per cent Price Oscillator (PPO) and the Moving Average of Convergence Divergence (MACD). The Fifty day and two Hundred day moving averages are frequently used to evaluate longer-term trends; a stock price breach of the 200-day moving average is regarded as an indication of a reversal. Technical analysis users recognize the significance of moving averages but exercise prudence regarding possible misinterpretations. Although moving averages (EMAs) function as effective lagging indicators, their primary utility lies in confirming market movements or quantifying the intensity of a trend. The opportune moment can enter a market to pass before a moving average reflects a trend change. By attributing greater significance to more recent data[186], EMAs partially alleviate the latency effect, thereby yielding an indicator that is more responsive and seamless. This is especially beneficial when EMAs are utilized to generate trading entry signals. For exhaustive trend analysis, traders must prioritize the examination of both the rate of change from one bar to the next and the direction of the EMA line. EMAs are potent instruments for

technical analysis, providing adaptability and promptness in identifying market trends. Traders can better employ EMAs, particularly in markets that are trending, by comprehending the intricacies of EMA computations and deciphering their ramifications for market dynamics.

$$\text{EMA}_{\text{Today}} = (\text{Value}_{\text{Today}} \times (1 + \text{DaysSmoothing})) + \text{EMA}_{\text{Yesterday}} \times (1 - (1 + \text{DaysSmoothing})) \tag{7.4}$$

7.1.4 William % R

The Williams Percent Range, or Williams%R, is an influential momentum indicator utilized in financial markets to provide valuable insights into situations characterized by overbought and oversold conditions. This metric, devised by Larry Williams, spans a value from 0 to -100 and serves as an indicator to evaluate the present value of a security in relation to its highest-to-lowest range during a designated time frame, typically fourteen days. For the calculation, the highest and lowest prices for each of the fourteen periods are recorded, with particular attention paid to the current price, the highest price, as well as the lowest price in the fourteenth and subsequent periods. Williams%R indicates overbought conditions when it fluctuates between -0.0 and -1.0, suggesting that the price is approaching the peak of its most recent range. On the contrary, an indicator value ranging from -80 to -100 indicates oversold conditions, wherein the price is significantly below its most recent peak. Its operation and implementation are identical to those of the stochastic oscillator. Traders utilize William%R to determine possible entry and departure positions in the market. It can be significant to observe the indicator declining below -80 during an uptrend. A subsequent increase in price, with the indicator re-crossing above -80, could potentially indicate that the uptrend has resumed. On the contrary, during a downtrend, traders closely observe the indicator as it ascends above -20, anticipating a price decline accompanied by the Williams%R retracing below -20, which would signify a possible extension of the downtrend. Additionally, momentum failures can be measured with Williams%R. An indicator of a strong uptrend generally attains a value of -20 or greater. If the indicator descends without regaining -20 prior to a subsequent decline, it could indicate that the upward momentum of prices is encountering difficulties, which could potentially result in a more significant decrease in price. Like a downtrend, it is not uncommon to observe readings of -80 or less. It may portend an impending price increase if the price chart fails to ascend from these low levels prior to doing so. It is crucial to acknowledge that indications of overbought and oversold conditions do not necessarily portend a reversal; conversely, overbought conditions may validate an uptrend, which is consistent with the notion that a

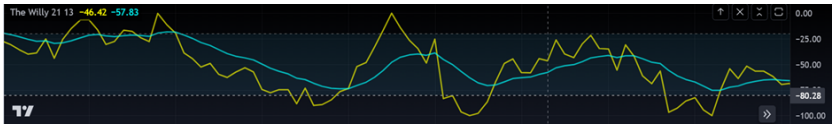


Figure 7.4: William % R

strong upward price movement frequently propels prices to or beyond previous peaks. Even with its practicality, the Williams%R may subject one to spurious signals and an overemphasis on the previous 14 periods. Over time, traders should consider possible differences in indicator movements and real price changes. In essence, Williams%R serves as a versatile instrument for traders, aiding them in the navigation of market conditions through the identification of overbought and oversold conditions and the prediction of possible price momentum shifts.

$$\text{Williams \%R} = \frac{\text{Highest High} - \text{close}}{\text{Highest High} - \text{Lowest Low}} \tag{7.5}$$

Where "Highest High" denotes the price at which the item peaked during the lookback period, which is usually fourteen days.
The most recent closing price is denoted by close.
Lowest Low = Price at which a lookback period, usually fourteen days, concluded.

7.1.5 CCI

The Commodity Channel Index (CCI), which Donald Lambert devised, is an oscillator based on momentum that is employed to evaluate whether an investment vehicle has entered an excessively bought or sold state. By analyzing the intensity and direction of price trends, this technical indicator equips traders with the knowledge necessary to make well-informed decisions concerning trade entries, exits, or modifications to current positions. Typically, twenty periods are analyzed by the CCI to monitor the peak, low, and close prices. The average price and moving average (MA) of these average prices are subsequently computed, and the mean deviation is then determined. The CCI reading that is obtained serves as a dynamic instrument utilized to identify trends, overbought or oversold conditions, and trend weaknesses by analyzing price divergences. A transition of the CCI from low or near-zero values to values above 100 may indicate the beginning of an uptrend; traders should be on the lookout for a price pullback that follows a rally, which would indicate an advantageous time to purchase.
On the contrary, during the onset of a downtrend, a transition in positive or near-zero measurements to values below -100 indicates the possibility of

a downtrend starting. This causes traders to contemplate selling long positions or investigating opportunities for shorting. The levels of overbought and oversold conditions are not fixed, as the CCI is unbound. Traders utilize prior readings to estimate reversal points, which can differ across assets. Additionally, divergences are identified by the CCI when price movement contradicts the indicator. For example, a rising price accompanied by a falling CCI could indicate a deterioration in the trend. Despite not being a highly significant trade signal, divergence functions as an alert to traders, compelling them to increase stop-loss thresholds or abstain from initiating new positions in the direction of the current trend. To summarize, the CCI functions as a highly beneficial instrument for traders, furnishing real-time evaluations of price movements, overbought and oversold illnesses and trend vulnerabilities that yield insightful observations of market conditions and prospective trading prospects. [1]

$$\text{CCI} = \frac{\text{Typical Price} - \text{MA}}{.015 \times \text{Mean Deviation}} \tag{7.6}$$

$$\text{Typical Price} = \sum_{i=1}^{P} ((\text{High} + \text{Low} + \text{Close}) \div 3)$$
$$P = \text{Number of periods}$$
$$\text{MA} = \text{Moving Average}$$
$$\text{Moving Average} = (\sum_{i=1}^{P} \text{Typical Price}) \div P$$
$$\text{Mean Deviation} = \sum_{i=1}^{P} (|\text{Typical Price} - \text{MA}|) \div P$$

7.1.6 Chande Momentum Oscillator

Established by Tushar Chande in 1994, the Chande Momentum Oscillator is a highly esteemed technical indicator utilized extensively in the field of market analysis. The metric performs an essential role in assessing market momentum through the computation of the differential between recent gains and losses over a specified time period, denoted as N periods. The Chande Momentum Oscillator, in contrast to the stochastic oscillator and Wilder's Relative Strength Index (RSI), does not employ averaging on its outputs. This attribute renders it more susceptible to frequent oscillations, thereby providing traders with a more discerning outlook on both bullish and bearish price fluctuations. The Chande Momentum Oscillator, which has an operational range of +100 to -100, offers traders a numerical depiction of the dominant momentum. This information assists them in discerning potential instances of overbought or oversold conditions within the market.

[1] https://www.investopedia.com

When the oscillator exceeds +100, it signifies strong upward momentum, which may suggest an overbought situation. A reading below -100, on the other hand, could indicate a substantial downward momentum and indicate a possible oversold condition. Traders frequently interpret these extraordinary readings as indicators of possible market sentiment shifts or reversals. The Chande Momentum Oscillator provides traders with a dynamic instrument for assessing the magnitude and trajectory of market momentum. This information is of significant value as it has the potential to impact their trading choices. Having a straightforward numerical representation and sensitivity to short-term price fluctuations, it is an adaptable and readily available element of technical analysis in the financial markets.

$$Chande\ Momentum\ Oscillator = \frac{sH - sL}{sH + sL} \times 100 \tag{7.7}$$

Where:
sH = the sum of N periods' worth of higher closes
sL = the aggregate of N periods' lower closures

7.1.7 Stochastic

[2] Since its inception by George Lane in the late 1950s, the Stochastic Oscillator has emerged as a standard momentum indicator in technical analysis. By examining the position of a stock's closing price in relation to its high and low prices over a designated time, usually 14 days, this oscillator provides valuable information. Lane underscores the notion that the Stochastic Oscillator emphasizes the velocity or momentum of price fluctuations rather than solely monitoring price or volume. Its objective is to predict shifts in the trajectory of a stock's price, thereby establishing itself as a valuable instrument for detecting possible reversals. The range-limited characteristics of the stochastic oscillator are evident, as it perpetually vervains between 0 and 100. This attribute makes it exceptionally practical for identifying instances of excessive buying and selling in the market. Readings above 80 have historically indicated overbought conditions, whereas readings below 20 have indicated oversold conditions. Nonetheless, it is imperative to acknowledge that robust trends may sustain protracted periods of these conditions. Traders ought to be vigilant regarding fluctuations in the oscillator to obtain valuable insights regarding possible alterations in the dynamics of trends. When graphing the Stochastic Oscillator, it is customary to have two lines present: one that signifies the oscillator's true value for every session and the other that illustrates its three-day simple moving average. A potential reversal signal is formed at the intersection of these lines, which signifies a substantial change in momentum. In addition, price action divergence relative to the Stochastic Oscillator

[2] https://www.investopedia.com

is considered a significant reversal signal. The notions of %K and %D are introduced in The Stochastic Oscillator, with %K being frequently denoted as the rapid stochastic indicator. The "slow" stochastic indicator, denoted as %D, smoothest the internal dynamics by incorporating a 3-period moving average of %K. Transaction signals are generated when the three-period moving average, denoted as %D, is traversed by %K. An integral distinction can be made between the slow and rapid stochastic oscillators due to the inclusion of a lowering period in percent K, which regulates internal smoothing. In essence, the Stochastic Oscillator offers traders a multifunctional instrument for assessing momentum, predicting possible reversals, and navigating conditions of overbought and oversold markets. By analyzing intersections, divergences, and the correlation between %K and %D, traders are provided with insightful information that enables them to make well-informed trading decisions.

$$\%K = \frac{(C - L14)}{H14 - L14} \times 100 \tag{7.8}$$

Here
C represents the latest closing price.
L14 = the price at which trading commenced fourteen sessions prior.
H14 represents the 14-day high price traded, while %K denotes the present value of the stochastic indicator.

7.1.8 Stochastic RSI

An indicator of technical analysis, the Stochastic RSI (StochRSI) applies the same principles as the Stochastic oscillator but to the relative strength index (RSI). The StochRSI, which ranges from zero to one or, on specific platforms, from zero to one hundred, utilizes a collection of RSI values calculated using the stochastic oscillator formula in lieu of conventional price data. This methodology offers traders valuable insights regarding whether the present RSI value signifies an overbought or oversold market condition. The StochRSI, which Tushar S. Chande and Stanley Kroll devised, excels in sensitivity and signal generation in comparison to conventional indicators. As a result, it provides a customized evaluation of the past performance of a particular security. RSI levels are recorded over a specified period, typically fourteen, and subsequently utilized in the StochRSI formula for the computation. When the value of the StochRSI falls below 0.20, it indicates that the security is oversold and may experience an upward trend.
Conversely, a reading above 0.80 indicates that the security may be overbought and may experience a pullback. Moreover, when considered within the framework of an oscillator featuring a centerline at 0.50, the StochRSI can be utilized to discern transient patterns. Values greater than 0.50 may be indicative of a positive trend, whereas values less than 0.50 indicate a negative

trend. To optimize its efficacy, it is recommended to employ the StochRSI in conjunction with additional technical indicators or chart patterns, owing to its tendency to produce a substantial quantity of signals. Non-momentum oscillators, including the accumulation distribution line, have the potential to supplement the StochRSI by offering an alternative set of insights. A valuable instrument for traders seeking a nuanced comprehension of a security's momentum and possible trend reversals, the StochRSI is characterized by its versatility.

$$StochRSI = \frac{\max[\text{RSI}] - \min[\text{RSI}]}{\text{RSI} - \min[\text{RSI}]} \tag{7.9}$$

Where:
RSI = Present RSI reading minimum [RSI] = Lowest RSI reading in the previous fourteen periods (or the lookback interval of your choosing)
Maximum RSI value from the previous 14 periods (or your preferred lookback interval)

7.1.9 MFI

The Money Flow Index (MFI) is a technical oscillator specifically engineered to detect divergences that may signify upcoming price trend changes in an asset and identify overbought or oversold signals. In contrast to conventional oscillators like the Relative Strength Index (RSI), the MFI is distinguished by its amalgamation of price and volume data, which gives it the nickname "volume-weighted RSI." The process entails ascertaining the average price for each of the previous fourteen periods, classifying the average price as either higher or lower than the previous period, and subsequently multiplying the average price by the corresponding volume to compute raw money flow. To calculate the Money Flow Ratio, add the positive money flows from the previous 14 periods and divide by the negative money flows. The ratio is utilized in the computation of the Money Flow Index (MFI), a variable that varies from 0 to 100. One notable utilization of the MFI is in the detection of divergences, which occur when the oscillator deviates from the prevailing price trend and provides indications of possible reversals. For example, a potential downside reversal is indicated by a high MFI falling below 80 as the security continues to rise, whereas a potential upside reversal is suggested by a low MFI rising above 20 during a security sell-off. Traders also identify greater divergences by analyzing price and MFI waves in succession. The price reaching consecutive peaks while the MFI registers troughs of lower magnitudes could portend a downturn. Overbought and oversold conditions (above 90 and below 10, respectively) indicate possible trading opportunities. For instance, an MFI falling below ten could indicate an opportunity for long trade, whereas a move above 90 could indicate a possibility for short

trade. Furthermore, fluctuations in the proximity of overbought or oversold conditions can furnish traders with valuable indications of impending trend reversals, enabling them to formulate well-informed judgments amidst a multitude of market circumstances.

$$Money\ Flow\ Index = 100 - \frac{100}{1+Money\ Flow\ Ratio} \tag{7.10}$$

$$Money\ Flow\ Ratio = \frac{14\ Period\ Positive\ Money\ Flow}{14\ Period\ Negative\ Money\ Flow} \tag{7.11}$$

$$Typical\ Price = \frac{High\ +\ Low\ +\ Close}{3} \tag{7.12}$$

7.1.10 True Strength Index

A technical momentum oscillator, the True Strength Index (TSI) detects price divergence, overbought and oversold conditions, trends, and prospective reversals in an asset. To determine the TSI, exponential moving averages (EMAs) are computed for both absolute and price variations. After obtaining the double-smoothed EMAs, the TSI value is computed using them. The principal functions of this indicator are to detect overbought and oversold conditions, to emphasize short-term momentum via signal line crossovers, and to identify trend changes via centerline crossovers. Overbought and oversold conditions on the TSI are dependent on the asset being analyzed; therefore, traders must annotate extreme TSI levels that are particular to that asset. Buy or sell decisions are not invariably prompted by overbought or oversold readings in isolation. Traders generally observe additional indicators, such as a decline in price or the TSI, before acting on these factors. Although signal line crossovers are frequent, they must be implemented in conjunction with other TSI signals. They are produced when the TSI line crosses the signal line. For example, when the TSI is situated above the centerline, purchase signals may exhibit greater reliability, whereas sell signals might be preferred in overbought conditions. Another signal generated by the TSI is that centerline crossovers indicate positive momentum when the point is above zero and negative momentum when it is below zero. To directional bias, traders may consider entering long positions when the indicator is above the centerline and short positions when it falls below.
Divergence and breakouts are supplementary instruments furnished by the TSI. The TSI's support and resistance levels can serve as indicators for breakouts and changes in price momentum. Divergence, which occurs when the price and TSI exhibit contrasting movements, could potentially signify a reversal. However, divergence should be utilized in conjunction with other TSI

signals or technical indicators, as it is regarded as an unreliable timing signal. Notwithstanding its practicality, the TSI possesses certain constraints. It is not uncommon for false signals to occur, whereby the trajectory of the indicator may alter without an accompanying change in price. Frequent signal line crossovers require further filtration to ensure the reliability of trading decisions. Divergence may also exhibit unreliability, persisting for extended durations while failing to forecast tangible reversals. Traders should exercise prudence and supplement TSI signals with additional analytics to adopt a more comprehensive stance.

$$TSI \ = \ (PCDS/APCDS) \ \times \ 100 \tag{7.13}$$

$$\begin{aligned}
&PC \ = \ CCP \ - \ PCP \\
&PCS \ = \ 25 - period \ EMA \ of \ PC \\
&PCDS \ = \ 13 - period \ EMA \ of \ PCS \\
&APC \ = \ AVCCP \ - \ PCP \\
&APCS \ = \ 25 - period \ EMA \ of \ APC \\
&APCDS \ = \ 13 - period \ EMA \ of \ APCS \\
&where: \\
&PCDS \ = \ PC \ double \ smoothed \\
&APCDS \ = \ Absolute \ PC \ double \ smoothed \\
&PC \ = \ Price \ change \\
&CCP \ = \ Current \ close \ price \\
&PCP \ = \ Prior \ close \ price \\
&PCS \ = \ PC \ smoothed \\
&EMA \ = \ Exponential \ moving \ average \\
&APC \ = \ Absolute \ PC \\
&APCS \ = \ Absolute \ PC \ smoothed
\end{aligned} \tag{7.14}$$

7.1.11 Ultimate Oscillator

The Ultimate Oscillator, a distinctive technical indicator created by Larry Williams in 1976, is utilized to assess price momentum across various periods. The Ultimate Oscillator, in contrast to other oscillators that depend on a solitary timeframe, employs a weighted average of three timeframes to furnish an indicator that is both more dependable and less susceptible to volatility. The implementation of a multi-timeframe architecture in the oscillator mitigates the occurrence of spurious divergences, which is a prevalent concern with timeframe-only indicators. In order to calculate the indicator, Buying Pressure (BP) and True Range (TR) for each period are determined, BP and TR Sums are computed over those periods, and averages are calculated for periods 7, 14, and 28. The Ultimate Oscillator, which exhibits

a value range of 0 to 100, shares similarities with the Relative Strength Index (RSI) in that it detects overbought conditions exceeding 70 and oversold conditions falling below 30.
Larry Williams developed the Ultimate Oscillator as a solution to the prevalent issue of spurious divergences observed in alternative oscillators. These deceptive divergences are produced by misaligned price and oscillator readings, which result from price spikes followed by oscillator declines. Therefore, erroneous signals are generated. In order to generate buy signals, Williams proposed a three-step process: first, a bullish divergence must occur, wherein the price attains a lower low than the indicator registers a higher low; second, the initial low in the divergence must be below 30, which signifies an oversold condition; and third, a potential upside reversal must be indicated by the Ultimate Oscillator as it ascends above the divergence high. In contrast, sell signals are characterized by a bearish divergence, in which the price reaches a peak while the indicator demonstrates a trough. A divergence must exhibit a first high exceeding 70 to signify overbought conditions, while the Ultimate Oscillator must descend below the divergence low to indicate the possibility of a downside reversal. By leveraging the distinctive multitimeframe attributes of the Ultimate Oscillator, this tripartite approach empowers traders to render well-informed judgments, thereby augmenting the discerning comprehension of momentum and diminishing the probability of erroneous signals. [3]

$$UO = \frac{[(A_7 \times 4) + (A_{14} \times 2) + A_{28}]}{4 + 2 + 1} \times 100 \tag{7.15}$$

[3] https://www.investopedia.com

where:
UO = Ultimate Oscillator
A = Average
Buying Pressure (BP) = Close − Min(Low,PC)
PC = Prior Close
True Range (TR) = Max(High,Prior Close) − True Range (TR) = Min(Low,Prior Close)

$$\text{Average}_7 = \frac{\sum_{p=1}^{7} BP}{\sum_{p=1}^{7} TR}$$
$$\text{Average}_{14} = \frac{\sum_{p=1}^{14} BP}{\sum_{p=1}^{14} TR}$$
$$\text{Average}_{28} = \frac{\sum_{p=1}^{28} BP}{\sum_{p=1}^{28} TR} \tag{7.16}$$

Certainly indeed! While I have extensively examined several technical indicators, it is crucial to acknowledge that the development of trading algorithms employs a vast array of indicators. An indicator functions for a distinct objective when it comes to the examination of price fluctuations, market trends, and overall conditions. When utilized in conjunction, these indicators enhance the overall analysis of market conditions, enabling trading algorithms to generate well-informed decisions by considering diverse technical facets of price fluctuations and trends. Every indicator possesses unique advantages and disadvantages, and their efficacy is frequently contingent upon market conditions and the trading approach implemented.

7.1.12 Fisher Transform

Established by John F. Ehlers, the Fisher Transform Indicator is a highly effective instrument in the field of technical analysis, distinguished for its capability of transforming prices into a Gaussian normal distribution. This conversion facilitates the detection of possible pivotal moments in asset valuations through the prioritization of outliers determined by recent price fluctuations. To determine the indicator's distinctive methodology, a lookback period, usually consisting of nine periods, is chosen, and complex calculations are executed on the prices observed during this period. The values obtained, which fall within the range of -1 to +1, are subsequently subjected to a sequence of mathematical operations, which includes multiplication by the natural logarithm, in order to produce the Fisher Transform values. The principal aim of this indicator is to generate a Gaussian normal

distribution, which deviates from the conventional market price distribution. This indicator was devised by the eminent trader and engineer John F. Ehlers to overcome the difficulties associated with non-normally distributed data, including market prices. The Fisher Transform effectively emphasizes infrequent peak movements, thereby enabling a more precise detection of prospective price reversals depicted on a chart. When employing the Fisher Transform Indicator, traders generally pursue leading signals as opposed to lagging ones to comprehend asset price fluctuations and current market conditions. The Fisher Transform's unbounded characteristics suggest that extreme values may endure for a prolonged duration. These extremes, which frequently differ across various assets, function as critical indicators of possible reversals in price trends. A high value, such as seven or eight, or a low value, such as -4, is interpreted considering the historical data of the asset. In order to effectively implement the Fisher Transform Indicator in trading, professionals frequently seek out extreme readings that indicate the possibility of a reversal. This reversal is officially confirmed when the indicator undergoes a direction change. For example, a decline in the indicator after its peak value may indicate a forthcoming reduction in price. Furthermore, it is common for the Fisher Transform Indicator to include a signal line, which is commonly a moving average of the value of the indicator. Traders frequently employ the intersection of the indicator and its trigger line as a trade signal, which serves to improve the process of determining whether to enter or exit positions. Notwithstanding its efficacy, traders exercise prudence when employing the Fisher Transform Indicator on account of the innumerable signals it produces. A considerable number of traders favour combining this indicator with trend analysis and selectively employing it to generate buy and sell signals across various market conditions. For example, the indicator can serve as a guide for purchasing decisions during an uptrend and a valuable tool for identifying opportune points to cover positions and short-selling signals during a downtrend.

The following features are used in our algorithmic bot. There are many other features (indicators) used in our algorithmic trading bot. The algorithmic trading bot is discussed in detail below. The development and implementation of our algorithmic trading, as well as the working of algorithmic trading, will be discussed.

7.2 Algorithmic Trading Bot

The trading algorithm functions by utilizing a strategic framework that is constructed by aggregating the computations of various indicators. The indicators, each fulfilling a unique function, such as evaluating volatility or

ascertaining the direction of trends, synergistically enhance the bot's all-encompassing market analysis. The strategy of the algorithm, which is comprised of a collection of principles derived from these indicator-based insights, is fundamental to its operation. A purchase signal may be generated by the algorithm, for instance, during periods of high volatility and an uptrend. The bot's functionality is fundamentally dependent on continuous learning as it iteratively adjusts its approach in response to updated market data and ongoing indicator computations. The implementation of risk management principles, such as stop-loss orders and position size determination, is incorporated into the strategy to mitigate potential losses. Automation is a fundamental characteristic that empowers the bot to independently implement transactions and promptly adapt to market fluctuations without relying on irrational judgment. Prior to implementing the algorithm into production, developers perform comprehensive backtesting and optimization to verify its efficacy across various market conditions. The signal represented

Figure 7.5: CandleStick Chart of Bot

by the cyan colour in the output of the algorithm is of the utmost importance in directing traders through the intricacies of market trends. The discernible hue of teal signifies a favourable trajectory, thereby signifying a market-wide optimistic sentiment. Nevertheless, teal distinguishes itself through its subtlety; it implies the existence of a bullish trend that, although present, may not be as strong or dominant as alternative bullish signals that are more conspicuous. Traders attempting to decipher the significance of the teal hue should proceed with prudence, as it signifies that the favourable forces of the market are operational, albeit potentially weaker than those associated with more prevailing trends. The colour teal, which is utilized in the bot's output, encourages traders to approach the opportunities presented with a measured sense of optimism. This promotes the adoption of a strategic methodology, compelling participants in the market to further examine supplementary in-

dicators and market factors to verify and strengthen their trading judgments. The nuanced hue of teal serves to underscore the significance of conducting a thorough evaluation, instructing traders against depending exclusively on a solitary indicator and instead advocating for a holistic assessment of market conditions. Within the expansive realm of financial markets, the teal signal serves as a navigational aid, providing valuable indications of prospective price increases. This visual cue encourages traders to investigate the poten-

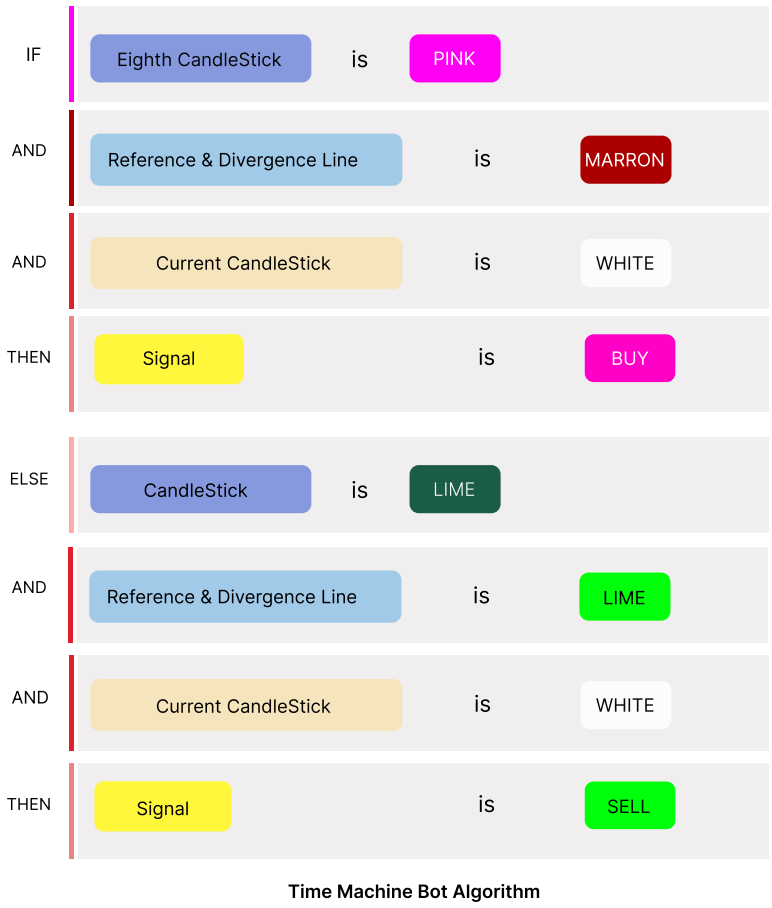


Figure 7.6: Step By Step Time Machine Bot

tial for bullish outcomes while maintaining an eye out for the subtleties that distinguish it from more reliable bullish signals. Like a dynamic terrain, the market necessitates traders to adjust accordingly. In this regard, teal plays a crucial role in elucidating the complex interplay between vendors and purchasers. Traders are reminded of the multifaceted nature of trends through the teal hue in the bot's output as they navigate the dynamic nature of financial markets. This inquiry stimulates an examination of the diverse levels of optimistic sentiment, in which the colour teal signifies a moderate propen-

sity for favourable market dynamics. This nuanced methodology is consistent with the dynamic characteristics of financial markets, which recognize that the intensity of trends can vary over time.

```
    ////////////////////////////////////////
buyCondition = oscLineColor[8] == PINK and oscLineColor[7]
== PINK and oscLineColor[6] == PINK and
oscLineColor[5] == PINK and oscLineColor[4] == PINK
and oscLineColor[3] == PINK and oscLineColor[2] == PINK
and oscLineColor[1] == PINK and
oscLineColor[0]==MAROON and barColor[0] == WHITE
and barColor[1]==PINK

sellCondition = oscLineColor[2]==LIME
and oscLineColor[1] ==LIME and
oscLineColor[0]== TEAL
and barColor[0] == WHITE and barColor[1]==LIME

////////////////////////////////////

if buyCondition
    entry_price:=close
    strategy.entry("Buy", strategy.long)
    entryLine:=line.new(bar_index,entry_price,bar_index,
    entry_price,color=color.yellow)
    entryLabel := label.new(bar_index,entry_price,
    text="BP : "+
    str.tostring(entry_price,format.mintick),
    color=color.yellow,style =
    label.style_label_left,size=size.small)
if sellCondition
    entry_price:=close
    strategy.entry("Sell", strategy.short)
```

The incorporation of reference lines and divergence lines into our bot's analysis provides an additional level of complexity, enabling traders to further examine the complexities of market trends. The notion is predicated on the intricate interplay between these lines and the hues of candles, which establishes an all-encompassing structure for the process of making decisions. A crucial component of our bot's arsenal, the divergence line illustrates the dynamic correlation between indicators and price fluctuations. Upon thorough examination, this line offers crucial insights regarding the potential for trend reversals or the robustness of current trends. Traders utilize its color, which

is reminiscent of maroon, pink, or citrus candles, as a visual indicator to decipher market sentiment. A further extension of the divergence line from the reference line, regardless of whether it is pointing bullishly or bearishly, indicates the robustness and durability of the dominant trend. The conviction that drives market movements is strengthened by a greater distance, which serves as a quantitative indicator of the robustness of trends. This functionality provides traders with a valuable instrument to assess the strength and direction of the market trend. The dynamic tango between the reference line and divergence line reflects the perpetual fluctuations that occur in market dynamics. When these lines intersect or cross, they transmit vital indications of possible shifts in the trend. As these crossover instances signify critical junctures where the market narrative may transform, traders closely monitor them. Traders are compelled to reevaluate their strategies and positions at this juncture, recognizing the dynamic characteristics of market trends. The

Figure 7.7: Divergence and Reference Line

integration of line and candle colour synchronization further enhances the sophistication of the analysis performed by our algorithm. Candles coloured lime correspond to a pronounced bullish sentiment, candles coloured maroon indicate a bearish tone and candles coloured pink emphasize a particularly robust bearish trend. The alignment described above improves the legibility of signals, thereby establishing a visual language that enables traders to comprehend and make decisions promptly. The fundamental function of the distance between the reference line and the divergence line is to provide traders with a quantitative assessment of market momentum by functioning as a dynamic indicator of trend strength. The crossing of these lines serves as an indicator of possible changes in trend, compelling traders to modify their approaches correspondingly. By incorporating candle colours into this framework, the analysis is enhanced, and a more comprehensive perception of market sentiment is achieved. Traders utilize our bot's cutting-edge method-

ology, which integrates reference lines, divergence lines, and candle hues, to navigate the intricate realm of financial markets effectively. This enables them to make informed trading decisions amidst the constantly changing price environment.

The code provided contains a purchase condition which specifies a collection of logical criteria that instruct the algorithmic trading program on when to produce a buy signal. This stipulation conducts a thorough analysis of the colour patterns exhibited by the reference line and divergence line (previously known as oscLineColor), as well as the candlestick colours (previously referred to as barColor), throughout various periods in history. The specific emphasis is on the final nine periods (numbered from 0 to 8) of the divergence line.

In each of these periods, a uniform PINK colour is expected, except for the

Figure 7.8: Buy Condition

most recent period (index 0), which should be rendered in MAROON. Furthermore, in accordance with the condition, the present candlestick (index 0) must be WHITE in colour, whereas the preceding candlestick (index 1) must be PINK in colour. The combination of these rigorous criteria establishes the prerequisites for recognizing a prospective purchasing opportunity. On the other hand, the sell condition comprises an additional set of logical criteria that are intended to initiate a sell signal by analyzing discernible patterns in the colours of the candlesticks, the reference line, and the divergence line. Analogous to the buy condition, it evaluates the colour patterns of historical candlesticks (previously known as barColor) and the divergence line and reference line (previously referred to as oscLineColor). It assesses the final three intervals (numbered from zero to two) of the divergence line and requires a uniform LIME colour throughout each of these intervals. Additionally, the condition mandates that the preceding candlestick (index 1) be coloured LIME and the present candlestick (index 0) be coloured WHITE.

Figure 7.9: Buy Conditions

Figure 7.10: Sell Condition

The algorithmic logic that directs the bot's decision-making process to initiate buy or sell actions in accordance with its predefined trading strategy is comprised of these conditions. The major and most important part is to test the bot strategy. This is done through the concept called backtest. For the backtest, we use the Trading View platform. The accuracy and results are shown below.

Figure 7.11: BackTest

How it works

Trading View functions as the central command hub for our algorithmic trading program, realizing the complexities of our trading strategy. While our algorithm continuously monitors the markets, particular circumstances serve as impetuses for strategic determinations. The purchase or sell criteria serve as the critical factors that determine whether to initiate our trading strategy. Upon discerning a favourable occasion through its computations and predetermined criteria, the algorithm initiates a notification within the Trading View interface. The alerts are crucial in guiding the trader's interaction with the market. As soon as the algorithm identifies favourable conditions that could lead to a buy signal, an alert is generated in a timely manner to notify traders of possible entry points for long positions. In contrast, another alert is generated when the algorithm detects conditions that are suggestive of a sell signal. This serves to direct traders towards the most advantageous exit points or opportunities for short selling. The true worth of these notifications resides in their promptness and pertinence. These alerts function as immediate notifications, slicing through the complexity of the market and providing the trader with practical insights. Trading View notifications serve as a conduit connecting the automated analyses performed by the algorithm with the strategic decision-making of the trader.

Figure 7.12: Order Triggered

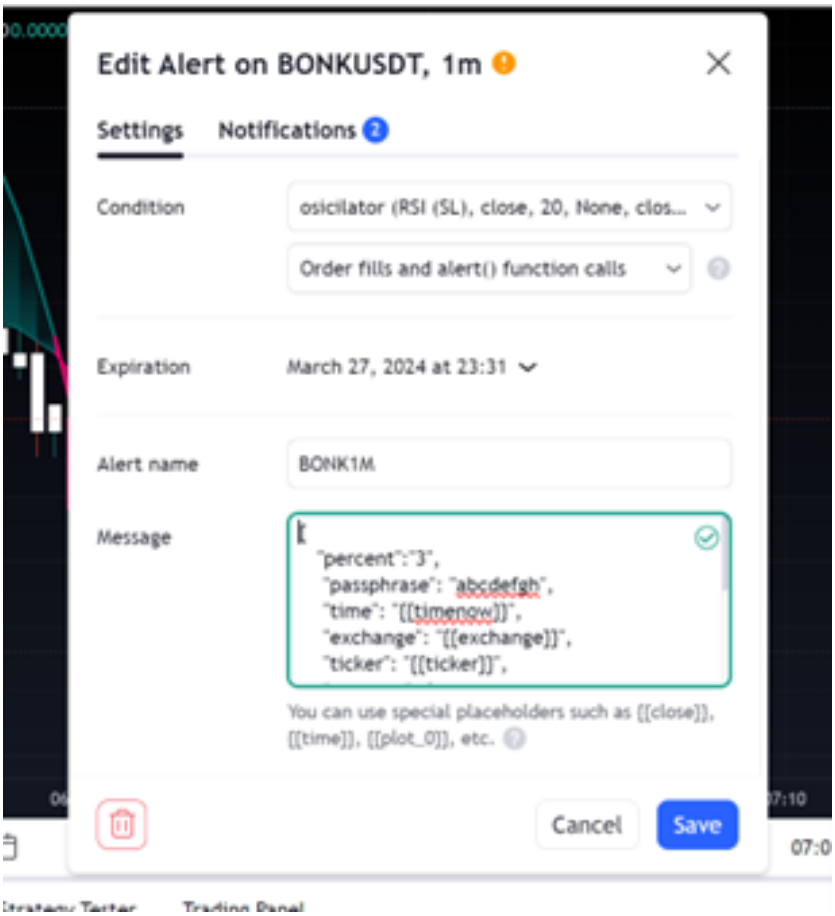


Figure 7.13: JSON Payload Data

The JSON sent from the front end has much information about the order, for example, order type, order price, and much other information for the fulfilment of the orders.

```
    {
   "per cent":"3",
    "passphrase": "abcdefgh",
    "time": "{{timenow}}",
    "Exchange": "{{exchange}}",
    "ticker": "{{ticker}}",
    "strategy": {
        "order_action": "{{strategy.order.action}}",
        "order_price": "{{strategy.order.price}}"
    }
}
```

This uninterrupted flow of information empowers traders to remain well-informed, exploit market prospects, and manoeuvre through the ever-changing environment with assurance and accuracy. Central to the oper-

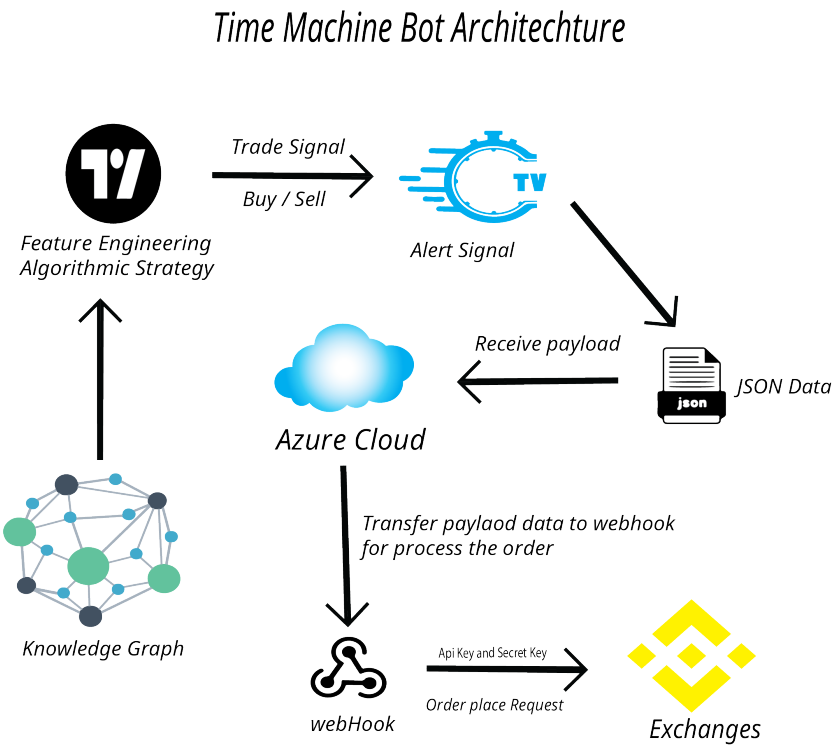


Figure 7.14: Architecture of Time Machine Bot

ational architecture of our algorithm is the smooth incorporation of Trading

View alerts into the ever-evolving functionalities of the Binance exchange. The seamless integration of data and execution is enabled by the secret key, Binance's API, and Flask, all of which are integral components in guaranteeing a dependable and protected trading environment. Flask, a nimble yet robust web framework, serves as the pivotal element, enabling the development of a responsive API. By means of this API, Trading View's signals are transmitted to Binance and subsequently converted into executable commands. Upon the activation of a trading signal on Trading View, a JSON data payload that has been carefully organized and structured is promptly transmitted to our Flask API. The payload functions as an all-encompassing transmission containing critical information, including the nature of the order (buy or sell), the designated execution price, and the trading pair in question. This is the request received from the alert of the trading View. Whenever the order conditions are met, a request with data is sent to the bot, and the bot fulfils the order using a library called Python Binance.

Developed with optimal performance in mind, the Flask API efficiently processes the incoming JSON data, extracting crucial information required to interpret the intricacies of the trading signal. The order type serves as the foundational criterion that dictates the subsequent course of action on Binance. The Flask API manages the execution of various trading strategies, including market buy and sell transactions, limit orders at specific prices, and the implementation of complex trading strategies. It effectively converts the signal into Binance orders that are both coherent and executable.

```
    WEBHOOK_PASSPHRASE = "abcdefgh"
API_KEY = 'j9T5c7dSTQVqeCiHrEkIxhU20RW1U9aLpag803pKxcuofrIn
PCNIKzbtA993Yb19'
API_SECRET = 'R5uPpGHKQocfcW4xu34jLHy08H6FrxeBipskSvH2K8ntRL
G6QrmIsKHa17af0GWN'
```

Nonetheless, security and integrity are of the utmost importance during this procedure; Binance 's API and secret key serve this purpose. By acting as the authenticated gateway, the API key provides secure access to Binance's trading functionalities. Concurrently, the secret key guarantees the preservation of communication integrity and confidentiality between our Flask API and the Binance exchange. By preventing unauthorized access, this comprehensive security mechanism fortifies the entirety of the transactional process against potential financial transaction risks. Fundamentally, the collaboration among Binance's API, Flask, and the secret key not only enables the smooth implementation of trading strategies but also guarantees a reliable and secure infrastructure for cryptocurrency trading. This integration provides traders with an advanced set of tools, utilizing technology to effectively and precisely navigate the constantly changing cryptocurrency markets.

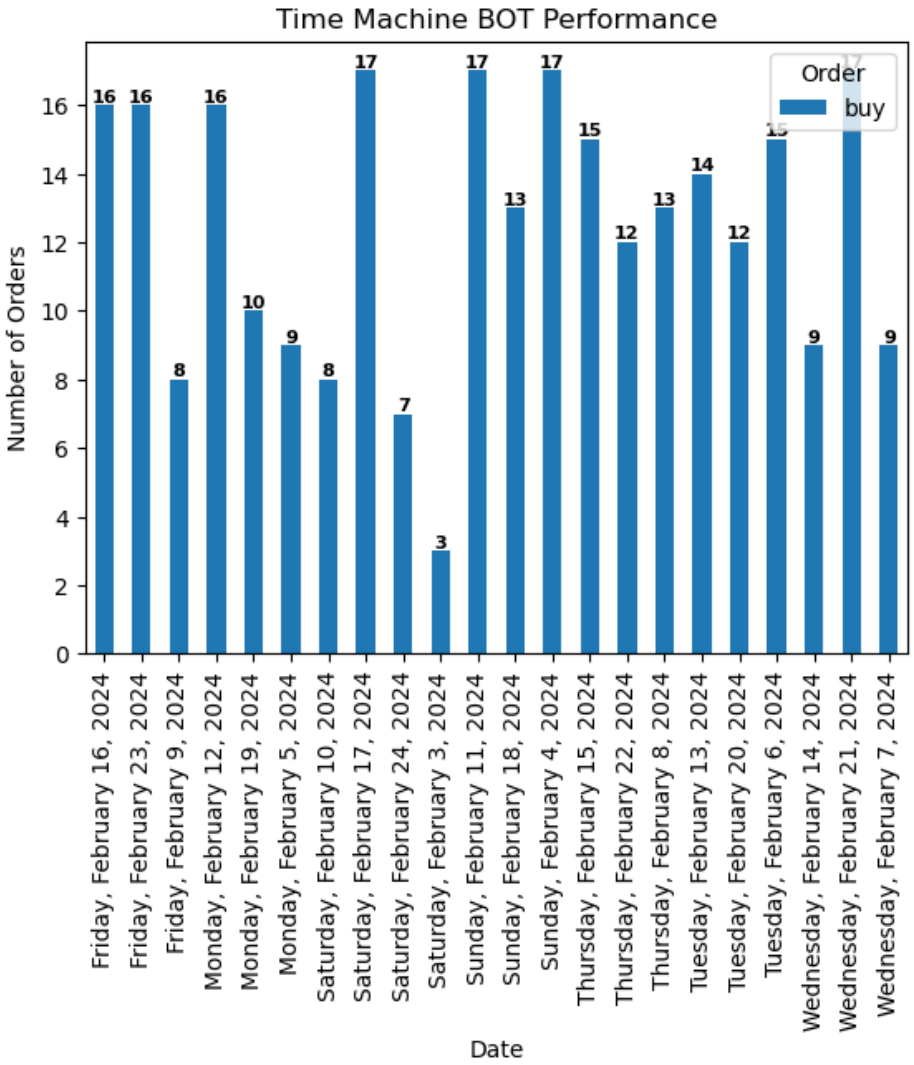


Figure 7.15: Time Machine BOT Results

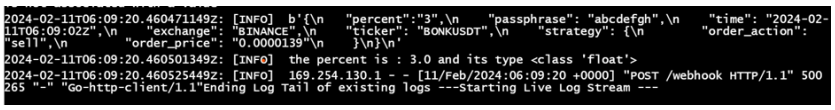


Figure 7.16: Payload Request

```
import json
from flask import Flask, request, jsonify, render_template
from binance import Client
from binance.enums import *
from binance.helpers import round_step_size
import math
import time
import config
app = Flask(__name__)
def Buy(ticker,order_price):
    print(ticker)
    balance= client.get_asset_balance(asset="USDT")
    open_orders = client.get_open_orders(symbol=ticker)
    sell_limit_orders = [order for order in open_orders if order['side
    sell_limit_orders_count = len(sell_limit_orders)
    print(f"sell limit : {sell_limit_orders_count}")

    # ////////////////////////////////////////////////

    quantity= 0
    order_size= 0

    if sell_limit_orders_count == 0:
        order_size = float(balance['free']) / 3.0
        quantity=int(float(order_size/float(order_price)))
    elif sell_limit_orders_count == 3:
        print("sell order 3 already placed, paisay kaam hain")
    elif sell_limit_orders_count == 2:
        order_size = float(balance['free']) / 1.0
        quantity=int(float(order_size/float(order_price)))
    elif sell_limit_orders_count == 1:
        order_size = float(balance['free']) / 2.0
        quantity=int(float(order_size/float(order_price)))
    print(f"Quantity is : {quantity}")
    try:
        order = client.order_market_buy(symbol=ticker,
        quantity=quantity)
        print(quantity)
    except Exception as e:
        print("an exception occurred - {}".format(e))
        return False
    return order
```

```
def Sell(ticker,order_price,percent):

    tickerData= ticker.replace('USDT','')
    print(f" => {tickerData} having type {type(tickerData)}")
    balance= client.get_asset_balance(asset=tickerData)
    print(f' => {balance} having type {type(balance)}')
    totalbalance=int(float(balance['free']))
    print(f" => {totalbalance} having type {type(totalbalance)}")
    # ////////////////// Calculate the limit price////////////
    print(f"the percent : {percent}")
    percentage=float((float(order_price)/100) * percent)
    Final_Price=float(order_price) + percentage
    rounded_price = format(round(Final_Price,8),".8f")
    print(rounded_price)
    try:
        print(ticker)
        order = client.order_limit_sell(symbol=ticker,
        quantity=totalbalance,price=rounded_price)
        print(f" => {totalbalance} having
        type {type(total balance)}")
    except Exception as e:
        print("an exception occurred - {}".format(e))
        return False
    return order
client = Client(config.API_KEY, config.API_SECRET, tld='com')

@app.route('/')
def welcome():
    return render_template('index.html')

@app.route('/webhook', methods=['POST'])
def webhook():
    print(request.data)
    data = json.loads(request.data)
    if data['passphrase'] != config.WEBHOOK_PASSPHRASE:
        return {
            "code": "Error",
            "message": "Nice try, invalid passphrase"
        }
    side = data['strategy']['order_action'].upper()
    ticker=data['ticker']
    percent=float(data["percent"])
```

```
    print(f"the percent is : {percent} and its type {type(percent)}")
    order_price=data['strategy']['order_price']
    order_res=False
    if side == "BUY":
        order_response=Buy(ticker,order_price)
        if(order_response):
            order_res=Sell(ticker,order_price,percent)

    if order_response and order_res:
        message={
            "code": "success",
            "message": "order executed"
        }

    else:
        print("order failed")
        message={
            "code": "Error",
            "message": "order failed"
        }
    return message
if __name__ == '__main__':
    app.run(debug=True)
```

The critical Flask API, which is an integral part of our trading infrastructure, is hosted virtually on the dependable and scalable Azure Web App Service. By deploying on the Azure cloud platform, this initiative demonstrates a deliberate effort to achieve dependability, expandability, and uninterrupted access. The Flask API, which has been carefully designed to process and respond to Trading View alerts, is prepared to receive trader signals. The pivotal factor is the URL of the Flask API that is deployed and strategically inserted into Trading View notifications. Upon the activation of the alert on Trading View, immediate communication is established with the Flask API through its Azure Web App URL. This initiates a sequence of occurrences that ultimately result in prompt and accurate trading operations on the Binance exchange. By selecting Azure for deployment, we not only guarantee the trading infrastructure's high availability but also emphasize our dedication to utilizing state-of-the-art technologies to deliver an optimized and resilient trading experience.

PART II

CONCLUSION

Chapter 8

Concluding Remarks

In conclusion, this thesis has provided a comprehensive exploration of the transformative impact of algorithmic trading bots in the context of cryptocurrency markets, underscoring their role in reshaping modern trading practices. From the elucidation of fundamental concepts in knowledge graphs to the implementation of advanced machine learning algorithms, each chapter has contributed to a deeper understanding of the integration of knowledge-driven approaches and automated trading solutions.

The advent of algorithmic trading bots has ushered in a new era of efficiency and objectivity in financial trading. By leveraging machine learning concepts, these bots offer precision and reliability, operating tirelessly to capitalize on market opportunities without the limitations of human fatigue or emotion. The comparison drawn between human traders and algorithmic bots highlights the significant disparity in trading capacity, emphasizing the unparalleled advantage of automated trading solutions in today's fast-paced markets.

The Real-Time AI-Trading Signal Machine represents a novel innovation, offering enhanced predictive capabilities and insights derived from automatic machine learning. Through rigorous analysis of real-time data and hyperparameter optimization, this system exemplifies the intersection of technological innovation and financial efficiency, providing traders with a competitive edge in cryptocurrency markets.

Likewise, the Time Machine Bot further exemplifies the transformative potential of algorithmic trading, facilitating automated trading decisions based on indicators derived from feature engineering and machine learning insights. Operating continuously with precision and objectivity, Traders are able to take advantage of opportunities in the markets while minimizing the drawbacks of manual trading with the assistance of the program.

Moreover, in selecting the LightGBM algorithm for predicting BTC across different time intervals, ranging from highly volatile 1-minute datasets to less

volatile 15-minute datasets, this thesis acknowledges the challenges posed by market dynamics. Despite the difficulty in predicting highly volatile datasets, the utilization of LightGBM underscores its efficacy in capturing and processing complex market behaviours.

Furthermore, the integration of alert generation and order placement functionality within the bot enhances its capabilities. When the bot's strategy generates a signal to buy or sell, an alert is generated, triggering a request to the server with payload information detailing the order. This data undergoes verification on the server before initiating a request to Binance using API and secret keys to execute the order. Notably, the bot is designed to trade on any digital financial market, providing traders with a versatile and efficient trading solution.

Overall, this thesis underscores the significance of integrating knowledge-driven approaches with advanced machine-learning techniques to enhance trading efficiency and automation in cryptocurrency markets.The technological landscape is undergoing constant evolution, the adoption of algorithmic trading bots represents a paradigm shift in financial trading, offering unparalleled opportunities for traders to navigate the complexities of modern markets with confidence and efficiency.

Chapter 9

Future work

As further advancements occur, the AI Real-Time Signal Machine will be expanded to generate forecasts over more extended time periods, potentially weekly, daily, and even hourly. The implementation of this expansion will necessitate further data preprocessing and model training to capture trends and patterns with greater precision over extended time periods. Furthermore, the expansion of the system's crypto pair support will necessitate the implementation of resilient data management strategies to efficiently process the augmented data volume.
In an effort to optimize trading achievement, the time machine automaton will incorporate sophisticated trading strategies. Investing at extreme lows and selling at extreme highs, in addition to utilizing wick trading to profit from short-term price fluctuations, will be among these strategies. Ongoing efforts will be made to optimize and enhance the performance of the AI models and trading strategies. This will entail refining hyper parameters, investigating novel machine learning algorithms, and integrating state-of-the-art technologies to increase profitability and efficiency. To augment stability and profitability, the organization will employ robust risk management strategies and portfolio diversification techniques. This entails the implementation of stop-loss mechanisms, the adoption of position-sizing strategies, and the diversification of investments across various asset classes. Adding outside factors like news sentiment analysis, macroeconomic indicators, and market sentiment analysis to the trading system will improve its ability to predict the future and give it more information. The trading strategies will undergo comprehensive back testing and substantiation in order to ascertain their efficacy and resilience in the face of diverse market conditions. This process entails the simulation of transactions using historical data and the assessment of the strategies' performance metrics. Ensuring adherence to regulatory obligations and guidelines that govern cryptocurrency trading will be a top priority, including keeping abreast of the dynamic regulatory environ-

ment in order to adjust trading strategies accordingly.
An effort will be made to improve the intuitiveness and usability of the trading system's user interface. This may entail the creation of mobile applications or web interfaces to facilitate user access to the system. In pursuit of joint ventures with fellow researchers and industry professionals, this study aims to investigate novel research and development pathways in the domain of AI-powered trading platforms for cryptocurrency markets, exchange concepts, and impart insights. The research endeavors to enhance the functionalities and efficacy of AI-powered trading systems within cryptocurrency markets by means of these forthcoming advancements. This will make a valuable contribution to the continuous improvement of automated trading strategies within the domain of digital assets.

Bibliography

[1] Alberto Elduque and Alicia Labra. Evolution algebras, automorphisms, and graphs. *Linear and Multilinear Algebra*, 69:331 – 342, 2019.

[2] Elisabete Barreiro, Antonio Jesús Calderón, Samuel A. Lopes, and José M. Sánchez. Leibniz algebras and graphs. *Linear and Multilinear Algebra*, 71:1994 – 2007, 2022.

[3] Izzat Qaralleh and Farrukh Mukhamedov. Volterra evolution algebras and their graphs. *Linear and Multilinear Algebra*, 69:2228 – 2244, 2019.

[4] Chen Zheng, Yushu An, Zhanxi Wang, Xiansheng Qin, Benoît Eynard, Matthieu Bricogne, Julien Le Duigou, and Yicha Zhang. Knowledge-based engineering approach for defining robotic manufacturing system architectures. *International Journal of Production Research*, 61(5):1436–1454, 2023.

[5] Jorge Morales and Andrés Melgar. Research on proposals and trends in the architectures of semantic search engines: A systematic literature review. In *2017 federated conference on computer science and information systems (FedCSIS)*, pages 271–280. IEEE, 2017.

[6] Yanlei Diao, Paweł Guzewicz, Ioana Manolescu, and Mirjana Mazuran. Efficient exploration of interesting aggregates in rdf graphs. In *Proceedings of the 2021 International Conference on Management of Data*, pages 392–404, 2021.

[7] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[8] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature biotechnology*, 36(5):411–420, 2018.

[9] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. Structgpt: A general framework for large language model to reason over structured data. *arXiv preprint arXiv:2305.09645*, 2023.

[10] Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

[11] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[12] Brian C O'Neill, Elmar Kriegler, Kristie L Ebi, Eric Kemp-Benedict, Keywan Riahi, Dale S Rothman, Bas J Van Ruijven, Detlef P Van Vuuren, Joern Birkmann, Kasper Kok, et al. The roads ahead: Narratives for shared socioeconomic pathways describing world futures in the 21st century. *Global environmental change*, 42:169–180, 2017.

[13] Akeem Pedro, Anh-Tuan Pham-Hang, Phong Thanh Nguyen, and Hai Chien Pham. Data-driven construction safety information sharing system based on linked data, ontologies, and knowledge graph technologies. *International journal of environmental research and public health*, 19(2):794, 2022.

[14] Lingyu Li, Xianrong Zheng, and Shuxi Wang. The effect of sustainability information disclosure on the cost of equity capital: An empirical analysis based on gartner top 50 supply chain rankings. *Journal of Risk and Financial Management*, 16(8):358, 2023.

[15] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[16] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A taxonomy and survey of dynamic graph visualization. In *Computer graphics forum*, volume 36, pages 133–159. Wiley Online Library, 2017.

[17] Jéssica Monteiro, Filipe Sá, and Jorge Bernardino. Experimental evaluation of graph databases: Janusgraph, nebula graph, neo4j, and tigergraph. *Applied Sciences*, 13(9):5770, 2023.

[18] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. In *International conference on machine learning*, pages 1115–1124. PMLR, 2018.

[19] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[20] Julia Sasse, Johannes Darms, and Juliane Fluck. Semantic metadata annotation services in the biomedical domain—a literature review. *Applied Sciences*, 12(2):796, 2022.

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[22] Yuan Guo, Jingyong Zhou, Qiang Qin, Yun Wei, and Weitang Zhang. An improved algorithm and implementation of data mining for intelligent manufacturing association rules based on pattern recognition. *IEEE Consumer Electronics Magazine*, 12(2):94–99, 2022.

[23] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[24] Ritu Parasher. Load flow analysis of radial distribution network using linear data structure. *arXiv preprint arXiv:1403.4702*, 2014.

[25] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017.

[26] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv preprint arXiv:2302.06466*, 2023.

[27] William Spoth, Ting Xie, Oliver Kennedy, Ying Yang, Beda Hammerschmidt, Zhen Hua Liu, and Dieter Gawlick. Schemadrill: Interactive semi-structured schema design. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–7, 2018.

[28] Zhengchun Lu, Mayu Morita, Tyler S Yeager, Yunpeng Lyu, Sophia Y Wang, Zhigang Wang, and Guang Fan. Validation of artificial intelligence (ai)-assisted flow cytometry analysis for immunological disorders. *Diagnostics*, 14(4):420, 2024.

[29] Shaoxiong Ji, Shirui Pan, E. Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33:494–514, 2020.

[30] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *ArXiv*, abs/2306.08302, 2023.

[31] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *ArXiv*, abs/2302.06466, 2023.

[32] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *North American Chapter of the Association for Computational Linguistics*, 2016.

[33] Besnik Fetahu, Sudipta Kar, Zhiyu Chen, Oleg Rokhlenko, and Shervin Malmasi. Semeval-2023 task 2: Fine-grained multilingual named entity recognition (multiconer 2). *ArXiv*, abs/2305.06586, 2023.

[34] Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. Named entity recognition and classification in historical documents: A survey. *ACM Computing Surveys*, 56:1 – 47, 2023.

[35] Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. Zero-shot temporal relation extraction with chatgpt. *arXiv preprint arXiv:2304.05454*, 2023.

[36] Somin Wadhwa, Silvio Amir, and Byron C Wallace. Revisiting relation extraction in the era of large language models. *arXiv preprint arXiv:2305.05003*, 2023.

[37] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[38] Ernest Davis. Benchmarks for automated commonsense reasoning: A survey. *arXiv preprint arXiv:2302.04752*, 2023.

[39] Marwin HS Segler, Mike Preuss, and Mark P Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018.

[40] Leonardo FR Ribeiro, Mengwen Liu, Iryna Gurevych, Markus Dreyer, and Mohit Bansal. Factgraph: Evaluating factuality in summarization with semantic graph representations. *arXiv preprint arXiv:2204.06508*, 2022.

[41] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[42] Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4498–4507, 2020.

[43] S Neelakandan, A Arun, R Ram Bhukya, Bhalchandra M Hardas, T Ch Anil Kumar, and M Ashok. An automated word embedding with parameter tuned model for web crawling. *Intelligent Automation & Soft Computing*, 32(3):1617–1632, 2022.

[44] Marc-André Zöller and Marco F Huber. Benchmark and survey of automated machine learning frameworks. *Journal of artificial intelligence research*, 70:409–472, 2021.

[45] Pieter Gijsbers, Joaquin Vanschoren, and Randal S. Olson. Layered tpot: Speeding up tree-based pipeline optimization. 1 2018.

[46] Jeff Heaton. An empirical analysis of feature engineering for predictive modeling. In *SoutheastCon 2016*, pages 1–6. IEEE, 2016.

[47] Barbara Kitchenham, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.

[48] Huong Thanh Le and Luan Van Tran. Automatic feature selection for named entity recognition using genetic algorithm. pages 81–87, 2013.

[49] William La Cava and Jason H. Moore. Learning feature spaces for regression with genetic programming. *Genetic Programming and Evolvable Machines*, 2020.

[50] Muhammad Wafi, Umar Faruq, and Ahmad Afif Supianto. Automatic feature selection for modified k- nearest neighbor to predict student's academic performance. In *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, pages 44–48, 2019.

[51] Dustin Y. Harvey and Michael D. Todd. Automated feature design for numeric sequence classification by genetic programming. *IEEE Transactions on Evolutionary Computation*, 19:474–489, 8 2015.

[52] Randal S. Olson, Nathan Bartley, Ryan J. Urbanowicz, and Jason H. Moore. Evaluation of a tree-based pipeline optimization tool for automating data science. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, GECCO '16, page 485–492, New York, NY, USA, 2016. Association for Computing Machinery.

[53] Aiguo Wang, Ning An, Guilin Chen, Lian Li, and Gil Alterovitz. Accelerating wrapper-based feature selection with k-nearest-neighbor. *Knowledge-Based Systems*, 83:81–91, 2015.

[54] Gongde Guo, Daniel Neagu, and Mark T. D. Cronin. Using $k$nn model for automatic feature selection. In Peng Wang, Maneesha Singh, Chidanand Apté, and Petra Perner, editors, *Pattern Recognition and Data Mining, Third International Conference on Advances in Pattern Recognition, ICAPR 2005, Bath, UK, August 22-25, 2005, Proceedings, Part I*, volume 3686 of *Lecture Notes in Computer Science*, pages 410–419. Springer, 2005.

[55] Y. Huang, P.J. McCullagh, and N.D. Black. Feature selection via supervised model construction. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 411–414, 2004.

[56] Yue Huang, Paul McCullagh, Norman Black, and Roy Harper. Feature selection and classification model construction on type 2 diabetic patients' data. *Artificial Intelligence in Medicine*, 41:251–262, 11 2007.

[57] Shreyal Gajare and Shilpa Sonawani. Improved automatic feature selection approach for health risk prediction. In *2018 Second International Conference on Computing Methodologies and Communication (ICCMC)*, pages 816–819, 2018.

[58] Yvan Lucas, Pierre Edouard Portier, Léa Laporte, Liyun He-Guelton, Olivier Caelen, Michael Granitzer, and Sylvie Calabretto. Towards automated feature engineering for credit card fraud detection using multi-perspective hmms. *Future Generation Computer Systems*, 102:393–402, 1 2020.

[59] Xuewen Chen, Association for Computing Machinery. Special Interest Group on Information Retrieval, Hypermedia, Web Association for Computing Machinery. Special Interest Group on Hypertext, and ACM Digital Library. *Automated Feature Weighting in Naive Bayes forHigh-dimensional Data Classification.*

[60] Haidar Osman, Mohammad Ghafari, and Oscar Nierstrasz. Automatic feature selection by regularization to improve bug prediction accuracy. pages 27–32. Institute of Electrical and Electronics Engineers Inc., 3 2017.

[61] Sergio Bermejo. Ensembles of wrappers for automated feature selection in fish age classification. *Computers and Electronics in Agriculture*, 134:27–32, 3 2017.

[62] Ofer Dor and Yoram Reich. Strengthening learning algorithms by feature discovery. *Information Sciences*, 189:176–190, 4 2012.

[63] Jiayi Duan, Ziheng Zeng, Alina Oprea, and Shobha Vasudevan. Automated generation and selection of interpretable features for enterprise security. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1258–1265, 2018.

[64] Matthias Reif and Faisal Shafait. Efficient feature size reduction via predictive forward selection. *Pattern Recognition*, 47:1664–1673, 4 2014.

[65] Ambika Kaul, Saket Maheshwary, and Vikram Pudi. Autolearn - automated feature generation and selection. volume 2017-November, pages 217–226. Institute of Electrical and Electronics Engineers Inc., 12 2017.

[66] Gaurav Dhiman, Diego Oliva, Amandeep Kaur, Krishna Kant Singh, S Vimal, Ashutosh Sharma, and Korhan Cengiz. Bepo: A novel binary emperor penguin optimizer for automatic feature selection. *Knowledge-Based Systems*, 211:106560, 2021.

[67] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B Khalil, and Deepak Turaga. Learning feature engineering for classification, 2017.

[68] Hoang Thanh Lam, Tran Ngoc Minh, Mathieu Sinn, Beat Buesser, and Martin Wistuba. Neural feature learning from relational database. 1 2018.

[69] Gilad Katz, Eui Chul, Richard Shin, and Dawn Song. Explorekit: Automatic feature generation and selection.

[70] Udayan Khurana, Horst Samulowitz, and Deepak Turaga. Feature engineering for predictive modeling using reinforcement learning. pages 3407–3414. AAAI press, 2018.

[71] Xinbo Ban, Chao Chen, Shigang Liu, Yu Wang, and Jun Zhang. Deep-learnt features for twitter spam detection.

[72] Kai Hu, Joey Wang, Yong Liu, and Datong Chen. Automatic feature engineering from very high dimensional event logs using deep neural networks. Association for Computing Machinery, 8 2019.

[73] Gavin Brown, Xin Yao, Jeremy Wyatt, Heiko Wersing, Bernhard Sendhoff, and Honda R&d. Exploiting ensemble diversity for automatic feature extraction.

[74] Horst Samulowitz Elias Khalil and Deepak Turaga Udayan Khurana, Fatemeh Nargesian. Automating feature engineering. 2016.

[75] K. Ghosh Dastidar, J. Jurgovsky, W. Siblini, L. He-Guelton, and M. Granitzer. Nag: Neural feature aggregation framework for credit card fraud detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 92–101, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society.

[76] W. Fan, K. Liu, H. Liu, P. Wang, Y. Ge, and Y. Fu. Autofs: Automated feature selection via diversity-aware interactive reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1008–1013, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society.

[77] Mengying Xu and Xing Li. Bgp anomaly detection based on automatic feature extraction by neural network. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 46–50. IEEE, 2020.

[78] Masafumi Oyamada. Extracting feature engineering knowledge from data science notebooks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 6172–6173, 2019.

[79] Kerman López de Calle, Susana Ferreiro, Aitor Arnaiz, and Basilio Sierra. Comparison of automated feature selection and reduction methods on the condition monitoring issue. volume 16, pages 2–9. Elsevier B.V., 2018.

[80] Valentin Kassarnig and Franz Wotawa. An approach to automatically extract predictive properties from nominal attributes in relational

databases. In Yang Song, Bing Liu, Kisung Lee, Naoki Abe, Calton Pu, Mu Qiao, Nesreen Ahmed, Donald Kossmann, Jeffrey Saltz, Jiliang Tang, Jingrui He, Huan Liu, and Xiaohua Hu, editors, *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018, pages 4932–4939, United States, January 2019. Institute of Electrical and Electronics Engineers. 2018 IEEE International Conference on Big Data, Big Data 2018 ; Conference date: 10-12-2018 Through 13-12-2018.

[81] Hoang Thanh Lam, Johann-Michael Thiebaut, Mathieu Sinn, Bei Chen, Tiep Mai, and Oznur Alkan. One button machine for automating feature engineering in relational databases. 6 2017.

[82] Afifa Fatima, Faizan Ali Khan, Aneeq Raza, and Abdul Basit Kamran. Automated feature synthesis from relational database for data science related problems. pages 71–75. Institute of Electrical and Electronics Engineers Inc., 1 2019.

[83] Eric Gaussier, IEEE Computational Intelligence Society, Association for Computing Machinery. Special Interest Group on Knowledge Discovery & Data Mining, Institute of Electrical, Electronics Engineers, and Association for Computing Machinery. *Deep Feature Synthesis:Towards Automating Data Science Endeavors*.

[84] Sainyam Galhotra, Udayan Khurana, Oktie Hassanzadeh, Kavitha Srinivas, Horst Samulowitz, and Miao Qi. Automated feature enhancement for predictive modeling using external knowledge.

[85] Weiwei Cheng, Gjergji Kasneci, Thore Graepel, David Stern, and Ralf Herbrich. Automated feature generation from structured knowledge. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1395–1404, 2011.

[86] Abhishek Bafna and Jenna Wiens. Automated feature learning: Mining unstructured data for useful abstractions. volume 2016-January, pages 703–708. Institute of Electrical and Electronics Engineers Inc., 1 2016.

[87] Martin Atzmueller and Eric Sternberg. Mixed-initiative feature engineering using knowledge graphs. Association for Computing Machinery, Inc, 12 2017.

[88] Michael Anderson, Dolan Antenucci, Victor Bittorf, Matthew Burgess, Michael Cafarella, Arun Kumar, Feng Niu, Yongjoo Park, Christopher Ré, and Ce Zhang. Brainwash: A data system for feature engineering.

[89] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. *Auto-sklearn: Efficient and Robust Automated Machine Learning*. 2019.

[90] Vikas C Raykar, Balaji Krishnapuram, Jinbo Bi, Murat Dundar, and R Bharat Rao. Bayesian multiple instance learning: Automatic feature selection and inductive transfer.

[91] Hirotaka Hachiya and Masashi Sugiyama. Feature selection for reinforcement learning: Evaluating implicit state-reward dependency via conditional mutual information.

[92] Andreas Gocht, Christoph Lehmann, and Robert Schöne. A new approach for automated feature selection. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4915–4920, 2018.

[93] R. An, X. Shi, and B. Xu. Fast automatic feature selection for multi-period sliding window aggregate in time series. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 906–911, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society.

[94] Rui An, Xingtian Shi, and Baohan Xu. Fast automatic feature selection for multi-period sliding window aggregate in time series. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 906–911. IEEE, 2020.

[95] Vitor Cerqueira, Nuno Moniz, and Carlos Soares. Vest: Automatic feature engineering for forecasting. *Machine Learning*, pages 1–23, 2021.

[96] Tizian Schneider, Nikolai Helwig, and Andreas Schütze. Automatic feature extraction and selection for classification of cyclical time series data. *Technisches Messen*, 84:198–206, 3 2017.

[97] Micah J. Smith, Roy Wedge, and Kalyan Veeramachaneni. Featurehub: Towards collaborative data science. volume 2018-January, pages 590–600. Institute of Electrical and Electronics Engineers Inc., 7 2017.

[98] Franziska Horn, Robert Pack, and Michael Rieger. The autofeat python library for automated feature engineering and selection. 1 2019.

[99] Jianyu Zhang, Françoise Fogelman-Soulié, and Christine Largeron. Towards automatic complex feature engineering. volume 11234 LNCS, pages 312–322. Springer Verlag, 2018.

[100] Vinícius Veloso de Melo and Wolfgang Banzhaf. Automatic feature engineering for regression models with machine learning: An evolutionary

computation and statistics hybrid. *Information Sciences*, 430-431:287–313, 3 2018.

[101] IEEE Staff and IEEE Staff. *Implementation of automatic feature selection methods for BCI realization.*

[102] IEEE Computer Society. and IEEE Computer Society. Technical Council on Software Engineering. *Predicting Performance via Automated Feature-Interaction Detection.* IEEE, 2012.

[103] Qitao Shi, Ya Lin Zhang, Longfei Li, Xinxing Yang, Meng Li, and Jun Zhou. Safe: Scalable automatic feature engineering framework for industrial tasks. volume 2020-April, pages 1645–1656. IEEE Computer Society, 4 2020.

[104] *Automatic Generation of Feature Models from UML Requirement Models.*

[105] Bin Liu, Niannan Xue, Huifeng Guo, Ruiming Tang, Stefanos Zafeiriou, Xiuqiang He, and Zhenguo Li. Autogroup: Automatic feature grouping for modelling explicit high-order feature interactions in ctr prediction. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 199–208, 2020.

[106] Mihir Gada, Zenil Haria, Arnav Mankad, Kaustubh Damania, and Smita Sankhe. Automated feature engineering and hyperparameter optimization for machine learning. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 981–986. IEEE, 2021.

[107] Wei Fan, Kunpeng Liu, Hao Liu, Pengyang Wang, Yong Ge, and Yanjie Fu. Autofs: Automated feature selection via diversity-aware interactive reinforcement learning. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1008–1013. IEEE, 2020.

[108] Mengnan Song, Jiasong Wang, Tongtong Zhang, Guoguang Zhang, Ruijun Zhang, and Suisui Su. Effective automated feature derivation via reinforcement learning for microcredit default prediction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[109] Isaac Appiah-Otoo. The impact of the russia-ukraine war on the cryptocurrency market. *Asian Economics Letters*, 2023.

[110] P. Hägg. *A Study on Algorithmic Trading*. PhD thesis, Abgerufen von `https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-328676`, 2023.

[111] Akanksha Jalan and Roman Matkovskyy. Systemic risks in the cryptocurrency market: Evidence from the ftx collapse. *Finance Research Letters*, 2023.

[112] Stanisław Drożdż, Jarosław Kwapień, and Marcin Wątorek. What is mature and what is still emerging in the cryptocurrency market? *Entropy*, 25, 2023.

[113] Donna L Hoffman, C Page Moreau, Stefan Stremersch, and Michel Wedel. The rise of new technologies in marketing: A framework and outlook, 2022.

[114] Pál András Papp and Roger Wattenhofer. Network-aware strategies in financial systems. *arXiv preprint arXiv:2002.07566*, 2020.

[115] Stanisław Drożdż, Jarosław Kwapień, and Marcin Wątorek. What is mature and what is still emerging in the cryptocurrency market? *Entropy*, 25(5):772, 2023.

[116] Zaghum Umar, Onur Polat, Sun-Yong Choi, and Tamara Teplova. The impact of the russia-ukraine conflict on the connectedness of financial markets. *Finance Research Letters*, 48:102976, 2022.

[117] Akanksha Jalan and Roman Matkovskyy. Systemic risks in the cryptocurrency market: Evidence from the ftx collapse. *Finance Research Letters*, 53:103670, 2023.

[118] Ahmad Al-Hiyari and Mohamed Chakib Kolsi. How do stock market participants value esg performance? evidence from middle eastern and north african countries. *Global Business Review*, page 09721509211001511, 2021.

[119] Victoria Tyshchenko, Svitlana Achkasova, Oleksii Naidenko, Serhii Kanyhin, and Vlada Karpova. Development of recurrent neural networks for price forecasting at cryptocurrency exchanges. 2023.

[120] Untung Rahardja. The economic impact of cryptocurrencies in indonesia. *ADI Journal on Recent Innovation*, 4(2):194–200, 2023.

[121] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th symposium on operating systems principles*, pages 51–68, 2017.

[122] Sina Shahab and Leonhard K Lades. Sludge and transaction costs. *Behavioural Public Policy*, pages 1–22, 2021.

[123] David Florysiak. Utility tokens, markets in crypto assets regulation (micar), and the costs of being public. *Markets in Crypto Assets Regulation (MiCAR), and the Costs of Being Public (December 7, 2022)*, 2022.

[124] Adiva Fiqri Nugraha, Herman Kabetta, I Komang Setia Buana, and R Budiarto Hadiprakoso. Performance and security comparison of json web tokens (jwt) and platform agnostic security tokens (paseto) on restful apis. In *2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs)*, pages 15–22. IEEE, 2023.

[125] Yizhi Wang, Brian M Lucey, Samuel A Vigne, and Larisa Yarovaya. The effects of central bank digital currencies news on financial markets. *Technological Forecasting and Social Change*, 180:121715, 2022.

[126] Kharis Surya Wicaksana, Rizki Fadhilah Ramadhan, et al. The effect of the russia-ukraine crisis on price fluctuations and trade in energy sector in indonesia. *Jurnal Nasional Pengelolaan Energi MigasZoom*, 4(1):6–18, 2022.

[127] Jie Liu, Chonglin Wu, Lin Yuan, and Jia Liu. Opening price manipulation and its value influences. *International Review of Financial Analysis*, 83:102256, 2022.

[128] Rui Xu, Wenjie Wu, Yanpeng Cai, Hang Wan, Jian Li, Qin Zhu, and Shiming Shen. Feature extraction and prediction of water quality based on candlestick theory and deep learning methods. *Water*, 15(5):845, 2023.

[129] Dave Berger. Investor sentiment: A retail trader activity approach. *Review of Accounting and Finance*, 21(2):61–82, 2022.

[130] Chi-Wei He and Yung-Jang Wang. Bitcoin price jumps and investor sentiment indicators. *Applied Economics Letters*, 30(18):2626–2630, 2023.

[131] David Huffman, Collin Raymond, and Julia Shvets. Persistent overconfidence and biased memory: Evidence from managers. *American Economic Review*, 112(10):3141–3175, 2022.

[132] Astrid Bertrand, Rafik Belloum, James R Eagan, and Winston Maxwell. How cognitive biases affect xai-assisted decision-making: A systematic review. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 78–91, 2022.

[133] Enkhbayar Choijil, Christian Espinosa Méndez, Wing-Keung Wong, Joao Paulo Vieito, and Munkh-Ulzii Batmunkh. Research in international business and finance.

[134] Damir Tokic and Dave Jackson. When a correction turns into a bear market: What explains the depth of the stock market drawdown? a discretionary global macro approach. *Journal of Asset Management*, 24(3):184–197, 2023.

[135] Filipe Correia, Ana Maria Madureira, and Jorge Bernardino. Deep neural networks applied to stock market sentiment analysis. *Sensors*, 22(12):4409, 2022.

[136] Zijian Shi and John Cartlidge. State dependent parallel neural hawkes process for limit order book event stream prediction and simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1607–1615, 2022.

[137] Marc Pilkington and Christine Sinapi. Crisis perception in financial media discourse: a concrete application using the minskian/mainstream opposition. *On the Horizon*, 22(4):280–296, 2014.

[138] Neha Arora and Brijesh K Mishra. Influence of bull and bear market phase on financial risk tolerance of urban individual investors in an emerging economy. *Review of Behavioral Finance*, 15(4):570–591, 2023.

[139] Marcel Boumans and Sabina Leonelli. *From dirty data to tidy facts: Clustering practices in plant phenomics and business cycle analysis*. Springer International Publishing, 2020.

[140] Weiwei Zhan and Hao Jing. Does fintech development reduce corporate earnings management? evidence from china. *Sustainability*, 14(24):16647, 2022.

[141] Sungwon Park, Sungwon Han, Donghyun Ahn, Jaeyeon Kim, Jeasurk Yang, Susang Lee, Seunghoon Hong, Jihee Kim, Sangyoon Park, Hyunjoo Yang, et al. Learning economic indicators by aggregating multi-level geospatial information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12053–12061, 2022.

[142] Lusheng Shao, Derui Wang, Xiaole Wu, et al. Competitive forward and spot trading under yield uncertainty. *Foundations and Trends® in Technology, Information and Operations Management*, 16(3–4):214–228, 2023.

[143] Saif Benjaafar, Guangwen Kong, Xiang Li, and Costas Courcoubetis. Peer-to-peer product sharing: Implications for ownership, usage, and social welfare in the sharing economy. *Management Science*, 65(2):477–493, 2019.

[144] Dominique Raynaud. Da vinci's codex atlanticus, fols. 395r and 686r–686v, refers to leonardo pisano volgarizzato, not to giorgio valla. *Historia Mathematica*, 64:1–18, 2023.

[145] Vijayakumar Mathaiyan. Fluid leaves: Effects of fluid flow on leaf shapes and fibonacci series. *International Journal of Fluid Mechanics Research*, 50(5), 2023.

[146] Mohd Fauzi Ramli, Ahmad Kadri Junoh, Mahyun Ab Wahab, and Wan Zuki Azman Wan Muhamad. Fibonacci retracement pattern recognition for forecasting foreign exchange market. *International Journal of Business Intelligence and Data Mining*, 17(2):159–178, 2020.

[147] Prodromos Tsinaslanidis, Francisco Guijarro, and Nikolaos Voukelatos. Automatic identification and evaluation of fibonacci retracements: Empirical evidence from three equity markets. *Expert Systems with Applications*, 187:115893, 2022.

[148] Andrea Coletta, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Learning to simulate realistic limit order book markets from data as a world agent. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 428–436, 2022.

[149] Gracia H Mamesah, Kristianto Harikatan, Lidya CM Hutapea, and Marchanda C Palalangan. The comparison and relationship of ihsg, nikkei, and nasdaq. *The Contrarian: Finance, Accounting, and Business Research*, 1(2):41–47, 2022.

[150] Muhammad Yusra. Pengaruh frekuensi perdagangan, trading volume, nilai kapitalisasi pasar, harga saham, dan trading day terhadap return saham pada perusahaan kosmetik dan keperluan rumah tangga di bursa efek indonesia. *Jurnal akuntansi dan keuangan*, 7(1):65–74, 2019.

[151] Ruiming Zhu, Wei Nong, Shuya Yamazaki, and Kedar Hippalgaonkar. Wycryst: Wyckoff inorganic crystal generator framework. *Available at SSRN 4658842*, 2023.

[152] KS Vishvaksenan, R Kalaiarasan, R Kalidoss, and R Karthipan. Real time experimental study and analysis of elliott wave theory in signal strength prediction. *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, 88:107–119, 2018.

[153] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2019.

[154] Xiangkun He, Haohan Yang, Zhongxu Hu, and Chen Lv. Robust lane change decision making for autonomous vehicles: An observation adversarial reinforcement learning approach. *IEEE Transactions on Intelligent Vehicles*, 8(1):184–193, 2022.

[155] Yaohu Lin, Shancun Liu, Haijun Yang, Harris Wu, and Bingbing Jiang. Improving stock trading decisions based on pattern recognition using machine learning technology. *PloS one*, 16(8):e0255558, 2021.

[156] Hadrien Oliveri and Alain Goriely. Mathematical models of neuronal growth. *Biomechanics and Modeling in Mechanobiology*, 21(1):89–118, 2022.

[157] Jonathan Munemo. Do african resource rents promote rent-seeking at the expense of entrepreneurship? *Small Business Economics*, 58(3):1647–1660, 2022.

[158] Haifeng Jin, François Chollet, Qingquan Song, and Xia Hu. Autokeras: An automl library for deep learning. *Journal of Machine Learning Research*, 24(6):1–6, 2023.

[159] Yuan-Peng Zhang, Xin-Yun Zhang, Yu-Ting Cheng, Bing Li, Xin-Zhi Teng, Jiang Zhang, Saikit Lam, Ta Zhou, Zong-Rui Ma, Jia-Bao Sheng, et al. Artificial intelligence-driven radiomics study in cancer: the role of feature engineering and modeling. *Military Medical Research*, 10(1):22, 2023.

[160] Angus Kenny, Tapabrata Ray, Steffen Limmer, Hemant Kumar Singh, Tobias Rodemann, and Markus Olhofer. Hybridizing tpot with bayesian optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 502–510, 2023.

[161] Phumudzo Lloyd Seabe, Claude Rodrigue Bambe Moutsinga, and Edson Pindza. Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm: a deep learning approach. *Fractal and Fractional*, 7(2):203, 2023.

[162] Anzhong Huang, Rui Xu, Yu Chen, and Meiwen Guo. Research on multi-label user classification of social media based on ml-knn algorithm. *Technological Forecasting and Social Change*, 188:122271, 2023.

[163] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. Linear regression. In *An introduction to statistical learning: With applications in python*, pages 69–134. Springer, 2023.

[164] Tingting Xu, Yunting Zong, Heng Su, Aohua Tian, Jay Gao, Yurui Wang, and Ruiqi Su. Prediction of multi-scale socioeconomic parameters from long-term nighttime lights satellite data using decision tree regression: A case study of chongqing, china. *Land*, 12(1):249, 2023.

[165] Rezaul Karim, Md Khorshed Alam, and Md Rezaul Hossain. Stock market analysis using linear regression and decision tree regression. In *2021 1st International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, pages 1–6. IEEE, 2021.

[166] Mark R Segal. Machine learning benchmarks and random forest regression. 2004.

[167] JH Graw, WT Wood, and BJ Phrampus. Predicting global marine sediment density using the random forest regressor machine learning algorithm. *Journal of Geophysical Research: Solid Earth*, 126(1):e2020JB020135, 2021.

[168] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[169] Todd G Nick and Kathleen M Campbell. Logistic regression. *Topics in biostatistics*, pages 273–301, 2007.

[170] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[171] Scott Menard. *Applied logistic regression analysis*. Number 106. Sage, 2002.

[172] Qi Shi, Mohamed Abdel-Aty, and Jaeyoung Lee. A bayesian ridge regression analysis of congestion's impact on urban expressway safety. *Accident Analysis & Prevention*, 88:124–137, 2016.

[173] Flavia Alves da Silva, Alexandre Pio Viana, Caio Cezar Guedes Correa, Eileen Azevedo Santos, Julie Anne Vieira Salgado de Oliveira, José Daniel Gomes Andrade, Rodrigo Moreira Ribeiro, and Leonardo Siqueira Glória. Bayesian ridge regression shows the best fit for ssr markers in psidium guajava among bayesian models. *Scientific Reports*, 11(1):13639, 2021.

[174] B Shamreen Ahamed et al. Prediction of type-2 diabetes using the lgbm classifier methods and techniques. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(12):223–231, 2021.

[175] Fachrul Rozi Lubis and Eddy Rahman Syahputra. Peramalan deret waktu untuk bisnis: Pendekatan algoritma lgbm regressor. *Data Sciences Indonesia (DSI)*, 1(2):69–75, 2021.

[176] Dimitri P Solomatine and Durga L Shrestha. Adaboost. rt: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 1163–1168. IEEE, 2004.

[177] G Shanmugasundar, M Vanitha, Robert Čep, Vikas Kumar, Kanak Kalita, and M Ramachandran. A comparative study of linear, random forest and adaboost regressions for modeling non-traditional machining. *Processes*, 9(11):2015, 2021.

[178] Shobhit K Patel, Jaymit Surve, Vijay Katkar, Juveriya Parmar, Fahad Ahmed Al-Zahrani, Kawsar Ahmed, and Francis Minhthang Bui. Encoding and tuning of thz metasurface-based refractive index sensor with behavior prediction using xgboost regressor. *IEEE Access*, 10:24797–24814, 2022.

[179] Mohit Arora, Abhishek Sharma, Sapna Katoch, Mehul Malviya, and Shivali Chopra. A state of the art regressor model's comparison for effort estimation of agile software. In *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pages 211–215. IEEE, 2021.

[180] Randal S Olson and Jason H Moore. Tpot: A tree-based pipeline optimization tool for automating machine learning. In *Workshop on automatic machine learning*, pages 66–74. PMLR, 2016.

[181] Ugur Sahin and A Murat Ozbayoglu. Tn-rsi: Trend-normalized rsi indicator for stock trading systems with evolutionary computation. *Procedia Computer Science*, 36:240–245, 2014.

[182] Yulius Hari and Lily Puspa Dewi. *Forecasting system approach for stock trading with relative strength index and moving average indicator*. PhD thesis, Petra Christian University, 2018.

[183] Mark Larson. *12 Simple Technical Indicators: That Really Work*, volume 69. John Wiley & Sons, 2012.

[184] Amit Kelotra and Prateek Pandey. Stock market prediction using optimized deep-convlstm model. *Big Data*, 8(1):5–24, 2020.

[185] Andrea Kolkova. Testing ema indicator for the currency pair eur/usd. 2017.

[186] Masafumi Nakano, Akihiko Takahashi, and Soichiro Takahashi. Generalized exponential moving average (ema) model with particle filtering and anomaly detection. *Expert Systems with Applications*, 73:187–200, 2017.