|  |  |
| --- | --- |
| **2022- 20223** | **MSc Dissertation** |

|  |  |
| --- | --- |
| **Dissertation Title** | **STOCK PRICE FORECASTING**  **(COMPARISON BETWEEN MACHINE LEARNING AND STATISTICAL METHODS)** |

|  |  |
| --- | --- |
| **Author** | **OSAIGBOKAN GILBERT ENOFE** |
| **K-Number** | **K00278371** |

**MSc** **- Data Analytics (Level 9)**

|  |  |
| --- | --- |
| **Award Title** | **MSc in Data Analytics** |
| **Submitted to:** | **Technological University of the Shannon: Midlands.** |

Text

Description automatically generated

## Declaration of Ethical Practice

Definition: (The Technological University of the Shannon: Midlands)

“the deliberate or reckless representation of another’s words, thoughts, or ideas as one’s own without attribution in connection with submission of academic work, whether graded or otherwise”

As an aspiring professional I am aware of my responsibility to maintain a high level of integrity and to promote the standing of my profession.

Arising from this responsibility I wish to affirm that this dissertation is the result of my own effort and that I have rigorously referenced and acknowledged all sources of information, writing and ideas used in this dissertation.

|  |  |
| --- | --- |
| **Author** | **OSAIGBOKAN GILBERT ENOFE** |

|  |  |
| --- | --- |
| **Signature** |  |

|  |  |
| --- | --- |
| **Date** | **25th August,2023** |

## Acknowledgements

I would like to thank my supervisor (Eric McNamara) for his assistance in guiding me through this research process. I would also like to thank my family for their support during our entire study period. I love you guys!

## Table of Contents

[Declaration of Ethical Practice i](#_Toc143973446)

[Acknowledgements ii](#_Toc143973447)

[Table of Contents iii](#_Toc143973448)

[List of Tables v](#_Toc143973449)

[List of Figures vi](#_Toc143973450)

[Glossary of Abbreviations vii](#_Toc143973451)

[Abstract viii](#_Toc143973452)

[Chapter 1 | Introduction 9](#_Toc143973453)

[1.1 Background of The Study 9](#_Toc143973454)

[1.2 Problem Statement 10](#_Toc143973455)

[1.3 Research Aim and Objective 10](#_Toc143973456)

[1.3.1 Research Significance 11](#_Toc143973457)

[1.3.2 Research Questions 11](#_Toc143973458)

[1.4 Dissertation Structure 12](#_Toc143973459)

[Chapter 2 | Critical Review of Literature 13](#_Toc143973460)

[2.0 Introduction 13](#_Toc143973461)

[2.1 Stock Market 13](#_Toc143973462)

[2.2 Historical Approaches to Stock Price Forecasting 14](#_Toc143973463)

[2.3 Time Series Forecasting 15](#_Toc143973464)

[2.4 Machine Learning (ML) 18](#_Toc143973465)

[2.4.1 Random Forest (RF) 18](#_Toc143973466)

[2.4.2 Extreme Gradient Boosting (XGBOOST) 19](#_Toc143973467)

[2.5 Related Work 20](#_Toc143973468)

[2.6 Identification of Research Gaps 24](#_Toc143973469)

[2.7 Conclusion 25](#_Toc143973470)

[Chapter 3 | Research Methodology 26](#_Toc143973471)

[3.1 Introduction 26](#_Toc143973472)

[3.2 CRISP DM 26](#_Toc143973473)

[3.3 Business Understanding 29](#_Toc143973474)

[3.3.1 Determine Business Objectives 29](#_Toc143973475)

[3.4 Data Understanding 32](#_Toc143973476)

[3.4.1 Data Collection 32](#_Toc143973477)

[3.5 Data Preparation 34](#_Toc143973478)

[3.5.1 Data Selection 35](#_Toc143973479)

[3.6 Modelling 42](#_Toc143973480)

[3.7 Evaluation 46](#_Toc143973481)

[3.8 Deployment 47](#_Toc143973482)

[3.9 Summary 47](#_Toc143973483)

[Chapter 4 | Research Findings 48](#_Toc143973484)

[4.1 Introduction 48](#_Toc143973485)

[4.2 Evaluation Methodology 49](#_Toc143973486)

[4.2.1 ARIMA 49](#_Toc143973487)

[4.3 Experimental Results of Proposed Models 50](#_Toc143973488)

[4.3.1 ARIMA Model Prediction 50](#_Toc143973489)

[4.3.1.2 ARIMA on GlaxoSmithKline (GSK) 51](#_Toc143973490)

[4.3.2 Prophet model prediction 52](#_Toc143973491)

[4.3.3 RF Model Prediction 54](#_Toc143973492)

[4.3.4 XBBOOST Model Prediction 56](#_Toc143973493)

[4.4 Model Comparison and Discussion 58](#_Toc143973494)

[4.5 Summary 59](#_Toc143973495)

[Chapter 5 | Conclusions and Recommendations 61](#_Toc143973496)

[5.1 Introduction 61](#_Toc143973497)

[5.1.1 Limitations and Future Works 61](#_Toc143973498)

[5.1.2 Summary 62](#_Toc143973499)

[Appendix 63](#_Toc143973500)

[References 81](#_Toc143973501)

## List of Tables

[Table 1 - CRISP-DM process description Table 1 1 (Schröer, et al., 2021). 27](#_Toc143875361)

[Table 2 - Summary of the Models. 56](#_Toc143875362)

## List of Figures

[Figure 1 - Time Series Chart (David Dalisay, 2023). 15](#_Toc143874803)

[Figure 2 - CRISP-DM Process Model (Alliance, 2021). 27](#_Toc143874804)

[Figure 3 - Stock financial data. 31](#_Toc143874805)

[Figure 4 - Unilever Adjusted Close Price plot from 1988-07-01 to 2023-07-19. 33](#_Toc143874806)

[Figure 5 - GlaxoSmithKline Adjusted Close Price plot from 1988-07-01 to 2023-07-19. 33](#_Toc143874807)

[Figure 6 - Decomposition of GSK.L. 35](#_Toc143874808)

[Figure 7 - Decomposition of ULVR.L. 36](#_Toc143874809)

[Figure 8 - ACF for GSK. 37](#_Toc143874810)

[Figure 9 - ACF for ULVR. 38](#_Toc143874811)

[Figure 10 - PACF for GSK. 38](#_Toc143874812)

[Figure 11 - PACF for ULVR. 39](#_Toc143874813)

[Figure 12 - Actual vs Forecasted (ARIMA) for ULVR.L. 47](#_Toc143874814)

[Figure 13 - Actual vs Forecast (ARIMA) for GSK.L. 48](#_Toc143874815)

[Figure 14 - Prophet Actual vs Forecast For ULVR.L. 49](#_Toc143874816)

[Figure 15 - Prophet Actual vs Forecast for GSK.L. 50](#_Toc143874817)

[Figure 16 - RF Actual vs Forecast on ULVR.L. 51](#_Toc143874818)

[Figure 17 - RF Actual vs Forecast on GSK.L data. 52](#_Toc143874819)

[Figure 18 - XGBoost Actual vs forecast on ULVR.L. 54](#_Toc143874820)

[Figure 19 - XGBoost Actual vs forecast of GSK.L data. 55](#_Toc143874821)

## Glossary of Abbreviations

|  |  |
| --- | --- |
| **Abbreviation** | **Full Title** |
| FS | Finacial Stock |
| ML | Machine Learning |
| XGBoost | Extreme Gradient Boosting |
| IPO | Initial Public Offering |
| NYSE | New York Stock Exchange |
| NSE | Nigerian Stock Exchange |
| ARIMA | Autoregressive Integrated Moving Average |
| MA | Moving Average |
| AR | Autoregressive |
| KSE | Karachi Stock Exchange |
| FEX | Foreign Exchange |
| RF | Random Forest |
| MSE | Mean Squared Error |
| SVM | support vector machines |
| RMSE | Root Mean Squared Error |
| KNN | k-nearest neighbors |
| ANN | Artificial neural networks |
| MLP | multi-layer perceptron |
| MLR | Multiple Linear Regression |
| LSTM | Long Short-Term Memory |
| CRISP-DM | Cross -Industry Standard Process for Data Mining |
| PSO | Particle swarm optimization |
| ACF | Autocorrelation Function |
| PACF | Partial Autocorrelation Function |
| CV | Cross-Validation |
| LS-SVM | least squares support vector machine |
| OTC | Over-the-counter |
| AI | Artificial Intelligence |
| ULVR.L | Unilever PLC |
| GSK.L | GlaxoSmithKline |
| LSE | London Stock Exchange |

## Abstract

Stock price forecasting plays an important role in financial decision-making and risk management. This thesis aims to perform a comparative analysis between two statistical methods, ARIMA (Autoregressive Integrated Moving Average) and Prophet, and two machine learning methods, Random Forest (RF) and Extreme Gradient Boosting (XGBoost), in the context of stock price forecasting. The study examines their effectiveness, accuracy, and suitability for capturing the complex dynamics of the stock markets to determine which has the best performance when predicting future prices in the stock market. Two stock data (UNILEVER AND GLAXOSMITHKLINE) listed in the London Stock Exchange were used in the study. After analysis, RF emerged as the champion model for the GlaxoSmithKline stock data and XGBOOST was the champion model for the UNILEVER stock.

# | Introduction

Forecasting the prices of stocks has been a topic of great interest within the financial domain. Hence, Precise and timely forecasts are crucial for investors and traders to make informed decisions (Li, et al., 2014). Furthermore, analyzing market information can lead to better profits and substantial returns if done correctly (Oyewola, et al., 2019). Over the years, various methods and techniques have been developed to forecast stock prices, ranging from traditional statistical models to advanced machine learning (ML) algorithms. The increasing availability of financial data, coupled with advancements in computational power, has paved the way for the exploration of novel approaches to stock price forecasting. This thesis focuses on the comparative analysis of two categories of methods: traditional statistical models, AutoRegressive Integrated Moving Average (ARIMA) and Prophet, and modern ML techniques, Random Forest (RF) and Extreme Gradient Boosting (XGBoost). Findings from the research aims to assist financial institutions and investors by facilitating more precise forecasts of stock prices (Singh et al., 2019).

In this chapter, the introduction of the stock market, the problem statement, the research aims and objectives, the research questions, and the structure of the thesis will be discussed.

## 1.1 Background of The Study

The stock market has long been a subject of extensive scrutiny by investors due to its vast potential for returns (Daubechies, 1992). Consequently, stock forecasting remains a perennial area of great interest to both scholars and investors alike (Frankel, 1995). Moreover, the stock market plays a significant role in the financial market of a nation, reflecting and influencing the state of the economy in significant ways (Gençay, Selçuk and Whitcher, 2001). Although research into stock market predictions has historically been met with scepticism, it is nevertheless valuable in understanding the principles that govern specific market shifts and developments (Fama and Blume, 1966). Stock market analysis has dramatically benefited from the preservation of a significant amount of financial data thanks to technological progress (Fama and French, 1988).

The stock market is a marketplace for the purchase and sale of publicly traded company shares. The company initially generates funds by issuing an initial claim, which is known as an initial Public Offering (IPO). The stock market's price will rise or fall depending on the forces of demand and supply. The greater the investor's willingness to buy, the higher the stock price will rise, and vice versa. Although the quantities of sales and purchases may be calculated, it is extremely difficult to determine what factors contribute to these transactions (Faria and Verona, 2018). This could be due to a variety of factors, such as market behavior, inflation, trends, investor sentiment, and other news. Hence, the value of precise and timely information in understanding and forecasting stock market developments cannot be overstated (Sianturi and Rukmi, 2022). By carefully forecasting future trends, the investor seeks to generate income when making stock market investments.

## 1.2 Problem Statement

In this modern era of online trading, everyone has a chance to reap substantial rewards from the stock market. Therefore, if we can accurately forecast market behaviour, investors will know when and where to put their money. Professional traders study equities using time-honoured processes and procedures before making investing decisions. In the conventional approach, our gaze turns to the firm's revenue, market position, annual growth rate, etc., whereas in the technical approach, we look at how the price of the company has changed in the past. The decision of whether to enter or exit the market is based on a technical analysis of stocks, which considers the relationship between stock price and company fundamentals. This research aims to anticipate the upcoming stock price using both ML and statistical algorithms.

## 1.3 Research Aim and Objective

The thesis aims to provide accurate and timely forecasts that may help investors, traders, and financial institutions make educated choices in the volatile and complicated financial markets. The primary objective of this research is to conduct a comparative analysis of the effectiveness and performance of ARIMA, Prophet, Random Forest, and XGBoost in forecasting stock prices. Also, we aim to Compare how different models perform in terms of speed and accuracy. By evaluating these methods across different market conditions and periods, we aim to provide insights into the strengths, limitations, and potential use cases of each approach. By focusing on these goals, this study aims to enhance stock price forecasting using ml and statistical models, resulting in more accurate and dependable forecasts that may help market players make better investment choices.

### 1.3.1 Research Significance

This thesis research would help multiple stakeholders involved in the stock market in distinct ways. For investors, the findings could potentially guide investment decisions. This will help investors choose which stocks to invest in based on the forecast. Such characteristics would simplify the process of selecting an investment and save money by eliminating the need to employ third parties to forecast stock values for investors. Also, Financial analysts can benefit from insights into each approach's relative strengths and weaknesses, allowing them to tailor their strategies based on empirical evidence. Moreover, the study contributes to the broader academic discourse by addressing an existing research gap and advancing the understanding of the applicability of both traditional statistical techniques and modern ML algorithms in stock price forecasting. Overall, such a system would benefit not only investors but also authorities who monitor the market. Although the forecasts may not be 100% accurate, the system provides an early indicator of whether the stock price will rise or fall.

### 1.3.2 Research Questions

The following research questions will be discussed in this research.

1. How do ARIMA, Prophet, RF, and XGBoost models perform when applied to stock price forecasting? The purpose of the research topic is to undertake a conjunctive literature review and experiment to determine which ML techniques and statistical methods may be best applied to the stock market and yield the greatest results in forecasting stock prices.

1. What are the implications of the findings for investors, financial analysts, and decision-makers involved in stock market predictions? The aim of the thesis is to provide accurate and timely forecasts that may help investors, traders, and financial institutions make educated choices in volatile and complicated financial markets.
2. How do the selected methodologies perform in short-term versus long-term stock price forecasting?

### 1.4 Dissertation Structure

This thesis is structured as follows:

**Chapter 2:** Literature Review and related work

Under the Literature Review section, recent work that has been carried out in the field of stock price prediction will be evaluated. The time resolution of the forecasted data will be the focus of this section. Furthermore, This section provides a quick overview of ML and Statistical techniques. Based on the literature review that was done, knowledge gaps will be identified and included.

**Chapter 3:** Research Methodology

The architecture and working procedures of the proposed techniques and data descriptions are described in detail in this chapter. Specifically, it shows how raw data is gathered and cleaned up, features are extracted, and finished datasets are made. It also summarizes the various classifiers and statistical methods that have been applied to the data. The CRISP-DM strategy is used to structure the research methodology section of this thesis. The main steps of CRISP-DM were followed: business understanding, data understanding, data preparation, modelling, evaluation, and deployment.

**Chapter 4:** Data Analysis and Discussion of Findings

This chapter will present the results of the research. It Presents the findings of the experiments conducted, analyzes the performance of different ML algorithms, and examines the impact of feature selection techniques and preprocessing methods.

**Chapter 5:** Conclusion

Under this section, we will evaluate how the research has achieved the research objective and

identify insights on future research opportunities as an extension of this work.

**Chapter 6:** References

Under this section, research papers, journals, and other sources used in the thesis will be referenced.

**Chapter 7:** Appendix

# | Critical Review of Literature

## 2.0 Introduction

In the last few decades, there have been several significant changes in how financial markets operate. Investors now have more options because of advancements in trading and communication methods. Scientists have been enamoured with the prospect of forecasting stock market performance for a long time. Basic historical data that is publicly available is presumptively linked to potential future stock returns (Enke and Thawornwong, 2005). The processed information required to generate these predictions can be extracted from the existing data using data mining algorithms. Nevertheless, a few academics have focused on technical analysis and the use of cutting-edge math and science. Hence, data mining techniques and artificial intelligence have drawn much interest (Wang and Chan, 2006).

This section introduces the stock market, contemporary research on ML, and statistical methods for tracking stock market trends. It also reviews current developments in stock price forecasting.

### 2.1 Stock Market

The inception of the stock market dates to the 17th century (Stringham and Curott, 2015). The stock market is a marketplace for buyers and sellers of stocks and financial securities, primarily publicly traded corporate shares. It is a central marketplace where businesses, institutions, and individuals can trade stocks that represent ownership in publicly traded companies. By issuing shares to investors, businesses can raise capital. Various people invest in businesses they see favourably in exchange for company equity. The forces of supply and demand in the stock market determine the prices of stocks (Aldhyani and Alzahrani, 2022). Stock prices fluctuate depending on several variables, including company performance, the state of the economy, market sentiment, and investor expectations (Smith, 2003). In addition, share price movements are influenced by investor perceptions of company performance, with increasing prices signalling positive expectations and decreasing prices reflecting negative sentiments (Shiller, 2015).

### 2.2 Historical Approaches to Stock Price Forecasting

#### 2.2.1 Fundamental and technical analysis

Predicting stock prices is incredibly challenging due to the high uncertainties involved. Hence, Forecasting the stock market holds immense allure in both academic discourse and practical business endeavors. (WANG and CHAN, 2006). Currently, a range of techniques are used by financial professionals to examine and assess stocks. These techniques would enable short or long-term forecasting of the shares' value. Historically, the two basic techniques for evaluating stocks are fundamental analysis and technical analysis (Gitman et al., 2015;Lo et al., 2000). Investors have used these two major approaches to make financial decisions to make high profits while minimizing risks when investing in stocks (Arévalo, et al., 2017).

The financial health and operational performance of a particular organisation are the subject of fundamental analysis. Fundamental analysis uses financial accounts, economic indicators, and industry trends to determine a company's worth (Quah, 2008) . If the market quickly adjusts itself toward the present fundamental valuation of the company, it will indicate whether a stock price is undervalued or overvalued. Fundamental research helps investors set expectations for how the firm and its stocks will perform. Hence, it allows brokers to predict future stock values (Gitman, Joehnk, and Smart, 2011).

Conversely, Technical analysis is another widely used traditional method for stock price forecasting based on the belief that historical price and volume data, along with chart patterns and indicators, can provide insights into future price movements. It focuses on analysing market behavior rather than the health of the company's finances (Brown and Jennings, 1989). Technical analysis uses the historical stock prices of a company and trading volume information to determine what the stock price will be and make a trading decision. Furthermore, technical analysts use trend lines, moving averages, support and resistance levels, and various chart patterns to identify price trends. While technical analysis can provide short-term trading signals, it may not fully capture the fundamental factors influencing stock prices. Technical studies mainly predict the trend for a shorter period, like 1-3 months, and cannot decide the trend for a longer run, like one year (Wilder, 1978)**.** This problem can be solved by adding technical data to the fundamental data (Hargreaves and Hao, 2012). Furthermore, when these two analyses were combined, it was found to increase the accuracy of the trend prediction (Khairii, Zaki and Mahmood, 2019).

### 2.3 Time Series Forecasting

Time series forecasting is a valuable tool for predicting future values in various business, stock market, and weather forecasting applications. It aims to predict future values based on historical data patterns. Time-series data consists of observations recorded at different time intervals, making it a sequential dataset where the order of observations matters. A time series, such as the stock market index, is a collection of data points collected at regular intervals. It typically exhibits some patterns that reflect the progression of data points through time (Lin et al., 2003). The time series approach assumes a correlation between events that can be used to forecast the future. It is considered the most popular prediction in the financial business, especially if financial events have a history of repetition while the full scope of their implications remains unclear. A causal approach would look into the causes of past trends and then make predictions based on those findings. The experimental strategy entails conducting tests to get the necessary data for forecasting, whereas the judgmental approach entails gathering expert knowledge and opinions to foresee future events (Mahalakshmii, Sridevi and Rajaram, 2016). Line charts are widely used to plot time-series data. Figure 1. below displays the time-series chart.

A picture containing text, font, handwriting, screenshot

Description automatically generated

Figure 1 - Time Series Chart (David Dalisay, 2023).

In time-series forecasting, past performance is analyzed considering the relevant dynamics that determine it in order to make projections about the future. Thus, time-series forecasting is an approach that uses past data and the factors that affect performance to make predictions about the future (Hadavandi, Shavandi, and Ghanbari, 2010).

#### 2.3.1 Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a widely recognized time-series model that has been extensively applied to statistics, econometrics, and data analytics. It extends the Autoregressive Moving Average (ARMA) model by incorporating differencing, which helps address the limitation of ARMA in handling non-stationary time-series data. By combining autoregressive, moving average, and differencing components, the ARIMA model is a regression analysis technique widely employed for data analysis and forecasting future outcomes based on historical data (Ho and Xie, 1998).

Additionally, the integrated component involves differencing the data to achieve stationarity, enhancing model accuracy. ARIMA's effectiveness lies in capturing linear relationships within the data, making it a popular choice for short to medium-term stock price predictions. Furthermore, ARIMA has demonstrated its effectiveness as a valuable tool for analyzing small datasets (Fong et al., 2020). A study by (Ariyo et al. 2014) developed an ARIMA model for stock price prediction. They utilized stock data from both the New York Stock Exchange (NYSE) and the Nigerian Stock Exchange (NSE) to develop their predictive model.

The study's findings demonstrated that the ARIMA model exhibited significant potential for short-term prediction and showcased competitive performance compared to existing stock price forecasting techniques. Similarly, (Banerjee, 2014) investigated forecasting the Indian stock market using the ARIMA time series model. The study assesses the effectiveness of the ARIMA model in predicting stock market trends in India. Findings from the research indicated that the ARIMA model demonstrated promising results in forecasting the Indian stock market. The model captured the patterns and trends in the time-series data, providing valuable insights for investors and market analysts.

##### **2.3.2 Prophet**

The Prophet model is an open-source forecasting method for time series developed by Facebook's Core Data Science team (Taylor and Letham, 2017). Using an additive approach, it separates data into trends, seasonality, and holidays. It can automatically recognize seasonal trends, deal with missing data, and take holidays into consideration (Satrio et al., 2021). The additive model is a statistical modeling approach used in time series forecasting, among other applications. It assumes that the observed values in a time series can be expressed as the sum of trend, seasonality, and residual. Prophet is represented by the equation below:

**y(t) = g(t) model) h(t) + et (Equation 1** – **The Prophet Mode**l **)**

where y(t) represents the predicted value, g(t) is the trend function, s(t) describes seasonal patterns, h(t) captures the influence of holidays, and et is the noise error term.

Prophet has become renowned for its user-friendliness, adaptability, and capacity to furnish precise and interpretable prediction. As a result, it has been employed in numerous applications across several disciplines in recent years. For instance, (Almazrouee et al., 2020) utilised Prophet in energy forecasting. They investigated using the Prophet model to forecast long-term peak loads in Kuwait. The model uses data from January 2010 to May 2020 obtained from the Kuwait Ministry of Power and Water to forecast the energy peak load for 2020 to 2030. Afterwards, the maximum energy load in Kuwait for the past ten years was forecasted, showing the yearly seasonality of the maximum load. It was observed that the maximum energy peak period is during the summer months of June and August, with an increasing trend. This was attributed to the extreme weather conditions in Kuwait during the summer and cold winter seasons, where temperatures are extreme.

To evaluate the performance of the models, statistical analysis like relative error, mean absolute percentage error (MAPE), root mean squared error (RMSE), and R-squared (R2) is used. The research findings show that Prophet had an R2 value of 99.42%, which shows that the model was successful and robust. Also, the study suggests that the period between November 2020 and March 2021 was the best month for scheduling maintenance on the Kuwait energy grid for the years 2020 and 2021. Also, (Yenidoğan et al., 2018) used Prophet in Bitcoin prediction. Bitcoin price historical data for the period from May 3rd, 2016, to August 30th, 2018, was used in making a 90-day Bitcoin price projection. Before dataset training is done using the R analytics platform, data preprocessing (timestamp conversion and feature selection) is done to add variables to the model to improve forecasting accuracy. These additional variables were selected based on different correlations between cryptocurrencies and real currencies, which were achieved by constructing a correlation matrix. To measure the performance of models and find the best splitting ratios, Mean Squared Error (MSE), Mean Absolute Error (MAE), and R2 are used. Findings show that the model had a high prediction accuracy close to real life, with an R2 value of 94.5%. Both studies showed the robustness of Prophet in making forecasts (Yenidoğan et al., 2018).

## 2.4 Machine Learning (ML)

ML is a branch of computational intelligence concerned with the problem of creating computer programs that automatically improve with experience(Mitchell, 1997). In recent years, ML has been used to develop various applications in industries like healthcare, finance, military hardware, and space exploration. Nowadays, ML is a very dynamic field that is always changing. By feeding them data, it improves the performance of computers (Ayodele Oladipupo, 2010). Also, ML can learn from input data without human intervention. It involves learning from the provided data to generate the desired output by identifying patterns and trends within the data (Rao and Gudivada, 2018). The availability of a large amount of data is essential for ML. The quality of the data used to train a ML algorithm increases with its accuracy. An algorithm's goal is to find a relationship between its input and its output. The data is split into two sets: the training and test sets. The training data is used to build the model, and the test data is used to see how effectively it performs. ML relies heavily on having access to high-quality data (Shen, Jiang and Zhang, 2012).

### 2.4.1 Random Forest (RF)

RF is an ensemble learning technique used for both regression and classification problems. The technique builds multiple decision tree models and combines their predictions to make more accurate forecasts. Each decision tree is trained on a random subset of features and data points, reducing the risk of overfitting, and increasing the model's robustness (Breiman, 2001). RF is known for its ability to handle high-dimensional data, capture non-linear relationships, and handle missing values effectively (Caiola and Reiter, 2010). In the context of stock price forecasting, RF has shown promising results due to its adaptability and ability to handle noisy and complex data. A research study by (Liu et al. 2016) uses RF to build a model to predict fluctuations in gold prices. A 30-day projection was created using gold prices after data preprocessing was carried out.

To improve the model's predictive accuracy, important factors such as the US dollar index (USDX), the crude oil price (COP), the Dow Jones Industrial Average (DJIA), the CPI of the US (USCPI), the prices of US ten-year bond futures (US10BFP), the Hang Seng Index (HIS), and the Standard & Poor’s 500 Index (S&P500) were all taken note of. Two important research findings were obtained; firstly, the RF proved to be a powerful predictive method in showing intricate trends and fluctuations within the gold price. Secondly, DJIA and S&P500 were the two factors necessary to improve the RF prediction model's performance. Similarly, (Lohrmann and Luukka, 2019) developed an RF model using data from the S&P 500 to forecast the intraday stock closing price. (Sadorsky, 2021) used RF to forecast energy stock prices. A 20-day projection for clean energy stock price was made using RF. The results obtained show a prediction accuracy of between 80% and 90% for the RF model.

This clearly shows RF has a high prediction accuracy in forecasting stock prices (Lohrmann and Luukka, 2019). Aside from stock prediction, RF has also shown promising results in other areas, including traffic accident prediction. According to the research (Yan and Shen, 2022), the RF algorithm was used to predict traffic accident severity and its potential to enhance road safety measures in the United States. A traffic accident dataset containing 2.25 million samples from February 2016 to March 2019 was used in the analysis. RF obtains a high precision value of 67%.

### 2.4.2 Extreme Gradient Boosting (XGBOOST)

XGBOOST is another powerful ensemble learning algorithm widely used for various ML tasks, including regression and classification. XGBOOST is an optimized implementation of the Gradient Boosting framework and is known for its speed, scalability, and high performance. It sequentially adds weak learners (typically decision trees) to the ensemble, focusing on minimising the loss function, which results in improved predictive accuracy (Chen and He, 2015). XGBOOST has gained popularity in financial forecasting due to its exceptional ability to handle large datasets, feature importance analysis, and effectively handle missing values. The Boosting algorithm is based on the concept of combining numerous weak classifiers to create a strong one. XGBoost, being a lifting tree model, combines various tree models to form a robust classifier (Chen and Guestrin, 2016). A study conducted by (Dey et al., 2016) uses XGBoost to forecast the direction of stock market prices. A 60-day and 90-day forecast projection was done utilizing Apple stock data obtained from Yahoo Finance.

The data set comprises columns such as closing price, opening price, High, low, and Volume. In building the model, the time-series historical stock data is exponentially smoothed before feature extraction is done. Further, a measuring parameter is applied to the dataset, with +1 used to indicate the rise in stock prices and -1 signifies a decline in prices. Then RMSE is used as the evaluation metric to test the model's accuracy for the 60-day prediction and the 90-day prediction. The model proved to be efficient, with over 87% accuracy for time spans of 60 and 90 days, it demonstrated significantly superior performance compared to conventional non-ensemble learning methods (Dey et al., 2016).

## 2.5 Related Work

Several studies have compared different forecasting methods for stock price prediction. This subsection reviews the existing literature on comparative analyses of forecasting models, focusing on studies that have evaluated the performance of ARIMA, Prophet, RF, XGBoost, and other stock prediction techniques. It highlights the strengths and weaknesses identified in previous research, paving the way for the in-depth comparison conducted in this thesis.

For instance, according to the research (de Faria et al., 2009), Artificial Neural Network (ANN) and Exponential smoothing methods (ESM) were compared and applied in predicting the Brazilian stock market. Results from the research showed that both methods had similar performance, but ANN outperformed ESM slightly when RMSE was used as the evaluation metric.

In a similar vein, (Ayushman Durgapal and Vrince Vimal, 2021) conducted a study comparing statistical and ensemble methods for predicting Google stock prices obtained from the NASDAQ stock exchange. The study compared the performance of ARIMA, RF, and XGBOOST using two evaluation metrics, RMSE and MAPE. The findings indicated that for short-term forecasts, the ARIMA model performed reasonably well but had high MAPE values. XGBOOST emerged as the top-performing model, displaying the lowest MSE and RMSE values. Additionally, the study highlights the potential of ensemble methods in building effective stock price prediction models, especially with proper hyperparameter tuning. Furthermore, (Lu and Wu,2009) conducted a study that utilized ANN to forecast future values of the S&P 500 Index, a prominent benchmark for the US stock market. The research paper compared the predictive performance of ANN against the ARIMA model. The outcomes revealed that ANN outperformed ARIMA, but this superiority was evident primarily during periods of market stability.

During volatile market conditions, ANN exhibited an accuracy rate of 23%, while the ARIMA model achieved a higher accuracy rate of 42%. This observation underscores the significant impact of market stability on the performance of ANN models in stock market predictions. In a related work conducted by (Kamruzzaman and Sarker,2003), the focus shifted from stock market data to exchange rates. In this study, ANN models underwent training with a variety of algorithms, including the incorporation of technical indicators to anticipate future values. Impressively, the results demonstrated an accuracy rate of 80%, surpassing that of the ARIMA model. These studies collectively illustrate the potential of ANN in predicting both stock market and exchange rate values. However, they also emphasize the influence of market stability and the choice of algorithm, which can significantly affect the accuracy of ANN models.

(Usmani et al.,2016) explore the effectiveness of various ML algorithms in predicting stock market trends on the Karachi Stock Exchange (KSE). The model includes current oil, gold, and silver prices, interest rates, the Foreign Exchange (FEX) rate, news, and social media. The findings of the study revealed that ML techniques, including support vector machines (SVM), k-nearest neighbors (KNN), and ANN, showed promising results in predicting stock market movements. These algorithms could capture complex patterns and relationships within the stock market data, leading to accurate predictions. The multi-layer perceptron (MLP) algorithm produced the best results. The oil rate feature proved particularly useful in making accurate market forecasts. The Multi-Layer Perceptron could accurately forecast market performance by a margin of 70% (Usmani et al.,2016).

Similarly, (Nivetha and Dhaya,2017) compared and contrasted between three distinct algorithms: Multiple Linear Regression (MLR), SVM, and ANN. The market prices for the next day were predicted using a combination of monthly forecasts and daily projections. Sentiment analysis is utilized to forecast stock prices, employing the most effective prediction methodology. The MLR approach is the least developed of the three because it measures the relationship between volume and stock price. In contrast to MLR and SVM, deep learning algorithms are shown to have advanced further in the study. A study (Hossain et al. 2018) applied deep learning models to forecast stock prices for the S&P 500. In building the deep learning model, the researchers examined two variations of Long Short-Term Memory (LSTM) and Gated Recurrent units (GRU). The stock's closing price was forecasted for the following day. The dataset used in the analysis was divided into two halves, with 80% of each used for training and the remaining 20% preserved for testing.

The training set contains historical data from 1950 to 2002, and the rest of the data until 2016 is used for testing. 10% of the training data was used for validation as well. The models were evaluated using three metrics, Mean Absolute Error (MAE), Mean square error (MSE), and Mean Absolute Percent Error (MAPE). To evaluate the models' performance, the average of the two models was utilized as the error rate. Results obtained show the hybrid network achieved MSE = 0.00098 in making predictions, which outperforms all the previous neural network approaches that utilized the same dataset. Also, LSRM (two layers) had MAE = 0.086, MSE = 0.018, and MAPE = 11.583, and GRU (two layers) had MAE =0.028, MSE = 0.001, and MAPE = 4.649 (Hossain et al., 2018).

Additionally, a study by (Rounaghi and Nassir Zadeh, 2016) focuses on modelling and forecasting the stock prices of 350 companies listed on the LSE and S&P 500 between 2007 and 2013, using the ARMA model. Monthly and yearly forecasts were projected for both markets. Various metrics, including MAE, MAPE, Median Absolute Percentage Error (MDAPE), Symmetric Median Absolute Percentage Error (SMDAPE), and Mean Absolute Scaled Error (MASE), were employed to assess the model's performance. Research findings revealed that the ARMA model achieved medium and long-term forecasting with an error level of 1% in both the S&P 500 and the LSE. Moreover, efficiency and financial stability were observed in both markets, even during periods of economic boom and bust. Furthermore, the LSE had MASE values for monthly and yearly stock forecasts of 0.0074 and 0.034 respectively, while the MASE values for monthly and yearly forecasts for the S&P 500 were 0.0071 and 0.0043 respectively.

Furthermore, in predicting the closing stock price of Google stock during the COVID-19 pandemic, (Jin, Gao and Tao, 2022) used data from Google stock in building the model as well as comparing the accuracy of both models. Firstly, in building the model, the dataset was checked for stationarity, and then the ARIMA(0,1,1) was utilized in making predictions about the stock price during the pandemic. The Prophet model was trained using the stock price before January 1, 2021, and predictions were made for stock price after January 1, 2021. Model comparison was also done using prediction graphs for both models. The results obtained show that the ARIMA model outperformed the PROPHET model in predicting Google’s stock price during the pandemic. In a similar vein, (Vantuch and Zelinka, 2015) explore the application of evolutionary algorithms to improve the accuracy of ARIMA models for stock price prediction. Findings from the study indicated that incorporating evolutionary algorithms, such as genetic algorithms or particle swarm optimization, into the ARIMA modeling process can enhance forecasting performance. By optimising the ARIMA model parameters using evolutionary techniques, the researchers achieved more accurate stock price predictions. The study demonstrated the potential of evolutionary-based ARIMA models as a viable approach for stock price forecasting. It highlighted the benefits of leveraging evolutionary algorithms to fine-tune the parameters of traditional time-series models, improving their ability to capture the complex dynamics of stock market data. (Lu et al, 2020) investigate the use of CNN-BiLSTM-AM model to predict the stock closing price of the next day in the Shanghai Stock Exchange (SSE). The CNN-BiLSTM-AM method comprises convolutional neural networks (CNN), bi-directional long short-term Memory (BiLSTM), and attention mechanisms (AM). CNN is utilized to extract the prominent features of the input data.

The extracted feature data, in turn, is utilized by BiLSTM to make predictions regarding the closing price of stocks for the ensuing day. To enhance the accuracy of the predictions, AM is employed to capture the impact of feature states on the stock's closing price at various points in time in the past. Thereafter, a prediction projection for the stock closing price of the next day and the next 1000 trading days is made. The results obtained show that the performance of this model is robust, with an MAE value = 21.952, RMSE = 31.694, and an R2 = 0.9804. In contrast to other method, the CNN-BiLSTM-AM technique exhibits greater aptness for stock price forecasting and furnishes investors with a dependable means by which to formulate investment decisions.

From a different viewpoint, (Tiwari, Bharadwaj, and Gupta, 2017) explore the potential of applying data analytics to facilitate informed investment decisions. The study focused on utilizing Python and R as the primary platforms for analysis. The study utilized data from the Nifty 50 (ANSEI) stock market index, collected over a span of nine years. Various methods available in R, such as Arima, Holt Winters, neural networks, linear regression, and time series, were employed to predict the opening index price performance. For the prediction of the Nifty 50 stock price, Python's Multi-layer perceptron and support vector regression models were utilized, along with sentiment data gathered from Twitter. To assess the accuracy of the predictions, the researchers compared the outcomes of the R and Python models with the actual stock prices over a period ranging from two to three years. MSE were computed for each prediction system. The results indicated that the feed-forward network achieved a relatively low error of 1.81598342% when forecasting the opening price of the stock.

Research by (Hegazy et al.,2013) uses ML to build a stock market prediction model. Particle swarm optimization (PSO) and the least squares support vector machine (LS-SVM) were used in this method. Over-fitting and local minima can be avoided by utilizing the PSO algorithm to find the best possible free parameter combination for LS-SVM by examining historical stock data and technical indicators. Thirteen industry-standard financial datasets were used to analyze and compare the proposed model to a synthetic neural network built with the Levenberg-Marquardt (LM) algorithm. Prediction accuracy was higher for the proposed model, and the data suggested that PSO may be used to optimize LS-SVM. The study (Mittal and Goel, 2011) investigates the application of sentiment analysis and ML principles to forecast stock prices. To establish a link between public opinion and market reaction, sentiment analysis and ML techniques were used. Twitter feeds data and DJIA values from June to December 2009 were utilized to forecast public sentiment, which was then used to forecast stock market direction, together with DJIA values from the previous trading day. Using Self-Organizing Fuzzy Neural Networks (SOFNN) for cross validation, they discovered an accuracy of 75.56 percent. Also, (Shen, Jiang, and Zhang,2012) propose an alternate forecast technique that uses support vector machines to estimate the movement of stocks in the following trading day based on the temporal correlations between international stock markets and other financial instruments. Daily stock price data from 04-Jan-2000 to 25- Oct -2012 were collected from different financial markets such DJIA, S&P 500 and National Association of Securities Dealers Automated Quotations (NASDAQ) and utilised in the research. Results obtained shows a forecast precision of 74.4% for NASDAQ, 76% for S&P500, and 77.6% for DJIA.

## 2.6 Identification of Research Gaps

After extensive Literature evaluations on stock price prediction, it became clear that there were certain gaps in understanding. The absence of studies or the necessity for additional research are examples of knowledge gaps. Here is a list of them:

1. Very limited research has been conducted to Compare Statistical Methods and ML techniques for forecasting the stock exchange (Spiliotis et al., 2020). While some studies have compared statistical methods with ML techniques in stock price forecasting, there is still a lack of comprehensive research that directly compares the performance of these methods on the same data set. Many existing studies focus on evaluating either statistical methods or ML algorithms separately, making it challenging for investors or analysts to identify the most suitable approach for their specific forecasting needs.
2. Most academics have made predictions based on insufficiently available data. In the early days of stock exchange research, when there was nothing to go on, many studies were conducted. Given the minimal data employed by most researchers, their model would not have been able to obtain more intricate correlations and trends.
3. Stock price forecasting is crucial for investors looking to make long-term investment decisions. Research by (Khare et al., 2017), and (Ince and Trafalis, 2008) tends to focus on short-term forecasting of the stock market. However, there is a need to evaluate the long-term predictive capabilities of ARIMA, Prophet, RF, and XGBoost. Understanding their performance over longer periods can help investors make better-informed investment decisions.

## 2.7 Conclusion

The chapter provided a brief overview of the stock market and stock price forecasting methods. Also, Statistical methods (ARIMA and Prophet) and ML algorithms (RF and XGBoost) for stock forecasting were also discussed. It highlighted the need to comprehensively compare these methods on the same dataset, considering their strengths and weaknesses. The chapter identified research gaps related to robustness under different market conditions, long-term forecasting, and insufficient available data. These research gaps serve as the basis for the subsequent chapters, guiding the comparative analysis and contributing to the field of stock price forecasting.

# | Research Methodology

* 1. **3.1 Introduction**

Research is conducted to gather more information about a specific topic. It is a means of acquiring data that may be used to improve the existing system put in place by other researchers and advance the topic at hand. The research project would focus on researching historical stock price data and learning the principles of ML and time-series forecasting.

Most of the data utilized to construct this system is secondary data or information that does not require any new collection or analysis on the user's part. In other words, the python module known as "yfinance" is the most common method of getting data from reputable sources like Yahoo Finance. This research will leverage the Cross Industry Standard Process for Data Mining (CRISP-DM) in structuring the research methodology section of this paper.

**3.2 CRISP DM**

The CRISP-DM framework was created in 1999 and is a widely recognized framework for data science projects (Alliance, 2021). The framework serves as the established standard and a universally applicable process model for executing data mining projects across various industries (Schröer et al., 2021). Data-driven projects often adhere to the data science life cycle. CRISP-DM's primary goal is to improve team collaboration, which is often neglected in task-oriented project management. Data mining (DM) and ML can be used to interpret and analyse big datasets, and this tool aids data scientists in collecting, storing, and processing this data.

CRISP-DM comprises six phases: business understanding, data understanding, data Preparation, modeling, evaluation, and deployment. Each phase has specific tasks and objectives that help ensure a systematic and effective data mining process. Figure 2. illustrates the CRISP-DM six phases and how they are sequenced in a typical data analytics project.

Diagram of data processing process

Description automatically generated

Figure 2. - CRISP-DM Process Model (Alliance, 2021).

The research method will be structured using the CRISP-DM process approach. Table 1 provides a basic description of each phase, task, and output of these six iterative phases (Schröer et al., 2021).

Table 1 - CRISP-DM process description (Schröer, et al., 2021).

|  |  |
| --- | --- |
| **Phase** | **Description** |
| **Business Understanding** | During this stage, the focus is on comprehending the business goals, objectives, and requirements of the project. It includes defining the problem statement, identifying the project scope, and determining the success criteria. An evaluation of the business situation is also conducted to gain an overview of both the existing and necessary resources. The steps include 1) deciding the project output, 2) Current situation, 3) goals, and 4) Project Plan. |
| **Data Understanding** | During this stage, the process of gathering and exploring data takes place in other to gain insights into its characteristics, quality, and relevance to the project. Data sources are identified, and initial data exploration and descriptive statistics are performed to understand the data structure and identify any data quality issues. 1) Collect data; 2) Describe data; 3) Explore data; and 4) Validate data quality**.** |
| **Data preparation** | The data is pre-processed here in preparation for modelling. It involves cleaning, integrating, transforming, and selecting the data to create a suitable dataset for analysis. Tasks such as handling missing values, outlier detection and treatment, feature engineering, and data sampling may be performed in this phase. 1) Select data; 2) Clean Data 3) Construct data; 4) Merge Data. |
| **Modelling** | This modelling phase focuses on selecting and applying appropriate data mining techniques to build predictive or descriptive models. The model's performance is evaluated using different statistical tools. 1) Select modelling techniques; 2) Generate Test data; 3) Build the model; and 4) Assess the model**.** |
| **Evaluation** | Here, the performance of the models is evaluated against the project objectives and success criteria. The models are tested using validation datasets or cross-validation techniques. 1) Evaluate Result 2) Review Process 3) Next Step |
| **Deployment** | The final phase involves deploying the selected model into the production environment. This may include integrating the model into existing systems, creating user interfaces or APIs for model access, and establishing monitoring and maintenance procedures. |

## 3.3 Business Understanding

### 3.3.1 Determine Business Objectives

Developing a stock price forecasting model that can accurately predict price movements in the stock market will provide valuable insights to investors, traders, and financial institutions, enabling them to make informed decisions and improve their investment strategies. By selecting the most suitable ML algorithms and time series techniques for statistical analysis, the model will enhance the decision-making process for both short-term and long-term investments. Moreover, the precise prediction of stock price movements will aid stakeholders in maximizing their returns and minimizing potential risks in the stock market (Kumar, Sarangi and Verma, 2021). This forecasting model will significantly contribute to better investment outcomes and support effective decision-making in the dynamic and constantly evolving stock market milieu.

The precision of the stock forecasting model in predicting stock price movements will be a crucial indicator of the dissertation's success. This data can be utilized by investors, traders, and financial institutions to make well-informed decisions concerning stock investments, portfolio management, and risk mitigation strategies. The ability to accurately predict stock prices can contribute to maximizing returns and minimizing potential losses in the stock market.

Additionally, precise predictions of stock price movements can aid businesses in planning manufacturing, maintenance, and operational activities, ultimately resulting in improved financial performance and competitiveness (Spiliotis et al., 2020). By diminishing uncertainty and providing valuable insights into stock market trends, the stock prediction model can facilitate stakeholders in optimising their decision-making processes and accomplishing desired outcomes in the dynamic and volatile stock market environment.

#### 3.3.2 The nature of the stock market data

The proliferation of modern economies depends heavily on big data. Business organisations rely heavily on it to codify their collective ideas and optimal techniques. The healthcare industry has also used it to gain valuable insights and information to enhance today's healthcare systems. Big data also greatly benefits cloud computing, information technology, and other related fields. The banking and finance industries use big data to monitor the financial markets (Kavya et al 2021).

The stock market is simply a network of markets and exchanges where the issuance and trading of shares of publicly traded corporations take place. The financial transactions in question occur on regulated, official exchanges or in over the counter (OTC) marketplaces that adhere to a uniform set of rules and regulations. In any given country or location, there may be any number of stock trading venues where stocks and other securities can be bought and sold (Kavya et al 2021). When someone says they trade in the stock market, they mean they purchase and sell shares or stocks on the stock exchanges that make up the stock market. Though the stock market or equity market is most associated with the trading of individual stocks or shares of stock. Additionally, it enables the trading of various financial instruments, including exchange-traded funds (ETFs), corporate bonds, and derivatives linked to stocks, currencies, commodities, and bonds. (Chen, 2022). Being able to offer shares of stock to the public is a key part of becoming a publicly traded firm and subsequently raising funds for growth. Stock market liquidity allows traders to facilitate the swift and simple sale of their customers' assets. When compared to less liquid investments like real estate and other real estate, this is a major benefit of stock investing (Kavya et al 2021).

##### **3.3.3 Research Goals**

This research aims to develop a system that can accurately forecast or predict stock price movements in the stock market. An analysis of historical data will be done to identify patterns and insights pertaining to fluctuations in stock prices. A comparative analysis of two statistical models, ARIMA and PROPHET, and two ML models, RF and XGBOOST, will be conducted to ascertain the most accurate forecasting model.To evaluate the performance of each model, the MSE metric will be employed as the evaluation metric to determine the best-performing model. The model with the least minimum forecasting error using this metric will be selected as our champion model. These various goals must be created and met to achieve this study's aim. These goals can be summarized as follows:

1. Data Preparation

2. Exploratory data analysis

3. Data pre-processing

4. Implementation of Statistical models (ARIMA and Prophet) using the training data

5. Implementation of ML models (RF and XGBOOST) using the training data

6. Evaluation of results using the test data

###### 3.3.4 Produce project plan

The project plan will outline the various stages and activities required to successfully complete the research methodology. The project plan serves as a roadmap that guides the entire study process. In this thesis, the project plan will consist of multiple sections, with the second section addressing the basic tools and approaches. The project plan will also outline the activities to be undertaken throughout the study process.

1. **Project plan**

The project plan pertaining to this thesis focuses on stock price movement forecasting using data sourced from the Yahoo Finance Platform. The platform provides a wide range of data and tools for investors and traders, updated daily. Furthermore, It offers access to real-time stock quotes, historical price data, financial statements, news articles, and other relevant information for various financial instruments, including stocks, bonds, commodities, and currencies, spanning several years. A thorough explanation of the chosen data will be presented, with particular attention given to the variables it encompasses, their respective definitions, and any necessary procedures required to preprocess the data, ensuring its utmost quality and suitability for utilization in predictive models.

1. **Initial assessment of tool and techniques**

Considering their potential impact on the entire project, it is imperative to evaluate the selected tools and procedures. Python and R are two commonly used languages in data mining projects. These open-source languages have extensive libraries and frameworks that offer various tools and techniques that have strengths and limitations in data-driven innovation, Artificial intelligence (AI), and ML. Despite the many similarities between Python and R, one key distinction between the two is that Python is a general-purpose language, whereas R's roots are in statistical analysis. Therefore, Python is the preferred language for this project, as the research focuses primarily on statistical models and forecasting techniques. The forecast packages in Python offer several summary measurements of forecast accuracy for time series models. Moreover, Python is renowned for its user-friendly nature, adaptability, and extensive collection of libraries, such as NumPy, Pandas, statsmodel and Scikit-Learn. It provides an extensive range of statistical and ML tools that can be utilized for data analysis, modelling, and visualization. Python's ease of use has made it a preferred choice for data scientists and researchers.

Additionally, packages like Matplotlib and plotly can be combined to create interactive plots in Python, while NumPy and Pandas can be used for data wrangling.

Furthermore, Python's versatility allows for seamless integration with other tools and frameworks, making it suitable for tasks that require complex data processing pipelines and large-scale data analysis, such as developing and deploying ML models for stock price prediction. It is also an ideal choice for the entire data mining process, encompassing data cleaning, wrangling, modelling, analysis, visualisation, and deployment. Exploratory data analysis is valuable in discovering hidden patterns within data, leading to a better understanding of the information, and facilitating the development of dynamic models. To this end, Jupyter Notebook is used for data exploration. It is an interactive business intelligence software for data visualisation that can handle large volumes of data quickly and offer more options for data display. As there are no constraints on the number of data points, rows, or sizes, Jupyter Notebook enables the data to be viewed from every perspective.

## 3.4 Data Understanding

### 3.4.1 Data Collection

Yahoo! Finance provides free daily historical stock price data. Yahoo! Finance is a portal for business and financial information and news, stock quotations, reports, and releases. Yahoo! Finance has become the de facto standard for stock market research. From the day a firm first began trading on the stock market until the present, investors have had unrestricted access to the company's historical stock price data. (Hadavandi et al. 2010). To get all the useful FS data for the model, we utilised the Python package 'yfinance' to scrape the Yahoo Finance library for ticker values.

The Unilever PLC (ULVR.L) and GlaxoSmithKline (GSK.L) tickers were utilized to scrape the FS information for this investigation. **All data collected throughout all years for the two chosen equities totals 8997 rows for each stock in the FS dataset**. The chosen data was restricted to weekdays, or business days. The term ‘business day’ is used in the context of financial analysis to describe a time series frequency that excludes non-business days, weekends, and holidays.

In this research, two major LSE market data will be used, and these are ULVR.L and GSK.L.

1. **Unilever PLC (ULVR)**

ULVR is a rapidly expanding business that produces household necessities. The company has been around since 1929 and has its headquarters in London. It serves the health and beauty, personal care, home care, nutritional, and frozen desserts markets. ULVR.L is the ticker symbol for ULVR, a public company whose common stock is traded on stock exchanges. ULVR shares are listed on the stock exchange, where they may be bought and sold by investors (Nayak, 2008). It is one of the companies that form the Financial Times Stock Exchange (FTSE) 100 Index, which consists of the 100 largest companies on the LSE based on market capitalization.

1. **GLAXOSMITHKLINE (GSK)**

GSK plc was incorporated with limited liability in England and Wales on December 6, 1999, under the Companies Act 1985, with the registered number 3888792. GSK plc's primary objectives are not limited or regulated under its Articles of Association, and thus are unrestricted under Section 31 of the Companies Act 2006 (GlaxoSmithKline plc - GSK 2012). The Group is a worldwide healthcare company that creates and discovers, develops, manufactures, and markets pharmaceutical goods such as vaccines, over the counter (OTC) medicines, and health-related consumer products (GlaxoSmithKline plc - GSK 2012). GSK's stock can be found under the symbol "GSK.L" on the LSE. GSK is also included in the FTSE 100 Index, which tracks the performance of the top firms trading on the LSE (yahoo finance, 2023).

#### 3.4.2 Data Description

In this section, we will investigate the data and learn about its properties. Historical data is the most important component of stock price data that is often used in the stock market. These data points are often utilized by traders and investors to study market patterns, determine various technical indicators, and make informed decisions about stock purchases, sales, and holdings (Shahi et al., 2020; Li et al, 2020). These elements of information are freely accessible via stock market platforms and financial data sources and can be received in historical or real-time versions for specific equities. On the first look at Figure 3.1, we see it has six columns, all related to the stock price. To understand the data, we look into the details of each column. Each data point is explained below.

1. Date: The date denotes a particular day or time interval during which the stock information is documented.
2. Open Price: The open price signifies the value at which a stock commenced its trading activity within a particular period.
3. High Price: The high price denotes the highest trading price attained by the stock within the specified time frame.
4. Low Price: The low price signifies the lowest trading price reached by the stock during the given period.
5. Close Price: The close price refers to the price at which a particular stock concludes trading activities at the conclusion of a predetermined time frame.
6. Adjusted Close Price: In the stock market, an adjusted closing price is a modified closing price that includes any company actions that may have affected the stock's value. This is distinct from the raw closing price, which represents the cash value of the stock's last transaction before the market shuts. The adjusted closing price considers events such as dividends, stock splits, and other business actions that can affect the value of the stock. Using the adjusted closing price, investors gain a more exact picture of the stock's true worth, allowing them to make more educated decisions when purchasing or selling the shares (Bea Bischoff, 2019).
7. Volume: The volume signifies the aggregate quantity of shares that have been exchanged pertaining to a specific stock within the given time period.

A screenshot of a computer

Description automatically generated

Figure 3 - Stock financial data.

## 3.5 Data Preparation

### 3.5.1 Data Selection

This entails identifying the appropriate type, source, and gathering tools. This historical dataset was provided by Yahoo Finance. As at the time of writing, both the GSK and Unilever dataset was from 1988-07-01 to 2023-07-19 and are being updated daily. Jupyter Notebook will be used to visualize and model Python programming language. The adjusted closing price was the feature chosen for this prediction. In the stock market, the adjusted closing price is a modified closing price that includes any company actions that may have affected the stock's value.

#### 3.5.2 Data cleaning

Upon examining the dataset and assessing its integrity, we find that both datasets collected are already clean and do not require any additional cleansing.

##### **3.5.3 Dataset Exploration**

Data exploration is an initial phase of the data analysis process, which allows us to examine and understand our dataset to gain insights, identify patterns, and make informed decisions. It involves summarizing, visualizing, and manipulating the data to uncover its underlying structure and characteristics. Data exploration is crucial because it helps uncover potential issues, outliers, and relationships that can inform subsequent steps in our analysis pipeline**.**

To provide a summary of FS dataset, we created plots for each of the two stocks selected in this analysis. A consistent date range was selected so that the trends could be compared over the same time frame from 1988-07-01 to 2023-07-19. Figure 4. a trend line graph for ULVR.L stocks from 1/07/ 1988 to 19/07/2023 while looking at the adjusted closing price, hence this gives a clear picture of the data. Looking at the visualization, it can be seen that adjusted stock prices was upward sloping from left to right up to the year 1999 where a steep decrease in the adjusted closing stock prices, hence, from research, the reason for this unexpected sharp fall was as a result of the decline of 1999 pretax profit which lead to difficult economic conditions in European and Latin markets (The Wall Street Journal, 2000). Nevertheless, as seen in the trends, Unilever adjusted prices gained momentum and began to increase gradually in the year 2000. This upward sloping increase in the adjusted closing price saw Unilever stock reach its peak between 2019 and 2020, hence this was the all-time Unilever adjusted closing price.

Figure 5 shows a trend line graph for GSK.L stocks from the 1/07/ 1988 to 19/07/2023, hence focusing on the adjusted closing price. This visualisation below shows the trend line graph in the adjusted closing from the year 1988, starting at its lowest. GSK.L stocks had a consistent increase in its adjusted closing price over the years, however, there was a bit of downward sloping trend which happened in the year 2004. Nevertheless, GSK.L achieved its highest all-time adjusted price in the year 20222.

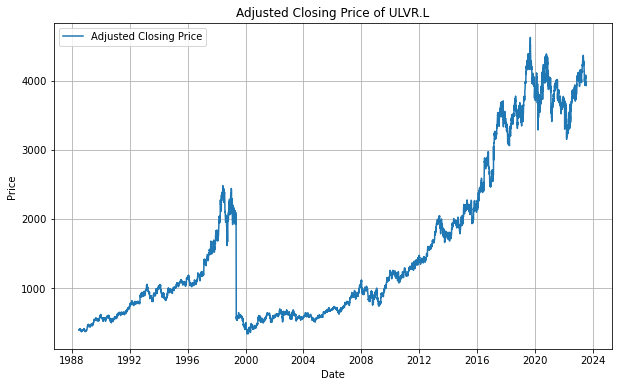


Figure 4 - Unilever Adjusted Close Price plot from 1988-07-01 to 2023-07-19.

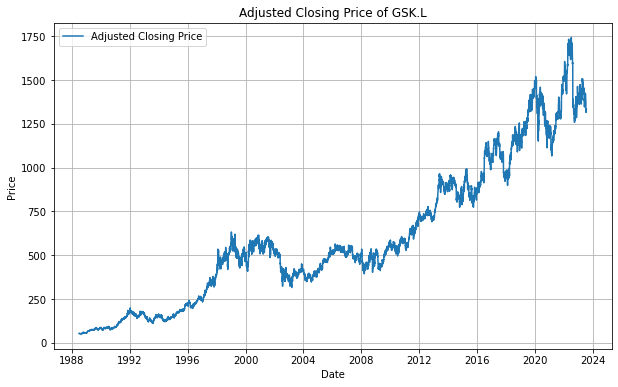


Figure 5 - GlaxoSmithKline Adjusted Close Price plot from 1988-07-01 to 2023-07-19.

###### 3.5.4 Data Preprocessing

**3.5.4.1 Times Series Decomposition**

Time series decomposition is a fundamental technique used to break down a time series data into its constituent components, revealing underlying patterns and structures. This technique is essential for understanding the various contributing factors that influence a time series, such as trend, seasonality, and residual noise. By isolating these components, analysts can gain insights into the inherent behaviour of the data and make more accurate predictions. Due to its noisy, non-stationary, and unpredictable nature, financial time series forecasting is one of the most difficult problems in time series prediction. Time Series Decomposition is an approach in data analysis that allows you to separate out distinct sources of variance (Huang et al., 1998). Decomposition presupposes the following about the data.

A black text on a white background

Description automatically generated

Equation 2

In the temporal data of a time series, three significant components are present: seasonality, trend, and random variation or noise (Jeffrey Xu Yu, Ng and Joshua Zhexue Huang, 2001).

1. Seasonality is a recurring movement that is present in time series variable. They are natural variations in the time series data, and they are reflected by the seasonal factor. It records recurrent seasonal cycles, such as those that occur on a monthly, quarterly, or yearly basis. Decomposing data allows us to isolate and analyze these seasonal components, which can be valuable for understanding regular patterns in stock price movements and identifying cyclic behaviors
2. Trends can be a long-term upward or downward pattern. Decomposing stock data helps identify the long-term trend, which is essential for understanding the overall direction in which a stock's price moves. Analysing trends can provide insights into whether the stock is generally increasing, decreasing, or experiencing periods of stability. The trend part of time series data indicates the general long-term trend or behavior. It reveals whether the overall trend in the data is one of expansion or contraction over time.
3. Noise is the component of variability in a time series that cannot be attributed to either seasonality or a trend. It represents the random, unpredictable fluctuations in the data.

When building models, you eventually merge different components into a mathematical equation. Seasonality and trend are two of the formula's components. A model that includes both will always have an inaccuracy in representing stock values. This is demonstrated by the noise factor.

The time series decomposition process aims to separate these components to gain insights into the behavior of the data. By doing so, analysts can better understand the factors influencing the data and make more accurate predictions .

A graph of data showing the number of data

Description automatically generated

Figure 6 - Decomposition of GSK.L.

From the graph above (Figure 6), it shows the original data, trend, seasonal and residual component respectively. The first graph shows the original data, the second graph shows the trend of the data in an upward-sloping trend which indicates that the data is increasing over time, suggesting a positive growth or a long-term upward movement. For the seasonal component graph, it shows that there is a significant and consistent seasonal pattern present in the data. This pattern can be seen as a dense or filled-in region in the graph. The filled-up appearance suggests that the seasonal component is contributing a substantial portion of the overall variability in the data. It indicates a prominent and consistent seasonal pattern in the data, which can provide valuable insights for understanding and modelling the time series.

For the residual graph, the residuals, also known as the error component, represent the random fluctuations or noise in the data that cannot be explained by the trend or the seasonal component. It consists of the differences between the observed data and the predicted values based on the trend and seasonal patterns. In this graph it appears as random fluctuations around zero. Randomness in the residuals indicates that the seasonal decomposition model has effectively captured the underlying trends and seasonal patterns, leaving minimal unexplained variability.

Figure 7. below shows the decomposition of ULVR.L data. This also have same interpretation as GSK.L data. The first graph is the visulization of the original data. The second graph shows the trend of the data which is upward sloping trend, and it indicates that the data is increasing over time, suggesting a positive growth or a long-term upward movement, while the third graph showing that there is a significant and consistent seasonal pattern present in the data in the seasonal component. The last graph displays that the seasonal decomposition model has effectively captured the underlying trends and seasonal patterns, leaving minimal unexplained variability.

A graph of a graph showing the number of years

Description automatically generated

Figure 7 - Decomposition of ULVR.L.

**3.5.4.2 Autocorrelation in Time Series**

The concept of autocorrelation is central to time series analysis. The correlation between the lagging values and the leading values of a time series. In other words, it determines how closely one piece of data is related to its predecessors at different intervals of time. Serial correlation is another name for this concept. It helps to reveal hidden relationships and patterns in data. Data is said to be autoregressive if it may be used to predict a future value based on a known past value. In a similar vein, an autoregressive model is one that makes projections about the future by looking at the past.

Auto-Correlation Function (ACF) plot illustrates the relationship between the current value of a time series and its previous values. It is useful for seeing how the data is structured in terms of autocorrelation and for finding the relative positions of AR and MA factors in time series models.

Figure 8 and Figure 9 shows the ACF of the stock data that will be use in this research. Both figures simply show indicates that the two values are perfectly correlated. The time series is most correlated with itself at the current time and at the previous time interval.

A graph showing a graph of a graph

Description automatically generated

Figure 8 - ACF for GSK.

A graph showing a graph of a graph

Description automatically generated

Figure 9 - ACF for ULVR.

**Partial Auto-Correlation Function (PACF)**

The partial autocorrelation function (PACF) graphic depicts the relationship between a time series and its lag values without accounting for intermediate delays. It aids in the estimation of AR terms in time series models by revealing the direct relationship between a data point and the data point before it. The PACF, in contrast to the ACF, analyzes the strength of the association between a particular data point and its immediate antecedents while ignoring the impact of any intermediate lags. The two stock data sets used for this research are depicted graphically in Figures 10 and 11. After accounting for the effect of the intervening delays, the graph shows that the time series is most closely related with itself at the present and prior time intervals. The link between the time series and itself weakens as latency grows, as demonstrated by the decreasing partial autocorrelation coefficient.

A graph showing a line graph

Description automatically generated

Figure 10 - PACF for GSK.

A graph with a line going up

Description automatically generated

Figure 11 - PACF for ULVR.

## 3.6 Modelling

1. **Autoregressive Integrated Moving Average (ARIMA)**

ARIMA is a popular and effective method for time series analysis and forecasting. ARIMA combines three fundamental components to model and predict future values of a time series: autoregression (AR), differencing (I), and moving average (MA) (Gujarati D.N, 2022).

The ARIMA time series model is a derivation of the autoregressive moving average (ARMA) model. ARIMA (p, d, q) is an expression of its fundamental techniques, which consist of modelling, evaluation, verification, and control. The fundamental premise of the concept is to view the ordered progression of time as a random sequence and then to use mathematics to describe that sequence. Time series values can be used to anticipate future values. The model's incorporation of autoregression (AR), differencing (I), and moving average (MA) makes accurate forecasts for future time points possible (Gujarati D.N, 2022).

Autoregression (AR): The present value of the time series and its previous values (lags) are captured by the autoregressive component (AR). Assuming that past values immediately impact the current value, ARIMA uses lagged values to make predictions about the future. In ARIMA, the number of lagged values utilized for prediction is equal to the order of the autoregressive component, indicated by "p". The notation AR (p) indicates an autoregressive model of order p. The AR (p) model is defined as:

(**Equation 3** – **AR Model)**

Moving Average (MA): The MA represents a model of the connection between the current time series value and the residuals from previous predictions. Assuming that past errors have an immediate impact on the current value, MA uses the lagged residuals to make predictions. When using ARIMA for forecasting, the "q" parameter specifies the order of the moving average component, or the number of lags in the residuals. The Moving Average (MA) can be represented as:

**(Equation 4** – **MA Model)**

Differencing (I): The integrated part (I) employs a differentiation technique to achieve stationarity in the time series data. Stationarity is essential for time series modeling because many models rely on the fact that the data maintains the same mean and variance over time. By utilizing differencing, ARIMA can transform a previously patterned and seasonal time series into a stationary time series. The "d" parameter in ARIMA specifies the differencing order required to achieve stationarity. A non-seasonal ARIMA model is created by combining differencing with autoregression and a moving average model. The ARIMA model can be represented as:

**(Equation 5** – **The** **ARIMA Model)**

where is the differenced series, an ARIMA model is represented as ARIMA(p,d,q).

*p*: order of the autoregressive part

*d*: degree of first differencing involved.

*q*: order of the moving average part

The ARIMA model is a powerful tool for time series forecasting, particularly when dealing with data that exhibits trends and seasonality. Also, the model is suitable for univariate time series data, where a single variable is observed over a sequence of time intervals. Furthermore, the ARIMA models have proven to be reliable, especially for short-term forecasts.

1. **Prophet**

The Prophet model was introduced in 2018 by Facebook (S. J. Taylor & Letham, 2018). S.J. Taylor and Letham originally created the model for forecasting daily data with weekly and yearly seasonality. The model also incorporates holiday effects. The model was later extended to cover more types of seasonal data. The model can best be represented as

**y(t) = g(t) + s(t) + h(t) + (Equation 6** – **The Prophet Model)**,

where g(t) describes a linear trend (or growth term), s(t) describes the various seasonal patterns, h(t) captures the holiday effects, and is a white noise error term. The prophet allows for multivariant analysis and defines holidays.

Furthermore, the model is a useful tool for forecasting time series, particularly for users with limited statistical knowledge. It provides a straightforward method for documenting trends, seasonality, and holiday effects, making it applicable to a vast array of applications. The time series data is predicted using an additive model, which breaks the data down into multiple parts before recombining them into a whole. Also, Prophet is robust in handling missing data, shifts in trend, and outliers, thus making it a popular time series prediction model.

1. **Random Forest (RF)**

RF is widely used and highly effective in ML for a wide range of problems, including classification and regression. To put it simply, RF is a bagging ensemble model. The model represents a substantial modification of the bagging technique, whereby an extensive collection of decorrelated trees is constructed and subsequently averaged (Breiman L., 2001).

If the predictions of the models in an ensemble are uncorrelated or just weakly correlated, the ensemble's overall performance will increase. The RF classifier alters the learning process so that the sub-models are integrated in such a way that predictions are only weakly correlated across models. The hyperparameters that are optimized for RF using cross validation (cv) are n\_jobs (the number of jobs to be executed for fit and forecast), min\_samples\_leaf (the minimum number of examples needed to be at a leaf node), n\_estimators (the number of trees in the forest), random\_state (seed used by random number generator), criterion (function for measuring split's quality), and min\_samples\_split (minimum number of instances needed for splitting an internal node)(Breiman L., 2001).

1. **Extreme Gradient Boosting (XGBOOST)**

In ML, XGBoost is a powerful gradient boosting technique that combines the predictions of relatively weak learners, such as decision trees. It takes care of missing values automatically, allows for user-defined goal functions, and features Lasso 1 and Lasso 2 regularization to prevent overfitting. Cross-validation and early halting are two performance evaluation and optimization tools that XGBoost provides. Because of its parallel and distributed computing capabilities, it can quickly and easily process massive datasets. The model produces feature importance ratings, which help to comprehend the role that features play in making predictions.In this effort, we optimize the n\_estimators hyper-parameter using the sample data. It's the required total of boost cycles. Due to its resistance to over-fitting, XGBOOST might potentially benefit from big numbers (Chen and Guestrin, 2016). The formula for the XGBOOST is given as follow.

Predicted Value = B0+ ∑M m=1fm(x)

**(Equation 7** – The **XGBOOST Model)**

Where:

B0 represent the initial prediction (base score)

M represent the total number of boosting rounds

fm(x) represent the prediction from the mth tree (weak learner) at input x.

Each weak learner fm(x) is a decision tree with shallow depth, and their predictions are combined to improve the overall predictive power of the model.

## 3.7 Evaluation

Model evaluation is critical in assessing the performance and effectiveness of the developed forecasting models for stock price prediction. In this section, we will delve into the process of evaluating the ARIMA, Prophet, RF, and XGBoost models using appropriate evaluation metrics.To objectively compare the performance of the different models, some of the following evaluation metrics will be used:

1. **Mean Square Error (MSE)**

Mean Square Error (MSE) expression is as follows:

MSE =  (Equation 8)

Where  is the observed value.

 is predicted value.

The relationship between the Mean Squared Error (MSE) and model accuracy is inverse. A higher MSE corresponds to a poorer predictive performance of the model (Chicco, Warrens and Jurman, 2021; Glen, 2022).

1. **Mean Absolute Error (MAE)**

Mean Absolute Error (MAE) is the most basic evaluation method. MAE is calculated by taking the average of the absolute differences between each predicted value and its corresponding actual value (Chicco, Warrens and Jurman, 2021). The formula for calculating MAE is as follows:

MAE =  (Equation 9)

Where:

n is the number of observations.

yi = the actual value of the ith observation.

^yi = the predicted value of the ith observation.

1. **Root Mean Square Error (RMSE)**

The gap between the measured and actual values can be approximated by calculating the Root Mean Square Error. Lower RMSE indicates better predictive accuracy (Chicco, Warrens and Jurman, 2021; Glen, 2022). The expression is as follows:

**RMSE = A black square with black letters and numbers

Description automatically generated (Equation 10)**

Where:

n is the number of observations.

Yi represents the actual value of the ith observation.

^Yi represents the predicted value of the ith observation

## 3.8 Deployment

This is the process of applying the knowledge from the preceding stages to improve decision-making and the client experience. Forecasting models are developed to forecast the movement of the two stocks. The developed prediction models can be used to forecast the movement of each stock.

## 3.9 Summary

CRISP-DM, the methodology used in this work, was discussed here. Business understanding, data understanding, data preparation, modelling, evaluation, and deployment are all part of the process. Each stage of this procedure was thoroughly described in relation to the research issue. The nature of stock data was examined in business understanding, as well as how this research will benefit stakeholders. Data understanding is concerned with the acquisition and description of data, where the data was acquired using the Python module "yfinance" and contains seven features. Data Preparation was also discussed, which includes data selection, cleaning, exploration, and preprocessing. MSE will be used to verify the accuracy and success of the model were discussed during the evaluation stage.

# | Research Findings

## 4.1 Introduction

The goal of stock market prediction is to forecast the movement of a company's stock price trend. Investing in stocks whose prices are predicted to rise and selling those whose values are predicted to fall is the basic logic behind stock market forecasting. This chapter will focus on an overview of the evaluation methodology and hyperparameter tuning of the used algorithms. Also, the chapter will look at the prediction models and how the champion model was chosen based on the performance metrics.

## 4.2 Evaluation Methodology

### 4.2.1 ARIMA

The Python module Pyramid ARIMA (Pmdarima) offers a simple user interface for automating the process of determining the appropriate ARIMA model for time series data. It is based on the statsmodels library and provides further functions for automatically determining the best arrangement of ARIMA components, such as moving average (q), order of differencing (d), and autoregression (p).

Furthermore, the function "ndiffs" in pmdarima stands for the number of differences required for data to be stationary. It aids in figuring out the bare minimum of differences necessary for a given time series to become stationary. The "order of differencing" (d) required by the ARIMA model is represented by the value produced by ‘ndiffs’. Also, another essential part of ARIMA is this order (d), which shows how many non-seasonal differences were applied to the time series. For this research, two tests were used to find the order of differencing (Augmented Dickey-Fuller and Kwiatkowski-Phillips-Schmidt-Shin), and the maximum value was chosen.

In addition, the Python pmdarima package has an auto\_arima function that performs an automated ARIMA model. By performing a stepwise search over different combinations of ARIMA hyperparameters, this function automatically selects the best ARIMA model for a given univariate time series. Hence, it determines the best values for the ARIMA hyperparameters p (order of autoregression), d (order of differencing), and q (order of moving average).

#### 4.2.2 Prophet

Prophet is particularly beneficial for evaluating data with daily, weekly, or yearly patterns since it is tailored to deal with time series forecasting problems with high seasonality and various seasonal components (Taylor and Letham, 2017). For modelling and predicting time series data, Prophet offers a straightforward and user-friendly interface. The Prophet library is used to execute these tasks efficiently. The Prophet library takes advantage of the Stan modelling language and its accompanying probabilistic programming libraries. Stan is a trained programming language for Bayesian inference and statistical modelling. With it, users can declaratively define complex statistical models by detailing the model's structure, priors, and likelihoods. It only accepts time series data in a specific format with two columns: datetime (‘ds’) and target variable (‘y’).

##### **4.2.3 ML models**

ML models were also used for comparison with the statistical model. The data was split into training and testing while using cross-validation to assess the models' performance. Nevertheless, an essential part of the ML process is hyperparameter tuning, which involves experimenting with different values for each hyperparameter until the model achieves the desired results. The objective is to determine which values of the hyperparameters produce the highest predicted performance on data that has not yet been seen. RandomizedSearchCV was effective for hyperparameter tuning to improve performance in vast search spaces.

## 4.3 Experimental Results of Proposed Models

### 4.3.1 ARIMA Model Prediction

The ARIMA model went through some pre-processing and implementing to achieve its performance. Firstly, the data was split into training and testing. This data was spread from January 1, 2021, to July 19, 2023, for modelling, hence getting rid of any data that might cause anomalies. Secondly, 40 (forty) days of adjusted closing price data was used for testing. Thirdly, using the auto Arima class in the pmdarima library to choose the best p, d and q hyperparameters. Lastly, a forty-day forecast plot comparing trends with the actual data was well visualized and performance was measured using the MSE as seen in table 4.1.

#### 4.3.1.1 ARIMA on Unilever Plc (ULVR)

The hyperparameter tuning Pmdarima was used to get the ARIMA parameters with p, d, and q as 1, 1, 1, in 11.8434 seconds. Figure 12 shows how the ULVR.L data learned over time. The line chart shows adjusted stock price trends over time. The orange trend line indicates the forecast, while the blue trend line shows the actual trends over time. As seen in this plot, the actual and the forecasted struggled to move in the same direction for the forty-day forecast of the adjusted stock closing price; hence, from this visualization, it could be easily seen that the ARIMA model wasn't a good model as the actual trend was downward sloping and the forecasted was stationary over time looking at the forty days forecast. Nonetheless, The MSE for this model falls at 24765.90, as seen in table 2. This shows that the Arima model on Unilever data is underfitting. In other words, the model has not learned enough from the data and performs poorly on both the training set and the unseen data (test set).

Figure 12 shows the graphical representation of how ARIMA performs on the Unilever dataset.

A graph with blue and orange lines

Description automatically generated

Figure 12 - Actual vs Forecasted (ARIMA) for ULVR.L.

## 4.3.1.2 ARIMA on GlaxoSmithKline (GSK)

Figure 13 shows how the GSK.L dataset learned over time. The line chart that shows adjusted stock price trends over time. The orange trend line indicates the forecast, while the blue trend line shows the actual trends over time. As seen in this plot, just like the ULVR.L in Figure 12, the actual and the forecasted also struggled to move in the same direction for the forty-day forecast; hence, from this plot, it could be easily seen that the ARIMA model wasn't a good model as the actual trend was downward sloping and the forecasted adjusted closing price was moving straight looking at the forty days prediction. Nevertheless, The MSE value for this model is seen as 2562.73 in table 2. The run time for this model was approximately 5.53 seconds.

A graph with orange and blue lines

Description automatically generated

Figure 13 - Actual vs Forecast (ARIMA) for GSK.L.

### 4.3.2 Prophet model prediction

Prophet provides a simple and user-friendly interface for modelling and forecasting time series data, making it accessible to users to be able to use the model for prediction. The Prophet model went through some pre-processing and implementing to achieve its performance. Firstly, the data was split into training and testing. This data was spread from January 1st, 2021, to July 19th, 2023 for modelling, hence getting rid of any data that might cause anomalies. Secondly, 40 (forty) days of data was used for testing. Lastly, a forty-day forecast plot comparing the adjusted closing price trends with the actual data was well visualised, and performance was measured using the MSE as seen in table 2.

1. **Prophet on Unilever PLC (ULVR.L)**

The Prophet model had a run time of about 0.90 seconds and an MSE value of 4043.38 on the test data. Fig. 4.3 shows how the ULVR.L data learned over time. The line chart shows how the adjusted stock price trends over time. The orange trend line indicates the forecast, while the blue trend line shows the actual trends over time. As seen in this plot, the actual and the forecasted struggled to move in the same direction for the forty day forecast; hence, from this visualisation it could be easily seen that the Prophet model wasn't a good model as the actual trend was downward sloping and the forecasted was upward sloping for the forty days forecast of the adjusted closing stock price.

A graph with blue and orange lines

Description automatically generated

Figure 14 - Prophet Actual vs Forecast For ULVR.L.

1. **Prophet on GSK.L**

Looking at the Prophet model on GSK.L adjusted closing price data in Figure 15. The run time for fitting the model took approximately 0.8289 seconds. The model had an MSE of 6996.65 on the test dataset, which makes the model overfitted. Furthermore, the line chart shows adjusted stock price trends over time. The orange trend line indicates the forecast, while the blue trend line shows the actual trends. As seen in this plot, the actual and forecasted adjusted closing price moved in opposite directions for the forty-day forecast. Hence, looking at this visualization, it could be easily seen that the Prophet model wasn't a good model as the actual trend was downward sloping and the forecasted trend was upward sloping, taking note of the forty days forecast of the adjusted closing stock price of GSK stock.

A graph of a graph showing the price of a stock market

Description automatically generated

Figure 15 - Prophet Actual vs Forecast for GSK.L.

## 4.3.3 RF Model Prediction

RF model was essential for this research, this model firstly passed through some preprocessing where 40 days of data was held for testing. Secondly, using Randomised search cross validation, the best hyperparameter tuning was selected consisting of n\_estimator and max\_depth. Thirdly, a 40-day test data was used for forecasting and lastly, the MSE was generated to know the performance of the model.

1. **RF Model on Unilever PLC (ULVR)**

The best parameters for the model on ULVR.L were 500 for the n\_estimators and 4 for the max\_depth. The MSE value of 1204.51 was obtained for the test data. As seen in Figure 16, the line chart shows adjusted stock price trends over time. The orange trend line indicates the forecast, while the blue trend line shows the actual trends. As seen in this plot, the actual and the forecasted adjusted closing price were intertwined for the forty-day forecast, unlike the statistical models that went in the opposite direction. The model was a good one as the actual trend and forecasted trend are intertwined, taking note of the forty days forecast of the adjusted closing stock price of ULVR stock.

A graph with blue and orange lines

Description automatically generated

Figure 16 - RF Actual vs Forecast on ULVR.L.

1. **RF Model On GSK.L**

Figure 17 below shows how the RF adjusted stock price learned over time with the GSK.L data. This line chart below shows the adjusted stock price trends. The orange trend line indicates the forecast, while the blue trend line shows the actual trends over time. As seen in this plot, the actual and the forecasted trend move in the same direction for the forty-day forecast. Hence, from this visualisation it could be easily seen that the RF model was a good model as the actual trend and the forecasted trend is intertwined for the forty days forecast of the adjusted closing stock price. The best parameters for the model were 100 for the n\_estimators and 4 for the max\_depth. The MSE of the model was 297.38. Nonetheless, looking at the forecast and the MSE, it is no surprise while the RF emerged as the champion model for predicting the GSK.L adjusted closing price.

A graph showing the growth of the stock market

Description automatically generated

Figure 17 - RF Actual vs Forecast on GSK.L data.

### 4.3.4 XBBOOST Model Prediction

The XGBOOST was another essential model for this research. The XGBOOST used randomized cross-validation for the selection of the best parameters for the model. This model passed through steps before achieving the performance of the model. Firstly, the last 40 days of data was kept for testing. Secondly, randomized search cross-validation was used to find the best hyperparameter tuning. The hyperparameters considered are, n\_estimators, max\_depth, and learning rate. lastly, a 40-day test data was used for forecasting and lastly, the MSE was generated to know the performance of the model.

1. **XGBOOST Model on Unilever PLC (ULVR)**

Figure 18 below shows how the XGBOOST adjusted stock price learned over time with the ULVR.L data. This line chart below shows the adjusted stock price trends . The orange trend line indicates the forecast, while the blue trend line shows the actual trends over time. As seen in this plot, the actual and the forecasted trend move in the same direction for the forty-day forecast. Hence, from this plot it could be easily seen that the XGBOOST model was a good model as the actual trend and the forecasted trend is intertwined for the forty days forecast of the adjusted closing stock price. The best parameters selected by randomized cross validation are 300, 2, and 0.1 for n\_estimators, max\_depth, and learning depth, respectively. The MSE of the model was 1060.93. Nevertheless, looking at the forecast and the MSE, it is no surprise while XGBOOST emerged as the champion model for predicting the ULVR.L adjusted closing price.

A graph with blue and orange lines

Description automatically generated

Figure 18 - XGBoost Actual vs forecast on ULVR.L.

1. **XG model on GlaxoSmithKline (GSK)**

Figure 19 below shows how the XGBOOST adjusted stock price learned over time with the GSK.L dataset. This line chart below shows the adjusted stock price trends. The orange trend line indicates the forecast, while the blue trend line shows the actual trends over time. As seen in this plot, the actual and the forecasted trend move in the same direction for the forty-day forecast. However, from this plot it could be easily seen that the XGBOOST model was a fairly good model as the actual trend and the forecasted trend somewhat closely moved in the same direction for the forty days forecast of the adjusted closing stock price. Nevertheless, though the actual and forecast trend moved in the same direction, we could closely see both trends struggled to keep the movement in the same direction. The best parameters selected by randomized cross-validation are 300, 2 and 0.1 for n\_estimators, max\_depth, and learning depth, respectively. The MSE of the model on held out data was 356.61.

A graph with blue and orange lines

Description automatically generated

Figure 19 - XGBoost Actual vs forecast of GSK.L data.

## 4.4 Model Comparison and Discussion

Table 2 shows the overview of this project so far. A comparison of two statistical and two ML models were used to train and test the two stock data sets that were the case studies.

Furthermore, as seen in Table 2 Arima had 24765.90 and 2562.73 MSE values for ULVR.L and GSK.L, respectively, which makes it a better model than the prophet model, which has MSE values of 113664.69 and 6996.65 for ULVR.L and GSK.L, respectively.Nonetheless, For the ML models, RF had a MSE value of 1204.51 and 297.38 for ULVR.L and GSK.L data, respectively. While for the XGBOOST model, the MSE for ULVR.L is 1060.93 and the MSE for GSK.L is 356.61.

Table 2. also shows the run time of each model on both ULVR.L and GSK.L data. The Prophet model had the fastest run time compared to other models implemented.

Lastly as seen in Table 2 The MSE gives a clear picture of the performance of the various models. Nevertheless, RF emerged as the champion model for forecasting the adjusted closing stock price for GSK.L with an MSE value of 297.38 while XGBOOST emerged as the champion model for forecasting the adjusted closing stock price for ULVR.L stocks with an MSE of 1060.93.

Table 2 shows a summary of the model's test metrics for both data sets.

Table 2 - Summary of the Models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Data** | **MSE** | **RunTime(s)** |
| **ARIMA** | ULVR.L | 24765.90 | 11.8434 |
| GSK.L | 2562.73 | 5.534 |
| **PROPHET** | ULVR.L | 113664.69 | 0.90304 |
| GSK.L | 6996.65 | 0.8289 |
| **RANDOM FOREST** | ULVR.L | 1204.51 | 29.4012 |
| GSK.L | 297.38 | 36.3898 |
| **XGBOOST** | ULVR.L | 1060.93 | 4.8007 |
| GSK.L | 356.61 | 5.7263 |

## 4.5 Summary

The chapter presents the outcomes and discussions related to stock market prediction. The primary objective of stock market prediction is to forecast the movement of stock prices or trends. Success in this endeavour offers significant rewards, as it aids in strategic investment decisions based on predicted price changes. The chapter starts by discussing the evaluation methodology's overview and the optimization of hyperparameters for the employed algorithms.

The proposed prediction models are then subjected to testing, and the results are thoroughly examined and discussed. The latter part of the chapter encapsulates the key takeaways from the analysis conducted. Within the evaluation methodology, ARIMA is introduced as a time series modeling technique with the help of the Pmdarima Python module. This module automates the selection of appropriate ARIMA components such as moving average, differencing order, and autoregression. The 'ndiffs' function assists in determining the minimum differences required for data stationarity, vital for ARIMA modeling. The ARIMA order of differencing is pivotal, indicating non-seasonal differences applied to the time series. Moreover, the Python 'pmdarima' package features an 'auto\_arima' function that automates ARIMA model selection. It accomplishes this by iteratively searching different ARIMA hyperparameter combinations, selecting the best model for a given time series.

The Prophet model emerges as a beneficial tool for handling time series data with daily, weekly, or yearly patterns, excelling in scenarios with high seasonality. ML models are introduced, highlighting the necessity of hyperparameter tuning via techniques such as RandomizedSearchCV. Balanced fit models, which capture underlying patterns without fitting noise, are discussed as optimal for generalization. Experimental results of the proposed models are unveiled, showing the ARIMA model's challenges with underfitting on Unilever PLC and GlaxoSmithKline data. Prophet's overfitting tendencies are highlighted, with high MSE on test data. The RF and XGBOOST models emerge as our champion models for GSK and ULVR respectively, given their MSE values.

# | Conclusions and Recommendations

## 5.1 Introduction

This chapter serves as the concluding segment of this thesis. The initial section expounds upon the methodologies employed in the study and presents the outcomes derived from the conducted tests. The thesis made a comparison between two ML algorithms and statistical methods to forecast stock prices, and using MSE as the evaluation metrics, the champion model was obtained.

Furthermore, we were able to see how each model fared in stock forecasting and the run time it took for each model to run. Consequently, the constraints encountered during the investigations are underscored, and certain plausible adjustments are proposed. Building upon this foundation, potential avenues for addressing these challenges in the future are put forth. Finally, the chapter culminates by providing a summary of its contents, summarizing the entire thesis report, and concluding with reflective insights concerning the project.

## 5.1.1 Limitations and Future Works

One of the limitations of this project is the quality of the data and the chosen time frame. The disparity in the bid prices of stocks between 1988 and 2023 had an influence on our model, thus making it underfit or overfit in some cases. This was addressed by reducing the time period for the modelling. Nevertheless, having a larger data set would better enhance our forecasting models.

Furthermore, the aims and objectives of the research were achieved, given the extent of the study. Future research opportunities to improve the forecasting models are identified. This may involve exploring ensemble approaches that combine the strengths of different methods or developing hybrid models’ approaches such as Convolutional Neural Networks (CNN) and other deep learning models can be used. Such models might be used for future works to broaden the comparisons. Also, only univariate data was evaluated in this study. Many studies have shown that having more data sources can significantly improve prediction accuracy. Finally, future work should incorporate external factors like macroeconomic indicators, news sentiment, and market sentiment to enhance the forecasting models.

## 5.1.2 Summary

This study compares the performance of two time-series analysis approaches and two ML techniques in predicting adjusted stock prices for two stocks listed on the LSE. Nevertheless, the thesis addressed the problem of insufficient data by using real time data from LSE. To begin, historical price data on these stocks was scrapped from the Yahoo Finance platform. After data exploration was carried out, the data was preprocessed, and then the models which are the ARIMA,Prophet, RF, and XGBOOST, became appropriate to fit and predict the future trend in these stock data. Nonetheless, the measures of model performance were compared, RF and XGBOOST performed better. RF emerged as the champion model for the GSK.L stock with an MSE of 297.38, while XGBOOST was the champion model for the ULVR.L stock with an MSE value of 1060.93.

# Appendix

from time import time

from pprint import pprint

import numpy as np

import pandas as pd

import yfinance as yf

import matplotlib.pyplot as plt

import seaborn as sns

from pandas.plotting import register\_matplotlib\_converters

from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

from statsmodels.tsa.stattools import adfuller

from statsmodels.tsa.seasonal import STL

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import RandomizedSearchCV

from sklearn.metrics import mean\_squared\_error as MSE

from xgboost import XGBRegressor

import pmdarima as pm

from pmdarima.arima import ndiffs

from pmdarima import auto\_arima

from statsmodels.tsa.arima.model import ARIMA

from prophet import Prophet

register\_matplotlib\_converters()

sns.set\_context("paper", font\_scale=1.4)

sns.set\_style("whitegrid")

plt.rc("figure", figsize=(12, 8))

plt.rc("axes.spines", top=False, right=False, left=False)

if "unilever" not in locals() or "gsk" not in locals():

unilever = yf.download("ULVR.L", start="2000-01-01", end="2023-07-20")

gsk = yf.download("GSK.L", end="2023-07-20")

# selecting only the adjusted close price

unilever\_cl = unilever["Adj Close"]

gsk\_cl = gsk["Adj Close"]

# use businessday frequency

gsk\_cl = gsk\_cl.asfreq("B").ffill()

unilever\_cl = unilever\_cl.asfreq("B").ffill()

def make\_title(data, start=None):

if start is None:

title = f"{str(data.index.min().date())} to {str(data.index.max().date())}"

else:

title = (

f"{start}: {str(data.index.min().date())} to {str(data.index.max().date())}"

)

return title

gsk\_cl.plot(title= make\_title(gsk\_cl, "GSK.L"))

make\_title(gsk\_cl)

unilever\_cl.plot(title= make\_title(unilever\_cl, "ULVR.L"))

print("limit to last 200 days")

stl = STL(gsk\_cl[-200:])

res = stl.fit()

fig = res.plot();

stl = STL(unilever\_cl)

res = stl.fit()

fig = res.plot();

print("limit to last 200 days")

stl = STL(unilever\_cl[-200:])

res = stl.fit()

fig = res.plot();

adf, pval, usedlag, nobs, crit\_vals, icbest = adfuller(gsk\_cl.values)

print('GSK ADF test statistic:', adf)

print('GSK ADF p-values:', pval)

print('GSK ADF number of lags used:', usedlag)

print('GSK ADF number of observations:', nobs)

print('GSK ADF critical values:', crit\_vals)

print('GSK ADF best information criterion:', icbest)

print(f'GSK Stationary? ==> {pval < 0.05}')

adf, pval, usedlag, nobs, crit\_vals, icbest = adfuller(unilever\_cl.values)

print('Unilever ADF test statistic:', adf)

print('Unilever ADF p-values:', pval)

print('Unilever ADF number of lags used:', usedlag)

print('Unilever ADF number of observations:', nobs)

print('Unilever ADF critical values:', crit\_vals)

print('Unilever ADF best information criterion:', icbest)

print(f'Unilever Stationary? ==> {pval < 0.05}')

plot\_acf(gsk\_cl, lags = 50);

plot\_acf(unilever\_cl, lags = 50);

plot\_pacf(gsk\_cl, lags = 10);

plot\_pacf(unilever\_cl, lags = 10);

gsk\_difs = (gsk\_cl - gsk\_cl.shift(freq="B")) / gsk\_cl \* 100

gsk\_difs = gsk\_difs.dropna()

gsk\_difs.plot(title= make\_title(gsk\_difs, "% change of GSK.L"));

sns.despine(left=True)

gsk\_short = gsk\_difs.loc["2023":]

gsk\_short.plot(figsize=(12, 12), grid=True,

title= make\_title(gsk\_short, "% change of GSK.L"));

sns.despine(left=True)

unilever\_difs = (unilever\_cl - unilever\_cl.shift(freq="B")) / unilever\_cl \* 100

unilever\_difs = unilever\_difs.dropna()

unilever\_difs.plot(figsize=(12, 12),

title= make\_title(unilever\_difs, "% change of ULVR.L"));

unilever\_short = unilever\_difs.loc["2023":]

unilever\_short.plot(figsize=(12, 12), grid=True,

title= make\_title(unilever\_short, "% change of ULVR.L"));

def make\_ml\_data(data, n=30):

data\_list = []

for sh in range(1,n+1):

data\_list.append(data.shift(periods=sh, freq=freq))

new\_data = pd.concat(data\_list, axis=1)

new\_data["month"] = pd.DataFrame(data.index.month, index=data.index)

new\_data = pd.concat([new\_data, data], axis=1).dropna()

new\_data.columns = [str(i) for i in list(range(1, n+1))] + ["month", "y"]

return new\_data

# window through the data

lag = 30

freq = "B"

gsk\_ml = make\_ml\_data(gsk\_cl, n=lag)

ulvr\_ml = make\_ml\_data(unilever\_cl, n=lag)

ulvr\_ml.head(30)[-1::-1]

def model\_arima(data):

"""

Return best ARIMA model (with its parameters). Modelling is done using

ndiffs to find best d parameter

pm.auto\_arima to find best p & q parameter

"""

# get param d

kpss\_diffs = ndiffs(data, alpha=0.05, test="kpss", max\_d=6)

adf\_diffs = ndiffs(data, alpha=0.05, test="adf", max\_d=6)

n\_diffs = max(adf\_diffs, kpss\_diffs)

# find best params (p, q) and model

best\_model = pm.auto\_arima(

data,

stepwise=True,

suppress\_warnings=True,

trace=1,

random\_state=seed,

out\_of\_sample\_size=days,

)

return best\_model

def plot\_forecasts(actual, pred, forecast\_days, size=200, name=None):

"""Make timeseries plot of the actual data vs the forecasted"""

actual\_pred = pd.concat([actual, pred], axis=1)

actual\_pred.columns = ["actual", "forecast"]

data = actual\_pred.iloc[-size:].copy()

fig = sns.lineplot(data, linewidth=2)

plt.title(

f"{name} forecast: {str(data.index.min().date())} to {str(data.index.max().date())}"

)

return fig

def model\_prophet(data):

"""modelling for prophet model"""

## preprocess data

new\_data = data.reset\_index()

new\_data.columns = ["ds", "y"]

## fit model

model = Prophet()

model.fit(new\_data)

return model

def t\_split(data, size=14):

"""

Split the data with the specified test size

Return the indices of the split for RandomizedSearchCV CV parameter.

"""

split = [

(

data.iloc[:-size].reset\_index().index.values,

data.iloc[-size:].reset\_index().index.values,

)

]

return split

def get\_best\_ml\_model(model, X, y, grid, n\_iter=30):

"""

Return best ML model for the passed base model (XGB/RF)

after using RandomizedSearchCV to get best hyperparameter

"""

best\_ml\_model = RandomizedSearchCV(

estimator=model,

param\_distributions=grid,

n\_iter=n\_iter,

cv=t\_split(X, days),

verbose=1,

random\_state=0,

n\_jobs=-1,

scoring="neg\_mean\_squared\_error",

return\_train\_score=True,

)

# Fit the random search model

best\_ml\_model.fit(X.values, y.values)

return best\_ml\_model

def plot\_all\_forecats(data, title, size= 150):

"""Plots all given timeseries data"""

fig = sns.lineplot(data[-size:], dashes=False)

title = make\_title(data, title)

plt.title(title)

return fig

seed = 0

results = {

"ARIMA": {

"GSK": {

"test": None,

},

"ULVR": {

"test": None,

},

},

"prophet": {

"GSK": {

"test": None,

},

"ULVR": {

"test": None,

},

},

"RF": {

"GSK": {

"test": None,

},

"ULVR": {

"test": None,

},

},

"XGB": {

"GSK": {

"test": None,

},

"ULVR": {

"test": None,

},

},

} # to hold the error metric (MSE) of each model

params = {

"ARIMA": {"GSK": None, "ULVR": None},

"prophet": {"GSK": None, "ULVR": None},

"RF": {"GSK": None, "ULVR": None},

"XGB": {"GSK": None, "ULVR": None},

} # to hold each model final parameters

forecasts = {

"GSK": {},

"ULVR": {},

} # to hold the forecasted data of the final models

timings = {

"ARIMA": {"GSK": None, "ULVR": None},

"prophet": {"GSK": None, "ULVR": None},

"RF": {"GSK": None, "ULVR": None},

"XGB": {"GSK": None, "ULVR": None},

} # to hold the models' runtime

plots = {

"ARIMA": {"GSK": None, "ULVR": None},

"prophet": {"GSK": None, "ULVR": None},

"RF": {"GSK": None, "ULVR": None},

"XGB": {"GSK": None, "ULVR": None},

} #to hold the models' plots

days = 40

train\_gsk, test\_gsk = gsk\_cl.iloc[:-days], gsk\_cl.iloc[-days:]

train\_gsk, test\_gsk = gsk\_cl.iloc[:-days], gsk\_cl.iloc[-days:]

train\_ulvr, test\_ulvr = unilever\_cl.iloc[:-days], unilever\_cl.iloc[-days:]

train\_ulvr, test\_ulvr = unilever\_cl.iloc[:-days], unilever\_cl.iloc[-days:]

start\_time = time()

gsk\_arima = model\_arima(train\_gsk)

gsk\_pred\_arima = gsk\_arima.predict(days)

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["ARIMA"]["GSK"] = end\_time

test\_score = MSE(test\_gsk, gsk\_pred\_arima)

results["ARIMA"]["GSK"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

params["ARIMA"]["GSK"] = gsk\_arima.order

print("MSE values")

pprint(results["ARIMA"]["GSK"])

print("\n\nfinal hyperparameter")

pprint(params["ARIMA"]["GSK"])

in\_sample = pd.DataFrame(gsk\_arima.predict\_in\_sample(), index=train\_gsk.index)

pred = pd.DataFrame(gsk\_pred\_arima, index=test\_gsk.index)

tot\_forecast = pd.concat([in\_sample, pred])

plots["ARIMA"]["GSK"] = plot\_forecasts(gsk\_cl, tot\_forecast, days, name="GSK.L (ARIMA)")

forecasts["GSK"]["ARIMA"] = tot\_forecast[0]

start\_time = time()

ulvr\_arima = model\_arima(train\_ulvr)

## predict

ulvr\_pred\_arima = ulvr\_arima.predict(days)

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["ARIMA"]["ULVR"] = end\_time

test\_score = MSE(test\_ulvr, ulvr\_pred\_arima)

results["ARIMA"]["ULVR"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

params["ARIMA"]["ULVR"] = ulvr\_arima.order

print("MSE values")

pprint(results["ARIMA"]["ULVR"])

print("\n\nfinal hyperparameter")

pprint(params["ARIMA"]["ULVR"])

in\_sample = pd.DataFrame(ulvr\_arima.predict\_in\_sample(), index = train\_ulvr.index)

pred = pd.DataFrame(ulvr\_pred\_arima, index = test\_ulvr.index)

tot\_forecast = pd.concat([in\_sample, pred])

plots["ARIMA"]["ULVR"] = plot\_forecasts(unilever\_cl, tot\_forecast, days, name = "ULVR.L (ARIMA)")

forecasts["ULVR"]["ARIMA"] = tot\_forecast[0]

start\_time = time()

## fit model

gsk\_prophet = model\_prophet(train\_gsk)

## make prediction

future = gsk\_prophet.make\_future\_dataframe(periods=days)

forecasts\_ = gsk\_prophet.predict(future)

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["prophet"]["GSK"] = end\_time

## test prediction

test\_score = MSE(

test\_gsk, pd.DataFrame(forecasts\_["yhat"][-days:].values, index=test\_gsk.index)

)

results["prophet"]["GSK"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

print("MSE values")

pprint(results["prophet"]["GSK"])

forecasts\_.index = gsk\_cl.index

forecasts\_ = forecasts\_[["yhat"]]

plots["prophet"]["GSK"] = plot\_forecasts(gsk\_cl, forecasts\_, days, name = "GSK.L (prophet)")

forecasts["GSK"]["prophet"] = forecasts\_["yhat"]

start\_time = time()

## fit model

ulvr\_prophet = model\_prophet(train\_ulvr)

## make prediction

future = ulvr\_prophet.make\_future\_dataframe(periods=days)

forecasts\_ = ulvr\_prophet.predict(future)

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["prophet"]["ULVR"] = end\_time

## test prediction

test\_score = MSE(

test\_ulvr, pd.DataFrame(forecasts\_["yhat"][-days:].values, index=test\_ulvr.index)

)

results["prophet"]["ULVR"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

print("MSE values")

print(results["prophet"]["ULVR"])

forecasts\_.index = unilever\_cl.index

forecasts\_ = forecasts\_[["yhat"]]

plots["prophet"]["ULVR"] = plot\_forecasts(unilever\_cl, forecasts\_, days, name = "ULVR.L (prophet)")

forecasts["ULVR"]["prophet"] = forecasts\_["yhat"]

days = 40

# cut out n samples for validation (holdout)

train\_gsk, hold\_out\_gsk = gsk\_ml.iloc[:-days], gsk\_ml.iloc[-days:]

train\_ulvr, hold\_out\_ulvr = ulvr\_ml.iloc[:-days], ulvr\_ml.iloc[-days:]

# Split remaining to X, y

X\_gsk\_ml, y\_gsk\_ml = train\_gsk.iloc[:, :-1], train\_gsk.iloc[:, -1]

X\_ulvr\_ml, y\_ulvr\_ml = train\_ulvr.iloc[:, :-1], train\_ulvr.iloc[:, -1]

# Split holdout to X, y

X\_hold\_out\_gsk, y\_hold\_out\_gsk = hold\_out\_gsk.iloc[:, :-1], hold\_out\_gsk.iloc[:, -1]

X\_hold\_out\_ulvr, y\_hold\_out\_ulvr = hold\_out\_ulvr.iloc[:, :-1], hold\_out\_ulvr.iloc[:, -1]

n\_estimators = [int(x) for x in np.linspace(start=100, stop=1000, num=10)]

# Number of features to consider at every split

max\_depth = [1, 2, 3, 4]

# Create the random grid

random\_grid = {

"n\_estimators": n\_estimators,

"max\_depth": max\_depth,

}

[int(x) for x in np.linspace(3, 13, num=7)] + [None]

# Create the random grid

random\_grid = {

"n\_estimators": n\_estimators,

"max\_depth": max\_depth,

}

print("grid")

pprint(random\_grid)

start\_time = time()

rf = RandomForestRegressor(

criterion="squared\_error", random\_state=seed

) # base model to tune

rf\_random = get\_best\_ml\_model(rf, X\_gsk\_ml, y\_gsk\_ml, random\_grid, 20)

print("best\_param", rf\_random.best\_params\_)

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["RF"]["GSK"] = end\_time

best\_random = rf\_random.best\_estimator\_

test\_score = MSE(y\_hold\_out\_gsk.values, best\_random.predict(X\_hold\_out\_gsk.values))

results["RF"]["GSK"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

params["RF"]["GSK"] = rf\_random.best\_params\_

print("MSE values")

pprint(results["RF"]["GSK"])

print("\n\n\nfinal hyperparamter")

pprint(params["RF"]["GSK"])

tot\_forecast = pd.DataFrame(

best\_random.predict(gsk\_ml.iloc[:, :-1]), index=gsk\_ml.index

)

plots["RF"]["GSK"] = plot\_forecasts(gsk\_ml.iloc[:, -1], tot\_forecast, days, name="GSK.L (RF)")

forecasts["GSK"]["RF"] = tot\_forecast[0]

# Number of trees in random forest

n\_estimators = [int(x) for x in np.linspace(start=100, stop=1000, num=10)]

# Number of features to consider at every split

max\_depth = [1, 2, 3, 4]

# Create the random grid

random\_grid = {

"n\_estimators": n\_estimators,

"max\_depth": max\_depth,

}

print("grid")

pprint(random\_grid)

start\_time = time()

rf = RandomForestRegressor(

criterion="squared\_error", random\_state=seed

) # base model to tune

rf\_random = get\_best\_ml\_model(rf, X\_ulvr\_ml, y\_ulvr\_ml, random\_grid, 20)

print("best\_param", rf\_random.best\_params\_)

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["RF"]["ULVR"] = end\_time

best\_random = rf\_random.best\_estimator\_

test\_score = MSE(y\_hold\_out\_ulvr.values, best\_random.predict(X\_hold\_out\_ulvr.values))

results["RF"]["ULVR"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

params["RF"]["ULVR"] = rf\_random.best\_params\_

print("MSE values")

pprint(results["RF"]["ULVR"])

print("\n\n\nfinal hyperparamter")

pprint(params["RF"]["ULVR"])

tot\_forecast = pd.DataFrame(

best\_random.predict(ulvr\_ml.iloc[:, :-1]), index=ulvr\_ml.index

)

plots["RF"]["ULVR"] = plot\_forecasts(ulvr\_ml.iloc[:, -1], tot\_forecast, days, name="UlVR.L (RF)")

forecasts["ULVR"]["RF"] = tot\_forecast[0]

n\_estimators = [int(x) for x in np.linspace(start=50, stop=500, num=10)]

# Number of features to consider at every split

max\_depth = [1, 2]

learning\_rate = [0.01, 0.1, 0.001]

# Create the random grid

random\_grid = {

"n\_estimators": n\_estimators,

"max\_depth": max\_depth,

"learning\_rate": learning\_rate,

}

print("\n\n", "grid")

print(random\_grid)

best\_random = xgb\_random.best\_estimator\_

test\_score = MSE(y\_hold\_out\_gsk.values, best\_random.predict(X\_hold\_out\_gsk.values))

results["XGB"]["GSK"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

params["XGB"]["GSK"] = xgb\_random.best\_params\_

print("MSE values")

print(results["XGB"]["GSK"])

print("\n\n\nfinal hyperparamter")

print(params["XGB"]["GSK"])

tot\_forecast = pd.DataFrame(best\_random.predict(gsk\_ml.iloc[:, :-1]), index=gsk\_ml.index)

plots["XGB"]["GSK"] = plot\_forecasts(gsk\_ml.iloc[:, -1], tot\_forecast, days, name = "GSK.L (XGB)")

fig4 = forecasts["GSK"]["XGB"] = tot\_forecast[0]

n\_estimators =[int(x) for x in np.linspace(start = 50, stop = 500, num = 10)]

# Number of features to consider at every split

max\_depth =[1, 2,]

learning\_rate = [0.01, 0.1, 0.001]

# Create the random grid

random\_grid = {'n\_estimators': n\_estimators,

'max\_depth': max\_depth,

'learning\_rate': learning\_rate

}

print("\n\ngrid")

pprint(random\_grid)

start\_time = time()

xgb = XGBRegressor()

xgb\_random = get\_best\_ml\_model(xgb, X\_ulvr\_ml, y\_ulvr\_ml, random\_grid, 20)

xgb\_random.best\_params\_

end\_time = time() - start\_time

print("\n\ntime", end\_time, "secs")

timings["XGB"]["ULVR"] = end\_time

best\_random = xgb\_random.best\_estimator\_

test\_score = MSE(y\_hold\_out\_ulvr.values, best\_random.predict(X\_hold\_out\_ulvr.values))

results["XGB"]["ULVR"]["test"] = {

"MSE": f"{test\_score:.2f}",

}

params["XGB"]["ULVR"] = xgb\_random.best\_params\_

print("MSE values")

pprint(results["XGB"]["ULVR"])

print("\n\n\nfinal hyperparamter")

pprint(params["XGB"]["ULVR"])

tot\_forecast = pd.DataFrame(best\_random.predict(ulvr\_ml.iloc[:, :-1]), index=ulvr\_ml.index)

plots["XGB"]["ULVR"] = plot\_forecasts(ulvr\_ml.iloc[:, -1], tot\_forecast, days, name = "ULVR.L (XGB)")

forecasts["ULVR"]["XGB"] = tot\_forecast[0]

df = pd.DataFrame.from\_dict(results, orient="index").stack().to\_frame()

# to break out the lists into columns

df = pd.DataFrame(df[0].values.tolist(), index=df.index)

# do it again

df = df.stack().to\_frame()

df = pd.DataFrame(df[0].values.tolist(), index=df.index).astype("float").unstack()

print("MSE VALUES")

df["MSE"][["test"]]

print("Hyperparamters")

pd.DataFrame.from\_dict(params, orient="index").stack().to\_frame().style

# References

Adebiyi, A.A., Ayo, C.K., Adebiyi, M. and Otokiti, S.O. (2012). Stock Price Prediction using Neural Network with Hybridized Market Indicators. *Journal of Emerging Trends in Computing and Information Sciences*, [online] 3(1). Available at:<http://eprints.lmu.edu.ng/3224/> [Accessed 5 Aug. 2021].

Aldhyani, T.H.H. and Alzahrani, A. (2022). Framework for Predicting and Modeling Stock Market Prices Based on Deep Learning Algorithms. *Electronics*, [online] 11(19), p.3149. doi:https://doi.org/10.3390/electronics11193149.

Almazrouee, A.I., Almeshal, A.M., Almutairi, A.S., Alenezi, M.R. and Alhajeri, S.N. (2020). Long-Term Forecasting of Electrical Loads in Kuwait Using Prophet and Holt–Winters Models. *Applied Sciences*, 10(16), p.5627. doi:<https://doi.org/10.3390/app10165627>.

Ayodele Oladipupo, T. (2010). *Types of Machine Learning Algorithms*. [online] Available at:<https://pdfs.semanticscholar.org/c4ae/802491724aee021f31f02327b9671cead3dc.pdf>.

Ayushman Durgapal and Vrince Vimal (2021). Prediction of Stock Price Using Statistical and Ensemble learning Models: A Comparative Study. doi:https://doi.org/10.1109/upcon52273.2021.9667644.

Breiman, L. (2001). Random Forests. *Machine Learning*, [online] 45(1), pp.5–32. doi:<https://doi.org/10.1023/a:1010933404324>.

Brown, D.P. and Jennings, R.H. (1989). On Technical Analysis. *The Review of Financial Studies*, [online] 2(4), pp.527–551. doi:<https://doi.org/10.1093/rfs/2.4.527>.

Caiola, G. and Reiter, J. (2010). Random Forests for Generating Partially Synthetic, Categorical Data. *TRANSACTIONS ON DATA PRIVACY*, [online] 3, pp.27–42. Available at:<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=069e26205b9dbdc82c5ab31721b0a03f325294db> [Accessed 27 Jul. 2023].

Chen, J. (2022). *Stock Market | Investopedia*. [online] Investopedia. Available at:<https://www.investopedia.com/terms/s/stockmarket.asp>.

Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, pp.785–794. doi:<https://doi.org/10.1145/2939672.2939785>.

Chen, T. and He, T. (2015). *xgboost: eXtreme Gradient Boosting*. [online] Available at:<https://cran.ms.unimelb.edu.au/web/packages/xgboost/vignettes/xgboost.pdf>.

Chicco, D., Warrens, M.J. and Jurman, G., (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, p.e623.

Citigroup, B., Suisse, C., Bank, D., Cazenove, J.M., Securities, M., Stanley, M. and Chartered, S., GlaxoSmithKline plc (2012).

de Faria, E.L., Albuquerque, M.P., Gonzalez, J.L., Cavalcante, J.T.P. and Albuquerque, M.P. (2009). Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods. *Expert Systems with Applications*, 36(10), pp.12506–12509. doi:https://doi.org/10.1016/j.eswa.2009.04.032.

Dey, S., Kumar, Y., Saha, S., & Basak, S. (2016). Forecasting to Classification: Predicting the direction of stock market price using Xtreme Gradient Boosting. *PESIT South Campus*, 1-10.

Enke, D. and Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29(4), pp.927–940. doi:<https://doi.org/10.1016/j.eswa.2005.06.024>.

Fama, E.F. and Blume, M.E. (1966). Filter Rules and Stock-Market Trading. *The Journal of Business*, 39(S1), p.226. doi:<https://doi.org/10.1086/294849>.

Fama, E.F. and French, K.R. (1988). Dividend yields and expected stock returns. *Journal of Financial Economics*, 22(1), pp.3–25. doi:<https://doi.org/10.1016/0304-405x(88)90020-7>.

Faria, G. and Verona, F. (2018). Forecasting stock market returns by summing the frequency-decomposed parts. *Journal of Empirical Finance*, 45, pp.228–242. doi:<https://doi.org/10.1016/j.jempfin.2017.11.009>.

Ferreira, M.A. and Santa-Clara, P. (2011). Forecasting stock market returns: The sum of the parts is more than the whole. *Journal of Financial Economics*, 100(3), pp.514–537. doi:<https://doi.org/10.1016/j.jfineco.2011.02.003>.

finance.yahoo.com. (n.d.). *GSK plc (GSK) Company Profile & Facts - Yahoo Finance*. [online] Available at:<https://finance.yahoo.com/quote/GSK/profile/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAC98ObQd26dvr_nNItAKTrSTICrGBXR_MFNlJd-JXEx81JlutTMvvikceknm2enelZTlhm5Ma277tNIzHni96ranu1YfJsOddrtTrlXM6ZmUUdcdPnqCLpfTLqd1zvaOeW5xIhLFveljIuxH1QxtT2rv9_IuywwTB0OrQvax7ftN> [Accessed 7 Aug. 2023].

Fong, S.J., Li, G., Dey, N., Gonzalez-Crespo, R. and Herrera-Viedma, E. (2020). Finding an Accurate Early Forecasting Model from Small Dataset: A Case of 2019-nCoV Novel Coronavirus Outbreak. *International Journal of Interactive Multimedia and Artificial Intelligence*, [online] 6(1), p.132. doi:<https://doi.org/10.9781/ijimai.2020.02.002>.

Gençay, R., Selçuk, F. and Whitcher, B.J. (2001). *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. [online] *Google Books*. Elsevier. Available at:<https://books.google.ie/books?hl=en&lr=&id=3ZLTt50EUT8C&oi=fnd&pg=PP1&dq=Gen%C3%A7ay> [Accessed 16 Aug. 2023].

Gitman, L.J., Joehnk, M.D., Smart, S. and Juchau, R.H. (2015). *Fundamentals of Investing*. [online] *Google Books*. Pearson Higher Education AU. Available at:<https://books.google.ie/books?hl=en&lr=&id=DB3iBAAAQBAJ&oi=fnd&pg=PP1&dq=Gitman> [Accessed 16 Aug. 2023].

Glen, S. (2022). [online] Statistics How To. Available at:<https://www.statisticshowto.com/probability-and-statistics/regression-analysis>.

Google (2019). *Classification: Accuracy | Machine Learning Crash Course*. [online] Google Developers. Available at:<https://developers.google.com/machine-learning/crash-course/classification/accuracy>.

Gujarati, D.N., (2022). *Basic econometrics*. Prentice Hall.

Gupta, A., Bhatia, P., Dave, K. and Jain, P. (2019). Stock Market Prediction Using Data Mining Techniques. *SSRN Electronic Journal*. doi:<https://doi.org/10.2139/ssrn.3370789>.

Hadavandi, E., Shavandi, H. and Ghanbari, A. (2010). Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowledge-Based Systems*, [online] 23(8), pp.800–808. doi:<https://doi.org/10.1016/j.knosys.2010.05.004>.

Hargreaves, C. and Hao, Y. (2012). *Does the use of technical amp; fundamental analysis improve stock choice? : A data mining approach applied to the Australian stock market*. [online] IEEE Xplore. doi:<https://doi.org/10.1109/ICSSBE.2012.6396537>.

Ho, S.L. and Xie, M. (1998). The use of ARIMA models for reliability forecasting and analysis. *Computers & Industrial Engineering*, 35(1-2), pp.213–216. doi:<https://doi.org/10.1016/s0360-8352(98)00066-7>.

Hossain, M.A., Karim, R., Thulasiram, R., Bruce, N.D.B. and Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. doi:<https://doi.org/10.1109/ssci.2018.8628641>.

Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C. and Liu, H.H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1971), pp.903–995. doi:<https://doi.org/10.1098/rspa.1998.0193>.

I Daubechies and 10 Lectures On Wavelets (1992). *Ten lectures on wavelets*. Philadelphia, Pa.: Society For Industrial And Applied Mathematics.

Ince, H. and Trafalis, T.B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems*, 37(6), pp.677–687. doi:<https://doi.org/10.1080/03081070601068595>.

J Welles Wilder Jr (1978). *New concepts in technical trading systems*. Greensboro/N. C.: Trend Research.

Jeffrey Xu Yu, Ng, M.K. and Joshua Zhexue Huang (2001). Patterns Discovery Based on Time-Series Decomposition. *Lecture Notes in Computer Science*. doi:<https://doi.org/10.1007/3-540-45357-1_36>.

Jin, B., Gao, S. and Tao, Z. (2022). ARIMA and Facebook Prophet Model in Google Stock Price Prediction. *Proceedings of Business and Economic Studies*, 5(5), pp.60–66. doi:<https://doi.org/10.26689/pbes.v5i5.4386>.

Kamruzzaman, J. and Sarker, R., (2003). Comparing ANN based models with ARIMA for prediction of forex rates. Asor Bulletin, 22(2), pp.2-11.

Khairi, T.W.A., Zaki, R.M. and Mahmood, W.A. (2019). *Stock Price Prediction using Technical, Fundamental and News based Approach*. [online] IEEE Xplore. doi:<https://doi.org/10.1109/SCCS.2019.8852599>.

Khare, K., Darekar, O., Gupta, P. and Attar, V.Z. (2017). Short term stock price prediction using deep learning. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. doi:<https://doi.org/10.1109/rteict.2017.8256643>.

Krollner, B., Vanstone, B. and Finnie, G. (2010). Financial time series forecasting with machine learning techniques: European Symposium on Artificial Neural Networks. *Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN 2010)*, [online] pp.25–30. Available at:<https://research.bond.edu.au/en/publications/financial-time-series-forecasting-with-machine-learning-technique>.

Kumar, D., Sarangi, P.K. and Verma, R. (2021). A systematic review of stock market prediction using machine learning and statistical techniques. *Materials Today: Proceedings*. doi:<https://doi.org/10.1016/j.matpr.2020.11.399>.

Kumar, G., Jain, S. and Singh, U.P. (2020). Stock Market Forecasting Using Computational Intelligence: A Survey. *Archives of Computational Methods in Engineering*. doi:<https://doi.org/10.1007/s11831-020-09413-5>.

Leung, C.K.-S., MacKinnon, R.K. and Wang, Y. (2014). A machine learning approach for stock price prediction. *Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS ’14*. [online] doi:<https://doi.org/10.1145/2628194.2628211>.

Lin, J., Keogh, E., Lonardi, S. and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery - DMKD ’03*. doi:<https://doi.org/10.1145/882082.882086>.

Liu, C., Wang, J., Xiao, D. and Liang, Q. (2016). Forecasting S&P 500 Stock Index Using Statistical Learning Models. *Open Journal of Statistics*, 06(06), pp.1067–1075. doi:<https://doi.org/10.4236/ojs.2016.66086>.

Liu, D. and Li, Z. (2016). Gold Price Forecasting and Related Influence Factors Analysis Based on Random Forest. *Advances in Intelligent Systems and Computing*, pp.711–723. doi:<https://doi.org/10.1007/978-981-10-1837-4_59>.

Lo, A.W., Mamaysky, H. and Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. *The Journal of Finance*, [online] 55(4), pp.1705–1765. doi:https://doi.org/10.1111/0022-1082.00265.

Lohrmann, C. and Luukka, P. (2019). Classification of intraday S&P500 returns with a Random Forest. *International Journal of Forecasting*, [online] 35(1), pp.390–407. doi:<https://doi.org/10.1016/j.ijforecast.2018.08.004>.

Lu, C.-C. and Wu, C.-H. (2009). Support Vector Machine Combined with GARCH Models for Call Option Price Prediction. doi:https://doi.org/10.1109/aici.2009.464.

Lu, W., Li, J., Wang, J. and Qin, L. (2020). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, 33(10), pp.4741–4753. doi:<https://doi.org/10.1007/s00521-020-05532-z>.

Mahalakshmi, G., Sridevi, S. and Rajaram, S. (2016). A survey on forecasting of time series data. *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE’16)*. doi:<https://doi.org/10.1109/icctide.2016.7725358>.

Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. (2014). *Machine Learning: An Artificial Intelligence Approach (Volume I)*. [online] *Google Books*. Elsevier. Available at:<https://books.google.ie/books?hl=en&lr=&id=Aw2jBQAAQBAJ&oi=fnd&pg=PP1&dq=Ryszard+S+Michalski> [Accessed 16 Aug. 2023].

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Education.

Mittal, A. and Goel, A. (2011). *Stock Prediction Using Twitter Sentiment Analysis*. [online] Available at:<http://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf>.

Nayak, A.K.J.R. (2008). Unilever PLC, 1932–2004. *Multinationals in India*, pp.107–134. doi:<https://doi.org/10.1057/9780230227385_6>.

Nicholas Refenes, A., Zapranis, A. and Francis, G. (1994). Stock performance modeling using neural networks: A comparative study with regression models. *Neural Networks*, 7(2), pp.375–388. doi:<https://doi.org/10.1016/0893-6080(94)90030-2>.

Nivetha, R.Y. and Dhaya, C. (2017). *Developing a Prediction Model for Stock Analysis*. [online] IEEE Xplore. doi:<https://doi.org/10.1109/ICTACC.2017.11>.

Parray, I.R., Khurana, S.S., Kumar, M. and Altalbe, A.A. (2020). Time series data analysis of stock price movement using machine learning techniques. *Soft Computing*. doi:<https://doi.org/10.1007/s00500-020-04957-x>.

Rao, C.R. and Gudivada, V.N. (2018). *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*. [online] *Google Books*. Elsevier. Available at:<https://books.google.ie/books?hl=en&lr=&id=gRJrDwAAQBAJ&oi=fnd&pg=PP1&dq=Rao+%26+Gudivada> [Accessed 17 Aug. 2023].

Rounaghi, M.M. and Nassir Zadeh, F. (2016). Investigation of market efficiency and Financial Stability between S&P 500 and London Stock Exchange: Monthly and yearly Forecasting of Time Series Stock Returns using ARMA model. *Physica A: Statistical Mechanics and its Applications*, [online] 456, pp.10–21. doi:<https://doi.org/10.1016/j.physa.2016.03.006>.

Russell, S.D. and Norvig, P. (2010). EBOOK : Artificial Intelligence, A Modern Approach, 3rd edition.

Sadorsky, P. (2021). A Random Forests Approach to Predicting Clean Energy Stock Prices. *Journal of Risk and Financial Management*, 14(2), p.48. doi:<https://doi.org/10.3390/jrfm14020048>.

Satrio, C.B.A., Darmawan, W., Nadia, B.U. and Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Computer Science*, [online] 179, pp.524–532. doi:<https://doi.org/10.1016/j.procs.2021.01.036>.

Schröer, C., Kruse, F. and Gómez, J.M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*, [online] 181, pp.526–534. doi:<https://doi.org/10.1016/j.procs.2021.01.199>.

Shahi, T.B., Shrestha, A., Neupane, A. and Guo, W. (2020). Stock Price Forecasting with Deep Learning: A Comparative Study. *Mathematics*, 8(9), p.1441. doi:<https://doi.org/10.3390/math8091441>.

Shen, S., Jiang, H. and Zhang, T. (2012). *Stock Market Forecasting Using Machine Learning Algorithms*. [online] Available at:<http://masters.donntu.ru/2015/fknt/pozhydaev/library/article3.pdf> [Accessed 27 Dec. 2022].

Shiller, R.J., (2015). Irrational exuberance. In *Irrational exuberance*. Princeton university press.

Sianturi, D. and Rukmi, A. (2022). *Correlation Analysis Between the Number of Confirmed Cases of COVID-19 and Stock Trading in Indonesia*. [online] www.atlantis-press.com. doi:<https://doi.org/10.2991/acsr.k.220202.030>.

Smith, B.M. (2004). *A History of the Global Stock Market: From Ancient Rome to Silicon Valley*. [online] *Google Books*. University of Chicago Press. Available at:<https://books.google.ie/books?hl=en&lr=&id=KJBJzXLEQggC&oi=fnd&pg=PP6&dq=Smith> [Accessed 16 Aug. 2023].

Spiliotis, E., Makridakis, S., Semenoglou, A.-A. and Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily SKU demand forecasting. *Operational Research*. doi:<https://doi.org/10.1007/s12351-020-00605-2>.

Stringham, E.P. and Curott, N.A., (2015). On the origins of stock markets. *The Oxford handbook of Austrian economics*, p.324.

Taylor, S.J. and Letham, B. (2017). Forecasting at Scale. *The American Statistician*, [online] 72(1), pp.37–45. doi:<https://doi.org/10.1080/00031305.2017.1380080>.

Tiwari, S., Bharadwaj, A. and Gupta, S. (2017). Stock price prediction using data analytics. *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*. doi:<https://doi.org/10.1109/icac3.2017.8318783>.

Usmani, M., Adil, S.H., Raza, K. and Ali, S.S.A. (2016). Stock market prediction using machine learning techniques. *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. doi:<https://doi.org/10.1109/iccoins.2016.7783235>.

Vantuch, T. and Zelinka, I. (2015). Evolutionary Based ARIMA Models for Stock Price Forecasting. *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, [online] pp.239–247. doi:<https://doi.org/10.1007/978-3-319-10759-2_25>.

WANG, J. and CHAN, S. (2006). Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, 30(4), pp.605–611. doi:<https://doi.org/10.1016/j.eswa.2005.07.006>.

Yan, M. and Shen, Y. (2022). Traffic Accident Severity Prediction Based on Random Forest. *Sustainability*, 14(3), p.1729.

Yenidoğan, I., Çayir, A., Kozan, O., Dağ, T. and Arslan, Ç. (2018). *Bitcoin Forecasting Using ARIMA and PROPHET*. [online] IEEE Xplore. doi:<https://doi.org/10.1109/UBMK.2018.8566476>.