Advanced High-Frequency Algorithmic Trading: Innovations, Impacts, and Regulatory Challenges in Modern Financial Markets.

Abstract

Objective: This paper will explore how machine learning algorithms may predict the future and make a better prediction for the investor to decide. It tries out many models in its endeavor to find which set of models will best predict stock price movement.

Method: We will apply this method to the historical stock price data using some ML algorithms such as Linear Regression, Random Forest Regressor, Support Vector Regressor, and SARIMAX. The evaluation metrics used here for validating these models are Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

Findings: From the results, it is evident that the more complex machine learning models like Random Forest Regressor, SARIMAX show a more precise stock trend forecasting. Large values were found out by SARIMAX, which depicts a lot of seasonal fluctuation in the market. Therefore, the probability of reflecting their original one shows that the usage of more significant models can help in better stock market predictions for investors to take informed decisions.

Limitations: The main problem of overfitting may occur with complex models when the quality of historic data used in an analysis. Future work may implement more real-time data and make use of numerous models to raise accuracy further.

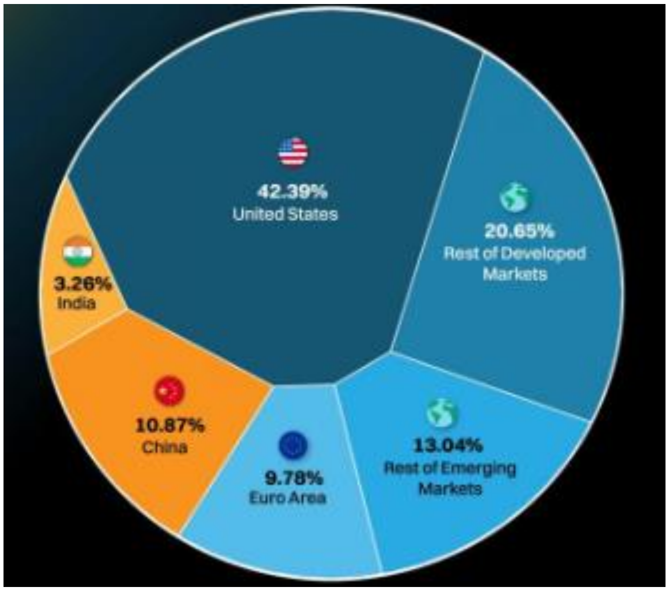
Keywords: Machine learning, stock market prediction, Random Forest Regressor, SARIMAX, time series analysis, investment decisions for stock prices.

Table of Contents

Chapter 1: Introduction

1.1. Background

Stock markets are integral to the international financial system as they enable investments and influence stability (Abbazov, 2024). They act as mediums on which companies go for funds, and investors make their choices based on market conditions. Stock price movements are influenced by several factors, including economic policies, corporate earnings, and market sentiment (Ho and Njindan Iyke, 2017). These changes can affect global investments, hence it is crucial to study stock market behaviors and trends for analysis to make an informed decision. Among all the stock market indices, the Dow Jones Industrial Average (DJIA) is one of the most widely recognised stock market indices, tracking ***30 large publicly traded companies*** in the United States (Biktimirov and Xu, 2019). Currently, the United States has 42. 5% of the global stock market share, stresses the importance of the Dow Jones Industrial Average (DJIA), which encompasses numerous industries and is considered to reflect the general condition of the stock market.



Global stock market share

(Source: Hafeez, 2023)

The above figure reflects the robust presence of the US stock market share worldwide and reflects the importance of DJIA which was ***established in 1896*** (Lin, et al., 2021). It has detailed historical records of stocks and the market which allows for evaluating the trends and changes of the stocks and the market.

Some of the traditional models, which are being used in stock market prediction include the AutoRegressive Integrated Moving Average (ARIMA) and the Generalised Autoregressive Conditional Heteroskedasticity (GARCH) (Zolfaghari and Gholami, 2021). ARIMA helps in identifying linear patterns in time series data, whereas GARCH models are quite effective for the forecasting of market volatility (Dadhich, et al., 2021). However, for indicators such as the DJIA, which witnessed an extremely volatile period of 20% during the year 2008 financial crisis, in this kind of situation the traditional models are not able to capture nonlinearity and shift (Özen and Tetik, 2019). This limitation points out the requirement of advanced methods. Since the conventional models lack in predicting the complex movements of stock an alternative in machine learning seems promising. This can be done through machine learning especially when working with large complicated data since such patterns may rather be hard to identify through conventional methods (Rahul, et al., 2020). Such models keep on upgrading with time based on their exposure to the data for even more accurate prediction processes. The use of this machine learning in the analysis of the stock market is helpful because it is characterised by lots of flexibility, particularly in the big data set.

But as machine learning algorithms progress to be utilised for stock prediction, then it becomes critical to understand the basic idea of a concept in the field of time series analysis, which is, that stock prices tend to vary over time. Periodic time series data has a time dimension and includes stock movement changes information which will be crucial to forecasting trends like the DJIA (Wen, et al., 2019). The analysis of this data is done using machine learning algorithms including LSTM, ARIMA Prophet, etc. LSTM uses long-term orientation, ARIMA addresses linear orientation while Prophet focuses on seasonal orientation (Yadav, et al., 2020). These algorithms increase the level of forecast hence making time series analysis important in the analysis of DJIA trends. Stock prediction using machine learning algorithms is on the rise however some issues still exist as well. Overfitting of this model is another challenge because this is a scenario where it performs excellently on training data sets especially the model but is unable to perform on new data sets effectively making wrong forecasts (Nabipour, et al., 2020). Also, stock indices like DJIA are very active, and therefore the algorithms have their limitations during market extremes. Also, interpretable Machine Learning is a challenge that most of the ML models face, and this makes it almost impossible for the models to be used by investors in making financial decisions (Çelik, et al., 2023).

Machine learning offers further scope in stock predictive analysis irrespective of the fact that there are existing challenges when it comes to conducting trading, as these algorithms are not only able to respond to new trends in the market but perform these tasks in a time-efficient manner as well (Albahli, et al., 2022). Machine learning models also excel at identifying complex patterns in stock data, making them useful for predicting market movements, especially in indices like the DJIA. There is promising progress in predicting stock advances for these models can always incorporate additional information in the system (Alzazah and Cheng, 2020). Likewise, the total value of world stock markets stands at $109 trillion where the U.S. holds the largest 42.5% share of the value. Forecasting within this kind of dynamic changing environment is very critical in ensuring a firm remains financially stable and not taken by surprise by emerging issues (Hafeez, 2023). This investigation is needed to examine how time series ML models can improve stock movement forecasting and direction determination, as shown in the example of DJIA. This study aims to contribute to filling up gaps in this research area and to give useful insights for purposes of stock trend analysis which can help out in the decision-making process in financial markets as an addressing of the shortcomings of the traditional methods.

1.2. Aims and objectives

Aims

The primary purpose of this research is to evaluate the role of machine learning-based time series models in predicting stock movements and identifying trends in the Dow Jones Industrial Average (DJIA).

Objectives

- To conceptualise time series machine learning algorithms in the context of stock market prediction and trend analysis.

- To investigate the key factors that influence stock movement predictions within the DJIA.
- To identify the challenges associated with the adoption of machine learning algorithms in stock market trend identification and prediction.
- To develop and test a predictive model for stock movements using time series machine learning algorithms based on historical DJIA data.
- To propose recommendations for improving the application of machine learning models in stock market predictions and their further development for real-world financial scenarios.

1.3. Research questions

- How accurately can time series machine learning models predict stock movements in the Dow Jones Industrial Average?
- What factors most influence the performance of these models in identifying stock market trends?

1.4. Area of investigation

This research is conducted to develop the ability to improve on the forecast of stock market trends especially the DJIA through time series machine learning algorithms. Most traditional forecasting methods, such as ARIMA and GARCH, fail to capture the complexities of stock market dynamics (Rubio, et al., 2023). With the recent growth of high rates of increase in machine learning, a situation will likely arise where these new algorithms can understand the complexities in markets and produce better results than those previously produced models. Since this study seeks to assess the comparative effectiveness of these algorithms against conventional methods in real-market scenarios, it is also significant to let investors and financial analysts get more accurate tools for financial forecasting. Thereby, there is a need to have this research, so this could therefore enable the formulation of better financial plans and enhanced investment opportunities with better predictive accuracy.

1.5. Research Significance

The present study is useful as it reflects on the utility of machine learning algorithms in improving forecasting capabilities in the stock market by observing movement in the Dow Jones Industrial Average. Traditional methods of forecasting in such a dynamic and complex financial market environment are inadequate (Yadav, et al., 2020). This study is beneficial in understanding how machine learning algorithms outperform to transform investment approaches and offer more accurate market information. The examination and verification of these techniques could contribute to the enhanced practical application of these techniques in financial analysis, enhancing approaches to risk management and maximising profitability in investments. This study aims to contribute to the financial sector by providing robust tools that enable more informed decision-making processes.

1.6. Research structure

Chapter 1 (Introduction): This chapter provides a basic outline and purpose for the research by reflecting on background information, aims and objectives, research questions, and problematic areas.

Chapter 2 (Literature review): This chapter provides a robust theoretical foundation related to the research area by analysing the previously published articles, journals, and research papers of various authors. Further, by examining the previous studies this chapter also proposes a gap in the current study, which helps to provide direction for the undertaken study.

Chapter 3 (Research methodology): This chapter provides an understating regarding the utilised method for data collection and analysis along with the justification, which aids in improving the reliability and validity of study findings.

Chapter 4 (Data analysis and findings): This chapter holds a significant value in the entire research by covering the analysis of collected data, which aids in attaining the specified research aim and objective.

Chapter 5 (Conclusion and recommendations): This chapter is the final chapter, which provides a representation of the extracted findings and results of overall research in a concise manner. In addition, this chapter also provides some key recommendations concerning the research problem and for conducting future studies on similar research areas.

Chapter 2: Literature Review

2.1 Introduction

The following chapter comprises various research studies focused on the development of understanding related to the concepts of the topic of research. It involves analysing the information, knowledge, findings, perspectives, and concepts offered by various authors.

2.2. Concept of time series machine learning algorithms in the context of stock market prediction and trend analysis.

According to Mehtab and Sen, (2020), ML is one of the strongest tools for predicting the stock market since it involves the quantity of data thought when finding complex patterns. This makes analysing past stock movements provide some extent of predictability over future trends by using ML algorithms. The most prominent technique that can aid in understanding past trends and predicting future stock prices is time series analysis. This turns out to be more efficient in dealing with volatility and complexity within financial data than the typical models in which the application of machine learning algorithms can easily take place. In stock market prediction, it will help in finding patterns or trends in historical stock prices; thus, one can accurately predict future movement. (Khan, et al., 2020). The stock market prices are highly responsive to the issues regarding economics, market sentiments, and other worldwide news. In fact, it cannot be forecasted using conventional methods because of its noisy data, irregular patterns, and higher volatility in the process (Dhingra, et al., 2024).

2.2.1. ML algorithm for time series analysis

According to Barrera-Animas, et al., (2022), various machine learning algorithms with each having been particularly developed for time series analysis can enhance the prediction in stock markets. Each algorithm is characterised by differences in the strengths of processing data, identifying trends, and forecasting future movements. In all, models bring in different approaches but focus more on handling complexity, volatility, and non-linear behavior in stock market data, which gives better accuracy in predictions.

2.2.2 ARIMA (AutoRegressive Integrated Moving Average)

As defined by Schaffer, et al., (2021), *ARIMA* is the widely deployed time series forecasting model which comprises three components: the ***AutoRegression (AR), Integration (I), and Moving Average (MA)***. It further argues that AR comprises the relationship of an observation with several lagged observations, and MA uses past forecast errors. Any kind of integration is useful in making data stationary by differencing it.

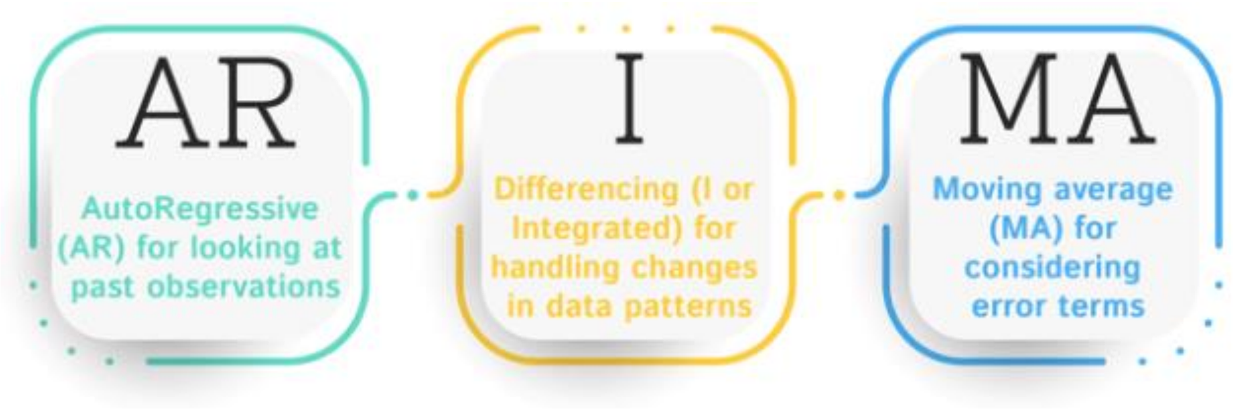

AR
AutoRegressive (AR) for looking at past observations
I
Differencing (I or Integrated) for handling changes in data patterns
MA
Moving average (MA) for considering error terms

Figure 1: Component of ARIMA

(Source: Quantinsti, 2024)

ARIMA is proven to be the best in identifying linear patterns within time series data. As such, it is frequently applied in the predictions of stock markets mainly because it excels at capturing trends within historical stock prices (Xiao and Su, 2022). For example, ARIMA models have been used to predict stock prices of indices like the Dow Jones Industrial Average and the S&P 500. Al-Gounmeein and Ismail, et al., (2021) posit that since ARIMA models prove effective at detecting linear trends, it is effective in stable market conditions. ARIMA struggles to handle non-linearities and sudden market changes, which are common in financial markets. It assumes that the future stock prices depend only on past data and due to this it becomes less useful when quite unpredictable external factors affect the market trends.

2.2.3. LSTM (Long Short-Term Memory) Networks

As per the perception of Chhajer, et al., (2022), ***LSTM*** is a specialised form of recurrent neural network (RNN) designed in the realms of learning long-term dependencies in time series data. Memory cells and such gates as the input gate, forget gate and output gate control information flows as opposed to traditional RNNs.

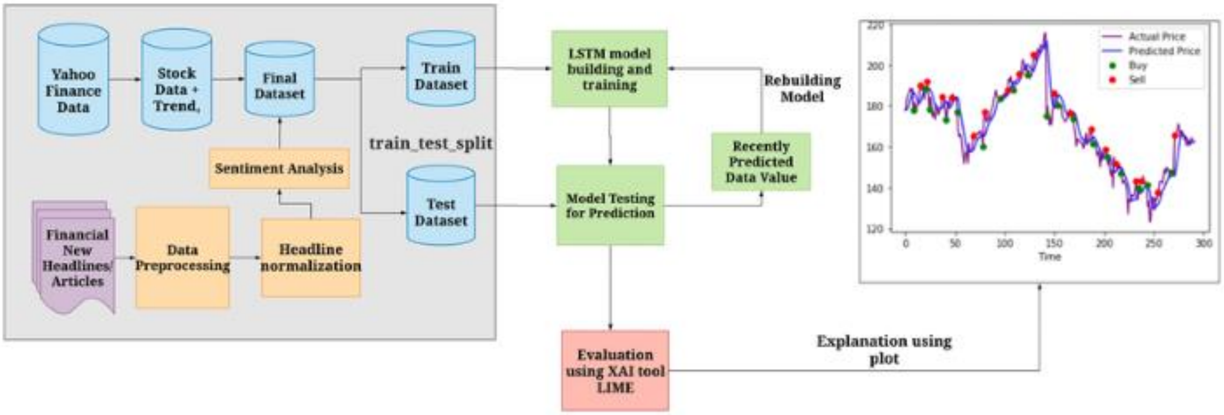


Figure 2: Structure of LSTM algorithm in stock market prediction

(Source: Chung and Shin, 2018)

This network structure helps LSTM retain information of high significance for somewhat longer, hence it is very useful in time-series analysis where the future prediction depends much on the earlier data. In similar lines, Chung and Shin, (2018), proposed that LSTM models of the stock market prediction are suitable as they can handle complex patterns such as volatility and irregular fluctuations. For example, LSTM has been applied to predict the stock prices of major financial indices, allowing for more accurate forecasting during turbulent market conditions. It captures short-term and long-term dependencies and hence is a stronger variant compared with other types of recurrent networks. It is indispensable for the application in stock market forecasting, though this can create heavy computation, models often overfit unless explicitly tuned to avoid fitting for large volatile market data (Chhajer, et al., 2022).

2.2.4. GARCH (Generalised AutoRegressive Conditional Heteroskedasticity)

According to Nyoni, (2018), ***GARCH models*** appear so frequently in the financial time series analysis concerning capturing volatility. GARCH models centre mainly on modelling conditional heteroscedasticity, meaning that they cover periods when stock price volatility changes over time. By combining both autoregressive terms and moving average terms of variance, GARCH models volatility clusters in stock market data. In a similar vein, Belasri and Ellaia, (2017), describe that GARCH is applied very often in the forecasting of market volatility, especially when a financial crisis or deep uncertainty occurs. For instance, GARCH models have been used to predict the volatility of indices like the S&P 500 during the 2008 financial crisis. This model predicts how market volatility might shift, which helps investors better assess risk. GARCH is strictly effective in capturing volatility and therefore is a much more valuable tool for the management of risk. Since the model is short-term and based on the assumption that past volatilities will recur, it limits its use in potentially long-term variations or unpredictable events taking place in the market.

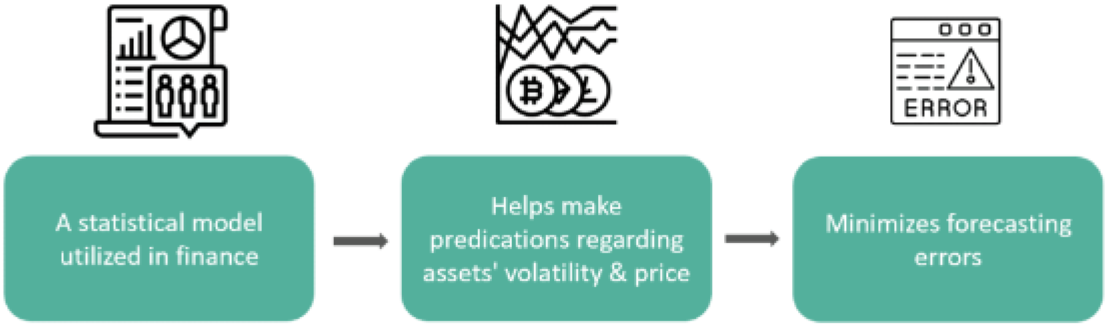

ERROR
A statistical model utilized in finance
Helps make predications regarding assets' volatility & price
Minimizes forecasting errors

Figure 3: Advantage of the GARCH algorithm

(Source: Wall Street, 2024)

2.2.5. Prophet algorithm

According to Xu, et al., (2016), the ***Prophet*** is a time series developed by ***Facebook***. This model has been established for handling seasonality and trend shifts from time series data, good practice in analysing the stock markets. Prophet decomposes the time series into trend, seasonality, and holiday effects, allowing for flexible predictions even with irregular data patterns. Moreover, prophet is found to be very effective in predicting the stock price, especially for seasonal sales companies with market trends. Its flexibility is useful in dealing with short-term fluctuations and long-term trends (Sunki, et al., 2024). Prophet is very strong in handling seasonality and missing data but less effective when irregularities dominate an already volatile market.

2.2.6. Exponential Smoothing

As described by Barrow, et al., (2020), ***Exponential Smoothing*** is a simple and very popular method of forecast which weights past data. This algorithm has three types, to deal with level data the ***Single Exponential Smoothing***; and trends are the ***Double Exponential Smoothing***; for both trends and seasonality is the ***Triple Exponential Smoothing*** known as the ***Holt-Winters method***. Ghania, et al., (2019), highlight that in the context of stock market prediction exponential smoothing is often used in short-term stock market predictions, particularly for identifying short-term trends in stable markets. The computation is fairly simple and fast hence becomes the primary choice for making short-term predictions. This method has less use in turbulent markets whereby the case of non-linear data with huge levels of volatilities holds.

2.2.7. Comparison of ARIMA, LSTM, GARCH, Prophet, and Exponential Smoothing in Stock Market Prediction

Jouilil, (2023), states that ***LSTM*** surpasses other models in treating the non-linearity and complexity of stock market patterns, whereas ***ARIMA*** and Exponential Smoothing are more suited to linear short-term patterns. GARCH performs very well on capturing the volatility, especially during a market crisis, while Prophet is strong at handling seasonality and trends. Additionally, Li and Zhou, 2024 have shown that ARIMA and Exponential Smoothing are extremely computationally efficient, but they have weaknesses in complex patterns. The use of LSTM requires high computational resources, but powerful. GARCH is relatively good at short-term volatility prediction, and Prophet is very well-balanced in terms of computing efficiency and precision in seasonal data.

In conclusion, time series machine learning models such as ***ARIMA to LSTM, GARCH, Prophet, and Exponential Smoothing*** have special strengths in modeling the stock market and fit a variety of market conditions, from the simplest volatility to seasonality.

2.3 Key factors that influence stock movement predictions within the DJIA

2.3.1 Economic Indicators

As per the views of Zakhidov, et al., (2024), economic indicators influence the stock market much, and the markets tend to react fast to a change in any of these indicators. Such indicators include GDP, unemployment, inflation, interest rates, retail sales, and exchange rates. The economic growth rate of a country's GDP happens to be one of the essential indicators of its being economically healthy. A ***country's GDP growth rate*** is an important indicator of economic health and even small changes can have a big impact on the stock market. A poor economy can lead to lower company earnings and lower stock prices. Similarly, He, et al., (2020), suggest that ***inflation*** is one of the most important macroeconomic variables in stock markets as it affects purchasing power and can influence interest rates. It can lead to higher interest rates which decrease stock valuations. This is because higher interest rates make credit more expensive for companies and consumers which can discourage spending and investing.

According to Nti, et al., (2020), the ***unemployment rate*** and the stock market have an inverse relationship which means that when unemployment is low, stock prices tend to be higher. This is because when unemployment is low, there is a reduced labour supply which can lead to higher wages and inflation. Higher wages can lead to inflation as employers raise prices to account for labour costs. However, Shah, et al., (2019), suggest that the ***exchange rate*** is a major factor as when a country's currency appreciates in this case USA as considered for DJIA, foreign investments become more expensive which can lead to capital outflows. When a country's currency depreciates, foreign investments become more attractive which can lead to capital inflows increasing the prices of stock. Rising stock prices are generally associated with an appreciation in exchange rates while declining stock prices are generally associated with a depreciation in exchange rates.

2.3.2 Open price and close price

According to Syed and Bajwa, (2018), A stock's opening price can affect its price throughout the day and can be influenced by many factors, including news, supply and demand, market sentiment, trading, after-market orders and corporate announcements. This price reflects the initial balance of supply and demand of a stock which works on the law of supply and demand. Additionally, it also affects the order flow throughout the day which is attractive for liquidity providers and market makers as they may adjust their quotes which influences the bid in turn the price of the stock. Further, Nenu, et al., (2018) believe that closing prices make a benchmark where by an investor can track a performance of stock as indicated in the references since it will help to judge historical returns thus making it easier to analyse and compare the stock performance. It also helps to discover the true value of stock which helps investors to adjust their positions impacting their decision-making. This process aids in refining stock prices and makes it more reflective of its underlying fundamentals.

2.3.3 Global events

Maqsood et al., (2020), consider global events as one of the significant causes that affect significantly the stock market, and it usually leads towards its increase in volatility. There are several dimensions of global events. For example, geopolitical events may result in a huge impact on stock markets due to uncertainty arising between the world economy. It can cause investors to react quickly which can increase volatility and impact the stock prices of DJIA creating lower lows with time. The investor may reduce their riskier assets and move to safer assets in response to geopolitical shocks. Similarly, Al-Thaqeb and Algharabali, (2019) suggest that economic policies also form part of the factors of global events that are assumed to affect the stock market returns and the overall economy. The volume of stock prices tends to fluctuate before and after a monetary policy announcement due to stimulation. The more the government spends, the more growth will be stimulated and stock prices will be higher. More cut spending hinders growth and thus results in the decrease of stock prices.

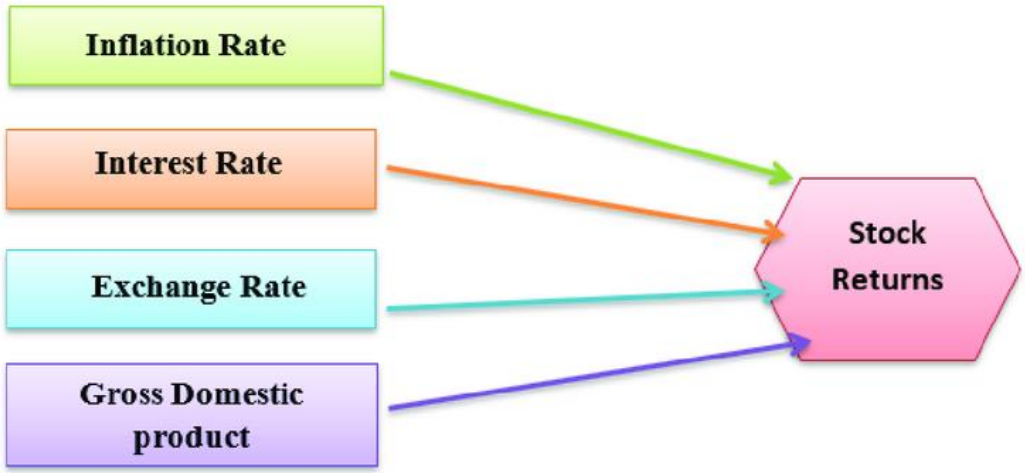


Figure 4: Factors affecting stock market returns

(Source: Kaab, et al., 2023)

2.3.4 Government Policies

The viewpoints of Baker, et al., (2019), suggest that government policies can influence the stock market through various channels such as, taxes, interest rates, monetary policies, regulations, subsidies tariffs and bailouts. Tax policies can influence corporate profits and investor's behavior since they can alter the percentage change of stocks. For example, if the government reduces corporate taxes, companies may retain more profits and stock prices and volume may rise resulting in higher percentage change. When the government raises tax levels, corporate profitability may decline, which may force investors to reassess a firm's valuation. Similarly, Phan and Narayan, (2021), show that the government can also establish monetary policies, such as the repo rate, that is the rate by which banks borrow money from the central bank. Changes in the repo rate will affect the lending rates and corporate borrowing which can highly influence the movements of DJIA share prices. However, policies that promote the growth of the economy like infrastructures can make investors have positive sentiments and hike up the stock prices

2.3.5 Market Liquidity

According to Naik and Reddy, (2021), market liquidity plays a crucial role in the stock market since it determines how quickly and how easily investors and traders can buy and sell assets. Its impacts on the stock market are far reaching since it affects price discovery, price volatility, transaction costs, investor confidence, and economic stimulation among others. In a liquid market, the price of stock accurately reflects all available information because of frequent trading by a variety of participants. Liquidity also impacts large trades on the stock price is minimised because the volume of buy and sell orders can absorb larger transactions without significant price changes. Similarly, the viewpoints of Abudy, (2020), indicate that investors are more likely to invest in a market where they know they can enter and exit their positions at fair prices. This makes it easier for participants to buy or sell stocks including small retail investors and large institutional traders. A market with high liquidity encourages rapid buying and selling, which stimulates the economy influencing the US stock market movements.

From the above discussion, it can be concluded that there are various factors which influence the stock market, especially of DJIA index such as economic indicators, corporate earnings, global events, government policies and market liquidity. They influence significantly as they are directly related to the country's economy and growth.

2.4. Challenges associated with the adoption of machine learning algorithms in stock market trend identification and prediction.

As mentioned by Ghania et al (2019), where there is substantial utilisation of machine-learning algorithms, time-series models in particular. However, the problem turns out to be a highly-coupled kind of dynamics and non-linearity assigned to financial markets-like things Dow Jones Industrial Average or DJIA stock indices. The great efficiency of machine learning models in improving their prediction accuracy poses several barriers on the path toward achieving optimum performance.

2.4.1. Data quality issues in financial markets

According to Gudivada et al., (2017), one of the biggest challenges while trying to predict stock markets with machine learning models is data quality. Even time series algorithms like LSTM-Long Short-Term Memory and ARIMA are keen on cleaning data for developing dependable forecasts; however, most financial datasets face noise, missing values, and inconsistencies. Similarly, Inaba, et al., (2020), most anomalous data quality is due to external events that include political occurrences, sudden market shifts, or unexpected changes in economics. These issues may result in inappropriate interference with the training of the model and lead to wrong prediction and misidentification of trends. Without data quality, even the most complex algorithms for an ML cannot produce relevant insights in stock market analysis.

2.4.2. Overfitting in machine learning models

According to Roelofs, et al., (2019), there are other essential issues with overfitting that define the application of machine learning algorithms where time series models fail, especially in domains such as forecasting the stock market. Overfitting is defined as the scenario whereby a model learns the specifics and the noise of the training data so much that it indeed deteriorates its ability to generalise over new data (Ying, et al., 2019). In a stock market, where conditions often change, overfitting leads to an otherwise great model that may work with historical data but fails at real-time predictions. Usually, trends in DJIA and any other financial time-series data do not seem persistent, that confuses ML models which cannot see between the bigger picture and minor fluctuations (Pardo, et al., 2020).

2.4.3. Interpretability of ML algorithms

From the study of Krishnan, (2020), another challenge in the financial sector is ***the interpretability of machine learning models***. Many advanced machine learning algorithms, such as deep neural networks, function as "black boxes," providing predictions without transparent reasoning. While the ARIMA and LSTM time series models can intuitively follow complex relations among stock data, their absence of interpretability limits their practical use. Financial analysts and investors are very often in need of clear explanations on how and why any such predictions are being made, particularly when large investments are involved (Krishnan, 2020).

2.4.4. Scalability and computational challenges

As per the perception of Potla, (2022), the ***scalability of machine learning algorithms***, in connection with the real-time prediction of the stock market develops a significant technical challenge. Financial markets, including the DJIA, generate vast amounts of data continuously, requiring models to process and analyse this data in real-time. Time series algorithms are very computationally heavy and require tremendous hardware resources-mostly so in the case of a large complex set of data. The demand for real-time predictions further compounds the computational challenges. Time series algorithms are computationally expensive and thus demanding in terms of utilisation of hardware resources. They are memory-intensive, and their scalability in dealing with high volumes of velocity in the stock market at almost the same level of performance is a significant challenge for their successful deployment in the field (Sanz and Zhu, 2021).

2.4.5. Real-time Data processing in Stock Markets

Chen et al., (2023), reflect on the importance of processing real-time data. ***Real-time data processing*** is highly dependent on machine learning models applied in any stock market trend predictions. However, a lot of machine learning models, including time series algorithms experience the well-known latency problem for the handling of huge amounts of data in real time. Financial markets require fast decision-making, and delays in predictions can lead to missed opportunities or losses (Weng, et al., 2017). Time-sensitive models like stock market modeling should be optimised such that they process the data in real time and act on them quickly. However, extremely high computation costs usually make the real-time predictions invalid. This calls for speed-accuracy tradeoff models where speed is an important consideration in a real-time environment.

2.4.6. Regulatory and privacy concerns in financial markets

According to Li, (2017), machine learning algorithms in the prediction of stock markets also have to deal with the ***complicated regulatory financial market environment***. In particular, the use of sensitive financial or personal data must comply with all data privacy regulations, such as the GDPR in the European Union (Voigt and Von dem Bussche, 2017). Additionally, financial markets are heavily regulated, and the use of automated prediction models may raise concerns about transparency and fairness. Machine learning models, including time series algorithms, need to comply with these legal frameworks, which increases the complexity of their adoption.

2.4.7. Adapting to market anomalies

According to Mahakud and Dash, (2017), financial markets are vulnerable to anomalies like economic shocks, which would stabilise the normal market dynamics. Traditionally, algorithms trained using historic data in a time series like LSTM or Prophet have been known to fail during abrupt changes in the market. The underlying non-linearity in a stock market scenario calls for an adaptive model to take on new and unforeseen events; be it a financial crisis or any unexpected geopolitical conditions (Nassif, et al., 2021). Among the biggest challenges are developing algorithms based on machine learning capable enough that they work even when a market anomaly happens without calling for multiple retraining

In conclusion, there is outstanding potential about the general trend and movement of the stock market using time series models and other machine learning algorithms, but several key challenges must first be addressed These include data quality issues, overfitting, interpretability, scalability, real-time data processing, regulatory concerns, and the need to adapt to market anomalies. Addressing these is going to be crucial in improving accuracy and reliability regarding predictions in the stock market with the use of machine learning.

2.5 Predictive model for stock movements using time series machine learning algorithms based on historical DJIA data

The study of Birant, D. and Işık, (2019), suggests that stock market data includes long-term dependencies and due to this, recurrent neural networks-based models could effectively work for such data. Hence, long short-term memory networks (LSTM) and gated recurrent unit (GRU) based models were developed. These models were compared with traditional machine learning approaches. With LSTM, the close prices of the datasets are predicted more consistently than other models. The LSTM model was more successful than the GRU model with similar error metrics in dealing with fluctuations in datasets. However, the study of Yusof, et al., (2020), indicated that the results of time series forecasting show that the prophet model is competitive in modelling the actual market movement with and adoption of appropriate parameters where the measure of mean absolute percentage errors (MAPE) was at 6% at most. In addition to this, the errors of the forecasting result are also compatible with the results of much more complex forecasting models.

The research of Verma, et al., (2023), reviews various articles with the selection based on input indicator's usage. The paper focused on technical indicators and sentiment analysis-based prediction approaches. Financial ratios and influential ratios are considered less. It has been identified that optimised deep learning models are highly efficient and using a metaheuristic-based optimised deep learning framework can help to predict the stock prices of DJIA by considering various possible input indicators. Similarly, Botunac, et al., (2019), emphasise the importance of time-series data filtering when neural network models are used for stock market direction forecasting. The research applied wavelet transformation on input financial data as a preprocessing step which resulted in better use of raw financial data or simple moving averages. Among various neural network models tested in the research, the best results were obtained by using long short-term neural networks and then by using other neural network models.

As per the study of Sheth and Shah, (2023), three methods were implemented in the prediction process of stock prices which were Artificial Neural Network (ANN), Long Short-Term Memory (LSTM) and Support Vector Machine (SVM). These techniques were analysed carefully and it was identified that ANN-based neural network provides the best results as it considers complex, non-linear relationships and recognises patterns. On the other hand, SVM is a new method and can be capable of providing better results in future while LSTM achieved good results only when the large dataset is given which can be considered a major drawback. However, Jena, et al., (2020), conducted a study using spark streaming for processing extremely large data and data ingestion tools such as Nodejs were used. The model developed rests on a distributed computing architecture called Lambda architecture which helped in achieving the goals as intended. It was found that the prediction of stock values is more accurate when support vector regression is applied with an accuracy of 93%.

Damrongsakmethee and Neagoe, (2020), further approximated the efficiency of the model by making use of several performance measures including the mean square error (MSE), root mean square error (RMSE), and mean absolute percentage error (MAPE) and the precision of the model. The findings of the study showed that the Deep LSTM model reached a maximum accuracy of 96% for forecasting stock market time series. Similar to the above case, Jyothirmayee, (2019) it is a research work focused on getting the best estimation of the stock market value. Algorithms used for developing the model are Support Vector Machine (SVM), Random Forest, K-Nearest Neighbour (KNN), and Bernoulli Naïve Bayes. From the results, it can be seen that SVC works best with huge size datasets, and random forest and naïve baye with low size of the dataset. They have managed to achieve 99.87% and 99.81% accuracy for SVC and random forest respectively.

The model developed in the study of Wang, et al., (2021), was a two-stage deep learning technique for the prediction of each feature sub-signal and implement its non-linear integration. This research uses Gaussian process regression to construct the interval prediction of the stock signal and analyse uncertainties of the stock market. After the comprise of time series of various models of the S&P 500, Dow Jones index and NASDAQ, mean absolute percentage errors were 0.55%, 0.65% and 1.11%, respectively. However, in the study of Saetia, K. and Yokrattanasak, (2022), during the prediction process, a combination of technical indicators and Google trends search terms while applying Logistic regression, random forest and extreme gradient boosting (XGBoost) exhibited the highest ROC curves. For successful prediction rate and annualise return, random forest and XGBoost were similar to each other. During critical conditions such as COVID-19, random forest performs marginally better than XGBoost. During normal market conditions, the success rates were average for each model.

Lu and Lu, (2021), indicated that predictive K-line patterns can also be used to predict stock prices. The study identified that the classification effect is better by using this model and the investor sentiment can be more accurately obtained. The achieved accuracy of the model was 85% which laid a foundation for the establishment of the whole stock prediction model. It has been experimental demonstrated that K-line patterns mined using attention mechanism have a highly significant predictive power than general K-line patterns which resulted in the prediction basis of a hybrid neural network.

2.6. Research summary

In summary, it has been identified that time series machine learning algorithms, such as ARIMA, LSTM, and GARCH, enhance stock market predictions by addressing volatility and complex patterns in data. Factors like economic indicators, government policies, market liquidity, and global events significantly influence stock movement predictions. Despite this, various challenges in adopting machine learning algorithms include data quality issues, overfitting, interpretability, and scalability, which affect the accuracy of real-time predictions. The review proposes predictive models that leverage machine learning to improve stock trend forecasting, providing more reliable tools for financial decision-making compared to traditional methods.

2.7. Research gap

This research identifies the inadequacies of existing research works in forecasting using machine learning algorithms for stock markets. Many models, including ARIMA and GARCH, have issues with real-time data and processing market anomalies. Scalability and computational challenges still stand unsolved for big financial datasets. Overfitting is yet another major issue where the model performs well on historical data but cannot perform in actual case studies. Moreover, the lack of interpretability in advanced models such as LSTM makes it difficult for financial professionals to trust their outputs. Current studies also fail to fully address the challenges posed by noisy and incomplete financial data.

Chapter 3: Research Methodology

3.1. Introduction

This chapter will outlines the procedures and strategies that form the cornerstone of conducting this research study on how to accurately forecast the stock market by using machine learning algorithms. It is on detailed methods used for the enhancement of predictive models in terms of accuracy while trying to forecast the trends of the stocks in the face of volatility and data complexity. This chapter discusses time series machine learning techniques. It also represents the evaluation metrics and factors that have been considered as essential toward evaluating to what extent these models correctly predict movement in DJIA stocks.

3.2 Development Environment

The development environment is a core where the machine learning algorithms are built. For this study, Visual Studio Code, or VS Code is utilised for creating and testing the machine learning model. VS Code is an open-source Integrated Development Environment that has gained great recognition for the purpose of writing code across different programming languages (Rask, et al., 2021). This provides one place for coding, editing testing, and packaging software, thus making work much easier and smoother.

VS Code is chosen on the basis of flexibility, speed, and ease of debugging (Tan, et al., 2023). For the purpose of programming language, Python has been selected because it is easy to understand, optimised for machine learning, and cross-platform compliant. Its scalability and readability support building applications that will work on various systems, which makes it suitable for this research study (Srinath, 2017).

3.2.1. Installation of Python Libraries

In this research, several Python modules have been made use of to implement the machine learning models for predictions of the stock market thus streamlining the undertaking of predictive analysis. Some of the most important libraries are Pandas, Sci-kit Learn, NumPy, and statsmodels.tsa.holtwinters, Prophet, and pmdarima.

Pandas Installation: The most widely used library is pandas. When considering handling large data, pandas is the open source primarily for managing most machine learning models (Lemenkova, 2019). To install Pandas, the following command can be used in the terminal:

Command: ***pip install pandas***

Sci-kit Learn Installation: Sci-kit Learn, commonly referred to as Sklearn, is a very powerful library used in building machine learning models. It is furnished with all the tools for classification, regression, clustering, and dimensionality reduction. It also supports data preprocessing handling outliers and missing values (Hao and Ho, 2019). To install Sci-kit Learn, use:

Command: ***pip install sci-kit learn***

NumPy Installation: NumPy is absolutely indispensable while doing mathematical operations; particularly it is handy in cases when one is operating arrays in a model. It supports various mathematical operations and hence is widely applied for data analysis (Lemenkova, 2019). To install NumPy, use:

Command: ***pip install numpy***

Statsmodels Installation (Exponential Smoothing): The ExponentialSmoothing module from the statsmodels.tsa.holtwinters library is an application of the time series forecasting; specifically the data trend is subject to smoothing. It helps predict the stock market trends. Install it using:

Command: ***from statsmodels.tsa.holtwinters import ExponentialSmoothing***

Prophet Installation: Prophet is a tool, developed by Facebook, to handle time-series data that includes seasonal trends. It can be very useful in predictive models in regard to stock prices and trends (Zemkoho, et al., 2022). Install Prophet with the command:

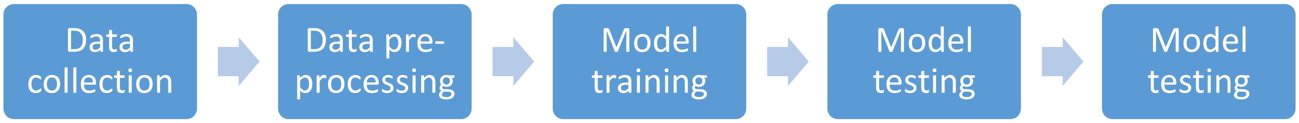
Command: ***from prophet import Prophet***

pmdarima Installation: pmdarima allows simplification in implementing ARIMA models which can automatically select the most appropriate model for efficient prediction (Hodeghatta and Nayak, 2023). To install pmdarima, use:

Command: ***import pmdarima as pm***

3.3. Research design

The research design explains the plan and structure guiding the process of conducting this study. In other words, it forms a kind of blueprint for the realisation of research objectives or solving the identified problem (Dannels, 2018). In particular, this study aims to develop models for the prediction of stock movement, especially for the Dow Jones Industrial Average (DJIA).



(Source: Created by Author)

Data gathering is very much an important way of obtaining the needed history of stock movements from the trusted public source of data, investing.com, which is quite well known and a trusted source for financial data and delivers credible information regarding the stock market. Once data is being collected, a cleaning or preprocessing phase occurs to account for any instance with missing values or outliers during this process and help the model not lose accuracy due to such issues (Adineh, et al., 2020). The cleaned data is now transformed to be applied on various algorithms for machine learning after that.

After implementing the model, a description of the results it yields is presented along with a performance measurement using key metrics. These would be crucial in establishing the most optimised and effective model concerning predicting stock movement within the DJIA. This research design strategy ensures that the study is structured efficiently to meet its objectives and give valuable information about the stock market prediction using machine learning techniques.

3.4. Dataset description

The study obtained historical data from the online platform Investing.com in developing its forecasting models in relation to stock market activities. The dataset contains historical data for the Dow Jones Industrial Average (DJIA) and is available at: ***https://in.investing.com/indices/us-30-historical-data.*** The data is simple, containing information like date, opening price, closing price, the highest and lowest prices, trading volume, and percentage change. This data is selected for its accuracy and regular updates, making it highly reliable for training machine learning models. The comprehensive and structured nature of the dataset aligns with the study's aim of analysing the stock market trend and predicting future movements.

3.5. Pseudocode

```
PROCEDURE: LOAD THE DATASET

Data= loaddata(“DJIAdata.csv”)

PREPROCESS THE DATA

Data =PreprocessData(data)

SPLIT THE DATA INTO TRAINING AND TESTING SETS

Train_data,  test_data = SplitData(data, test_size=0.2, ranadom_state=42)

INITIALIZE A LIST OF MODELS

Models=Initializemodels()

TRAIN AND EVALUATE EACH MODEL

Results

TRAIN REGRESSIVE ALGORITHMS

LINEAR REGRESSION
```

```
Lrmodel= linearregression()

Lrmodel.fit(X_train, y_train)
```

RANDOM FOREST REGRESSOR

```
Rfmodel=randomforestregressor(n_estimators-100, random_state=42)

Rfmodel.fit(X_train, Y_train)
```

SVR

```
svrmodel= SVR(kernel= ‘rbf’)

Svrmodel.fit(X_train, y_train)
```

XGBOOST

```
XGB-boost=xgb.xgbregressor(objective=’reg:squarrederror’,                  n_estimators=100,
random_state=42)

Xgb_model.fit(X_train, Y_train)
```

TIME SERIES ALGORITHM

ARIMA

Fitting the Auto ARIMA model

```
auto_arima_model = pm.auto_arima(train['Price'], start_p=1, start_q=1,
                     test='adf',
                     max_p=3, max_q=3,
                     seasonal=False,
                     stepwise=True)
```

SARIMA model

```
sarima_model = SARIMAX(train['Price'],

               order=(1, 1, 1),
               seasonal_order=(1, 1, 1, 12),
               enforce_stationarity=False,
               enforce_invertibility=False)


sarima_fit = sarima_model.fit()
```

Prophet model

```
# Initializing and fit the model
prophet_model = Prophet()
prophet_model.fit(df_train)
```

ETS model

```
# Fitting the ETS model with a damped trend
ets_model = ExponentialSmoothing(
  train['Price'],
  trend='add',
  damped_trend=True,
  seasonal='add',
  seasonal_periods=12
)
ets_fit = ets_model.fit()
```

PERFORMANCE EVALUATION

For each model

```
mae = mean_absolute_error(y_test, forecasted_values)
mse = mean_squared_error(y_test, forecasted_values)
r2 = r2_score(y_test, forecasted_values)
```

Make Predictions

For TIME SERIES ALGORITHM

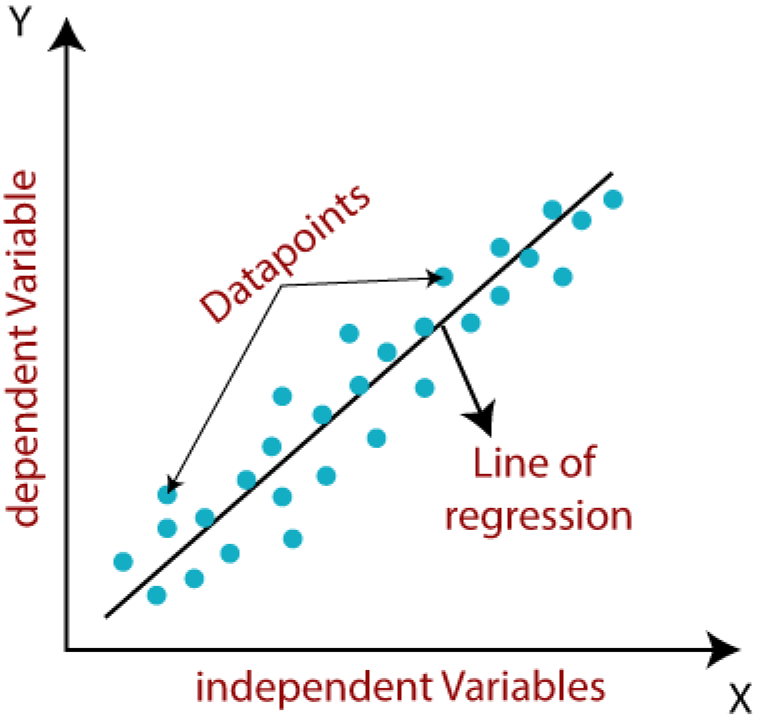
Visualise them

3.6. Research Deployment Experimental Setup

This research employs various machine learning techniques, including both regressive as well as time-series-based techniques. The regressive algorithms used are Linear Regression, Random Forest Regressor, Support Vector Regressor, and XGBoost Regressor. The applied algorithms are ARIMA, SARIMA, Prophet, and Exponential Smoothing. For this reason, such algorithms can very efficiently be able to handle complex stock market data in optimizing the prediction of the future shift of the Dow Jones Industrial Average (DJIA).

3.6.1. Regressive algorithm

Linear Regression: This algorithm is used for modeling the relationship between one dependent variable, which is stock prices, and one or more independent variables. It simply works by drawing a straight line in such a way that the difference between actual and predicted values is as minimal as it can get. This algorithm is helpful in identifying stock price data with a general trend, as this algorithm gives a very basic and straightforward idea about their movement pattern (Jia, et al., 2017).



(Source: Javatpoint, 2024)

This research could thus potentially make use of a base setup with Linear Regression over more sophisticated algorithms to give preliminary results about how the price of stocks may be.

Random Forest Regressor: The algorithm of the machine learning model called Random Forest Regressor constructs many decision trees and computes the average of their predictions to achieve higher accuracy. This is also another reason why this algorithm would do well in dealing with complex relationships presented by stock market data because the stock market non-linear pattern often occurs (Graw, et al., 2021).

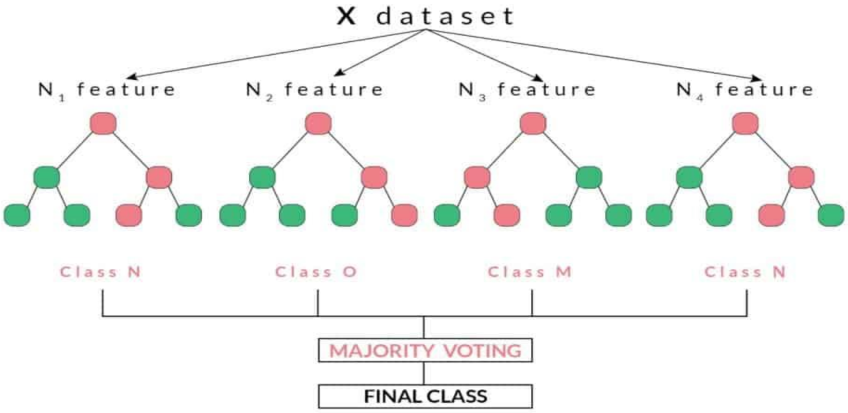


Figure 5: Random forest regressor

(Source: Graw, et al., 2021)

Using random subsets of data should be beneficial in avoiding overfitting; therefore, the generalising ability to new data will be improved. Random Forest in this study will exhibit some key determining factors that draw the pattern of stock prices while making pretty robust predictions by identifying complex multivariant patterns.

Support Vector Regressor: The procedure used to predict continuous variables through the hyperplane that provides minimum error in prediction is utilized in Support Vector Regressor. It actually applies very well, particularly when handling complex, high-dimensional data in the stock markets (Zhang and O'Donnell, 2020).

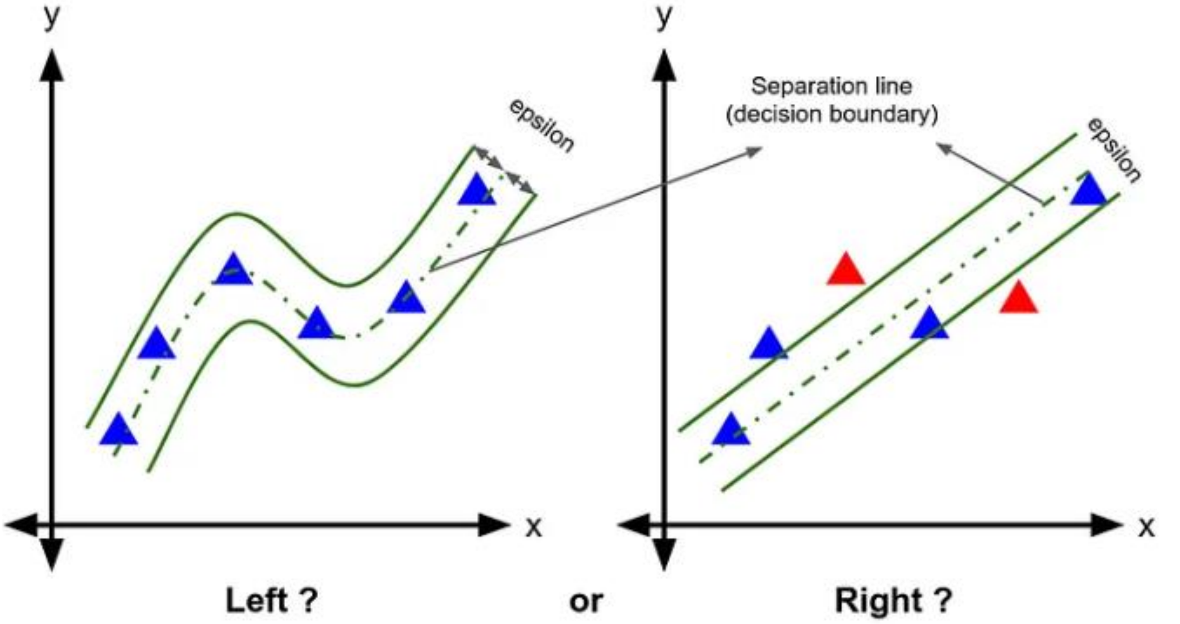


Figure 6: Concept of SVM Regressor

(Source: Achsan, 2020)

In this research, SVR helps forecast stock prices because it controls the non-linear relationship and outliers in the data. Its ability to accept irregular patterns gives it the necessary flexibility in projecting trends about monetary finances though anomalies and sudden changes do occur while forecasting stock prices.

XGBoost Regressor: XGBoost is considered the best among the gradient boosting algorithms that boosts the performance of prediction through an ensemble of several weak models. It is also very efficient and has good working capabilities with large complex datasets (Dong, et al., 2022).

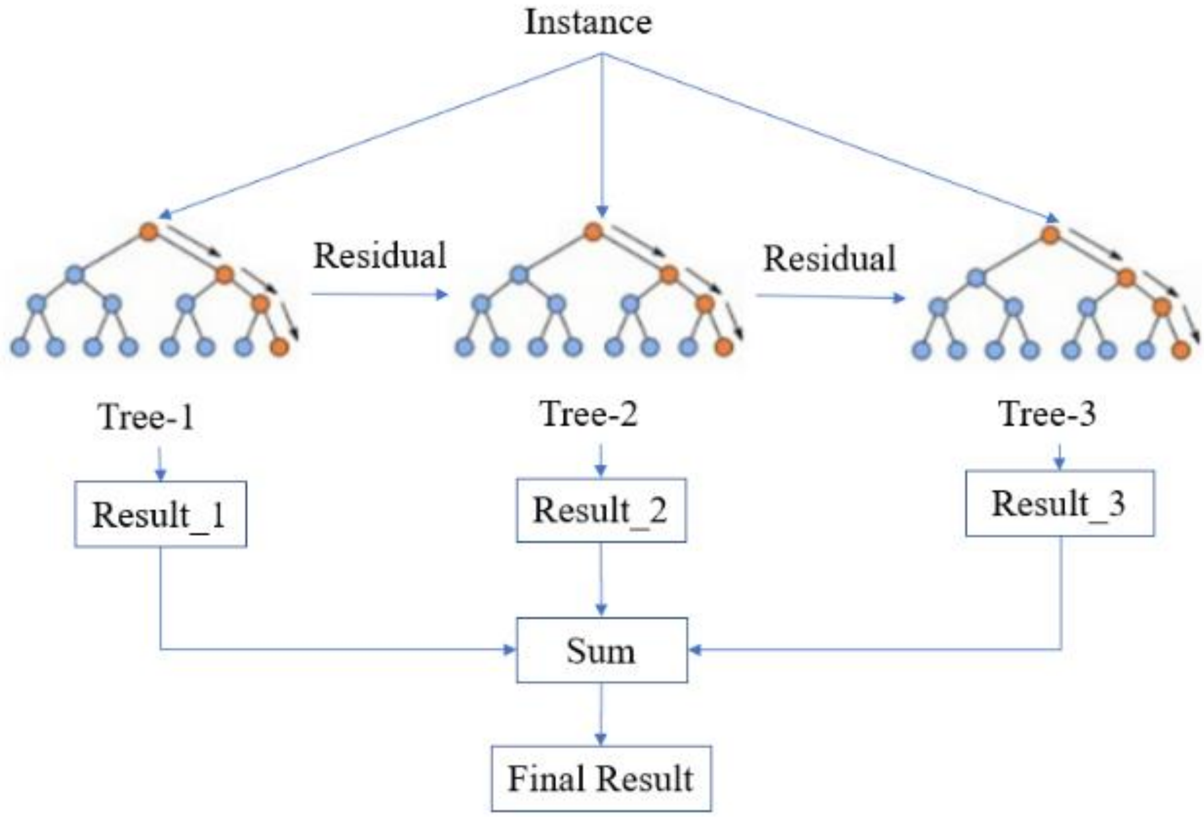


Figure 7: XGBOOST Flow chart

(Source: Dong, et al., 2022)

In this research, XGBoost is used in stock price forecasting where both linear as well as nonlinear relationships are solved. It can minimise error and process big amounts of data within a highly minimised time span, making it a prime choice for making predictions concerning the movements in a stock market with good accuracy.

3.6.2. Time-series based algorithm

ARIMA: ARIMA is an autoregressive integrated moving average model, that is widely used in time series prediction for forecasting future values based on certain patterns in past data. The basic characteristics are autoregression, differencing, and also a moving average to capture the trend in the stock price data.

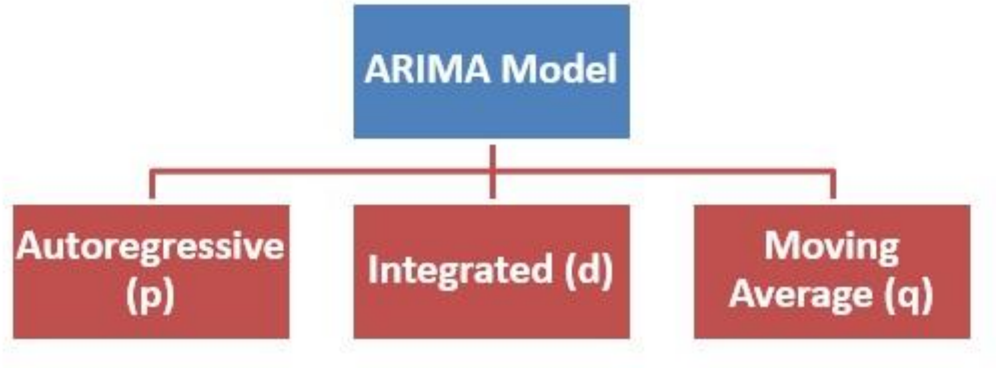


Figure 8: Component of ARIMA

(Source: Created by Author)

ARIMA is well-suited for short-term forecasting of stock price movements as this can process time series data with easily definable trends or patterns (Xue and Hua, 2016). It is useful in terms of financial analysis and trend prediction because it allows for the prediction from past data.

SARIMA: SARIMA (Seasonal ARIMA) is just an extension of ARIMA, capturing seasonality in the given data. SARIMA should be used with stock markets whose price trends repeat over fixed intervals, such as quarterly or yearly (Lee, et al., 2018).

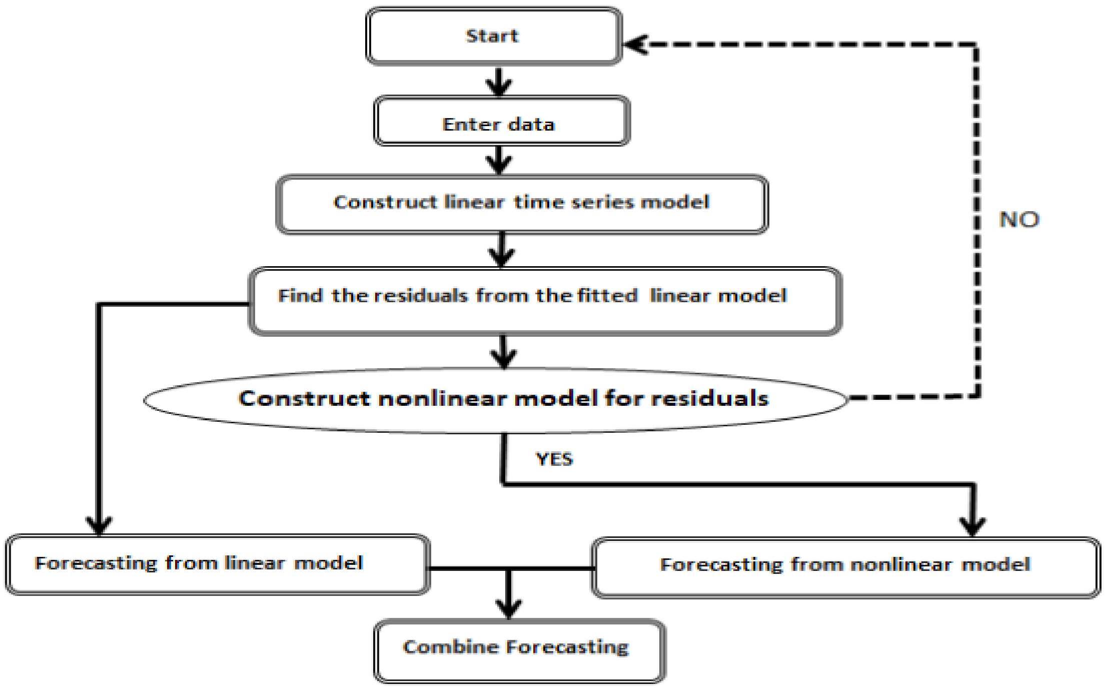

Start
Enter data
Construct linear time series model
Find the residuals from the fitted linear model
Construct nonlinear model for residuals
NO
YES
Forecasting from linear model
Forecasting from nonlinear model
Combine Forecasting

Figure 9: SARIMA model flow chart

(Source: Lee, et al., 2018)

In this research, SARIMA is applied to capture both regular and seasonal trends in stock prices, providing more accurate forecasts. Its nature makes it suitable for long-run predictions in finance as financial markets are dominated by cyclical trends, especially when it comes to stock prices.

Prophet: Prophet is an elastic time series forecasting method. The method was created for applications in which a data set exhibits both seasonality and missing values. Prophet is built by Facebook. It automatically finds long-term patterns and seasonality and this makes for very high-quality time series forecasts, for example, for the prediction of stock prices (Yusof, et al., 2020). In this research, the Prophet models and forecasts the stock prices based on a model that has been designed to learn recurring patterns over time. It has the vast ability in handling large data sets with provision to adjust missing data that renders it very useful in financial forecasting especially at times when data may not be perfectly aligned or complete.

Exponential Smoothing: Exponential Smoothing can be defined as a type of time series forecasting that assigns exponentially decreasing weights to past observations. This method aids in boosting the predictions regarding stock price movements by smoothing up short-term fluctuations and trying to reflect the nature of long-term trends (Hansun, 2016). In this research, Exponential Smoothing is used to capture underlying patterns in stock market data for more accurate trend prediction.

3.7. Performance evaluation metrics

The performance of the model should be evaluated in order to decide the success of a stock price forecasting model. For this research, three key performance metrics are used:

Mean Absolute Error (MAE): It is defined as the average magnitude of errors between predicted and actual values, giving an idea about the accuracy of the prediction (Hodson, 2022).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Mean Squared Error (MSE): MSE calculates the squared discrepancies between the predicted and actual values, making large errors count multiple times, thus it is easy to assess the accuracy of prediction (Hodson, et al., 2021).

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

R² (Coefficient of Determination): R² measures how much of the variance in the dependent variable the model explains, thus indicating how well the model fits the data (Sutrick, 2017).

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Chapter 4: Data Analysis

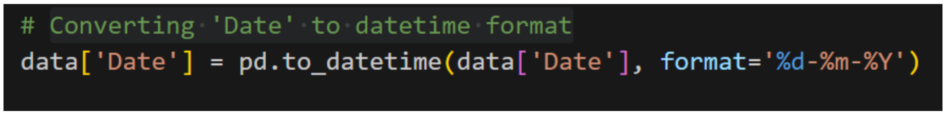
4.1 Introduction

This chapter includes a few basic machine learning concepts, principles, and algorithms, which have been utilized in creating an early-stage prediction model for diabetes. The research conducted about this study is, in this section, interpreted with pretreatment of data as well as the actual implementation of the model.

4.2 Exploratory data analysis

Exploratory Data Analysis (EDA), is the process of evaluating the nature of a dataset and general properties without building towards modeling. It also helps in finding out the type of distribution and the deviation present, relationships, abnormalities that are consequently used during data preparation procedures. Examples of typical EDA techniques include histograms, scatter plots, box plots, and correlation matrix.

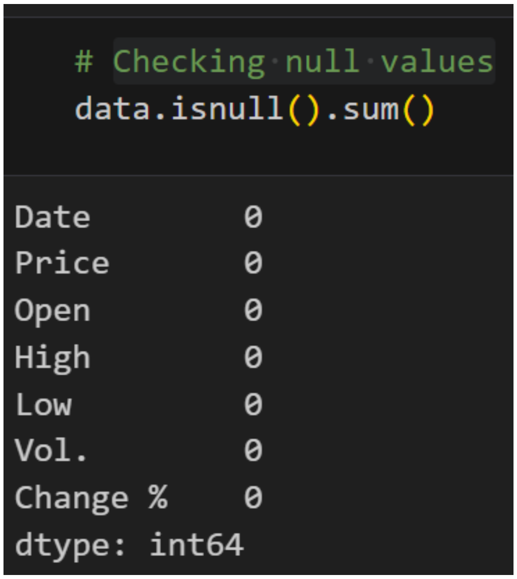
4.2.1 Data Preprocessing



```
# Converting 'Date' to datetime format
data['Date'] = pd.to_datetime(data['Date'], format='%d-%m-%Y')
```

Figure 10: Converting date to DateTime format

The 'Date' column in the stock market data is very important as it allows an analysis based on time. From datetime format, at least the dataset can be divided based on time functions like data slicing using dates and getting the information regarding the day, month, and year. The format conversion also aids in creating time-dependent features such as lag variables and provides instruments for time-series analysis. DateTime formatting is important to keep the time integrity of the stock data to be coherent.

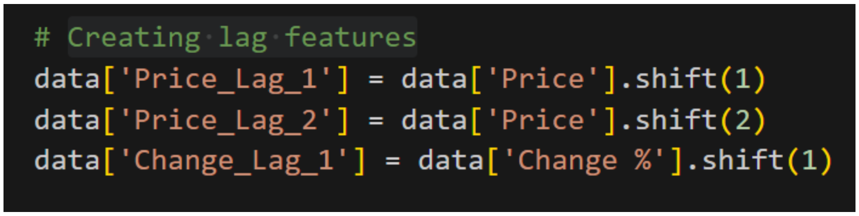


```
# Checking null values
data.isnull().sum()
```

```
Date          0
Price         0
Open          0
High          0
Low           0
Vol.          0
Change %      0
dtype: int64
```

Figure 11: Checking null values

Identifying missing or null values is essential for ensuring data quality. Missing data is a significant issue because a large number of machine learning algorithms do not work well with such data and can produce inaccurate predictions. In this data, there are no missing values present in the dataset which was checked by data.isnull().sum(). By checking for null values, it can be determined whether the dataset is complete or if there are anomalies present such as missing stock prices or volumes. These missing values are resolved at an early stage of the process to maintain the timeframe and quality of data for further tasks such as feature engineering and model training.

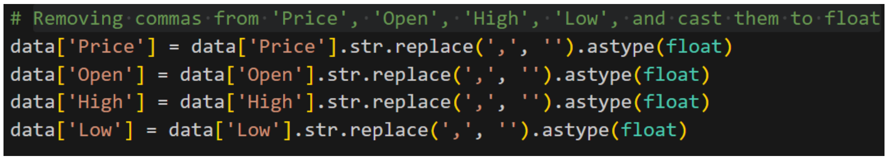


```
# Creating lag features
data['Price_Lag_1'] = data['Price'].shift(1)
data['Price_Lag_2'] = data['Price'].shift(2)
data['Change_Lag_1'] = data['Change %'].shift(1)
```

Figure 12: Creating lag features

Lag features are used where information about the price or another variable at a previous period is needed. These features are important in time series because stock prices are dependent on historical prices most of the time. When working with lag features, incorporating historical prices enables the model to identify dependencies. For instance, the price of the previous day or percentage change, the model will receive the information about the prices of the last day, which in turn will

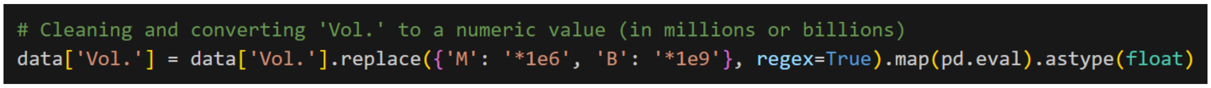
\`\`\`



```
# Removing commas from 'Price', 'Open', 'High', 'Low', and cast them to float
data['Price'] = data['Price'].str.replace(',', '').astype(float)
data['Open'] = data['Open'].str.replace(',', '').astype(float)
data['High'] = data['High'].str.replace(',', '').astype(float)
data['Low'] = data['Low'].str.replace(',', '').astype(float)
```

Figure 13: Removing commas from variables

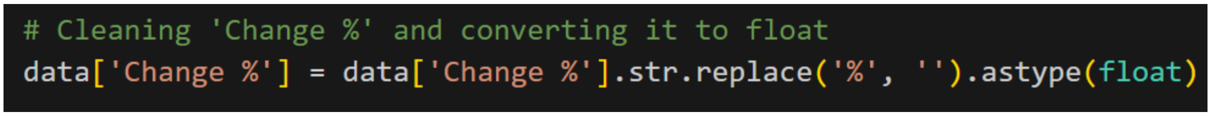
Stock market data often contains commas in numeric columns, such as 'Price' or 'Open'. These commas should be deleted before going through mathematical operations. Converting the cleaned data to float makes it possible to make computations such as means, standard deviations or scaling for a machine. Otherwise, any action performed on these columns would not be possible or it would return wrong results. Converting these values to float ensures that they are ready for numerical analysis.



```
# Cleaning and converting 'Vol.' to a numeric value (in millions or billions)
data['Vol.'] = data['Vol.'].replace({'M': '*1e6', 'B': '*1e9'}, regex=True).map(pd.eval).astype(float)
```

Figure 14: Converting volume to numeric values

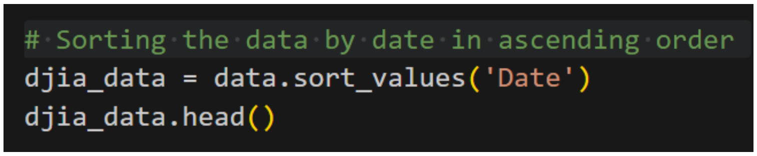
The 'Vol.' column represents the trading volume and is often formatted with abbreviations like 'M' (millions) or 'B' (billions). For this column to be useful for analysis, they have to be cleaned and converted into a common numerical format that can be read by statistical software. This step is particularly important when calculating volume-based metrics, such as the volume-weighted average price (VWAP), which relies on accurate volume data to reflect the market's activity. This helps in maintaining consistency to avoid inconsistencies in the volume data and in arriving at the right scaled value.



```
# Cleaning 'Change %' and converting it to float
data['Change %'] = data['Change %'].str.replace('%', '').astype(float)
```

Figure 15: Converting % change to float

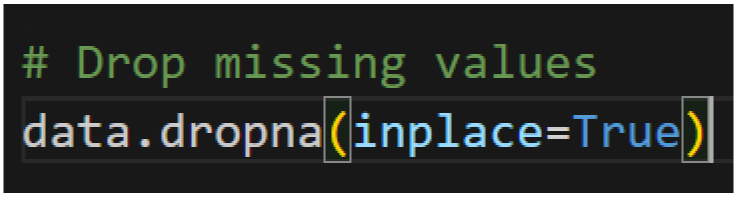
The 'Change %' column indicates the percentage change in stock prices and often contains a percentage symbol. To perform the numerical operation, the values must be converted to the float. This step enables the model to extract percentage differences in stock price changes which is key in calculating volatilities. Percentages have to be neat and properly formatted since lag features as well as average values based on changes in stock prices are produced.



```
# Sorting the data by date in ascending order
djia_data = data.sort_values('Date')
djia_data.head()
```

Figure 16: Sorting data in ascending order

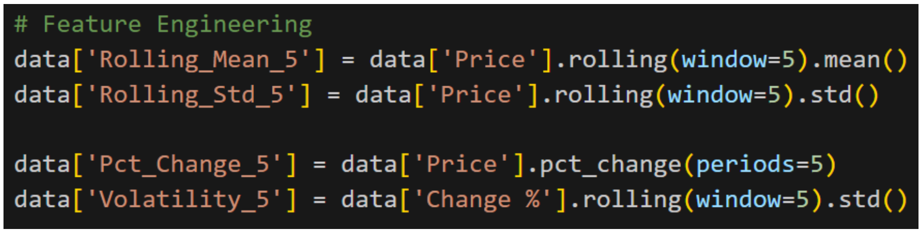
Arranging the data in advance by date ensures that time sequences of data are followed in the particular dataset applied to time series analysis. Most of the models in supervised learning, and especially those in the forecasting methods, rely on the sequential data structure. Sorting the data by date makes it easy to perform some operations such as feature engineering, computation of rolling statistics and the making of time-based predictions.



```
# Drop missing values
data.dropna(inplace=True)
```

Figure 17: Drop missing values

After identifying null values, any missing entries are removed from the dataset. In financial data, missing values often occur due to holidays or market closures. This can cause issues when performing data analysis if not properly handled. In other cases, imputation might be done but it is better to eliminate missing records for this field to ensure its integrity due to time sensitivity, especially for stock prices. This allows for cleanliness as well as credibility of data to be processed and analysed when certain entries are deleted.



```
# Feature Engineering
data['Rolling_Mean_5'] = data['Price'].rolling(window=5).mean()
data['Rolling_Std_5'] = data['Price'].rolling(window=5).std()

data['Pct_Change_5'] = data['Price'].pct_change(periods=5)
data['Volatility_5'] = data['Change %'].rolling(window=5).std()
```

Figure 18: Feature engineering

Feature engineering is the process, which involves the generation of new features that add more valuable information about the variable. Some examples are moving averages, rolling standard deviations and volatility measures, which help in tracking the trends and momentum of stock prices. For instance, the moving average deletes short-term oscillations while strengthening the long-term trends or tendencies and also the rolling standard deviations assist in assessing the price movements. By generating these features, we enrich the dataset and provide the model with more comprehensive information to improve its predictive capabilities.

4.2.2 Visualisations

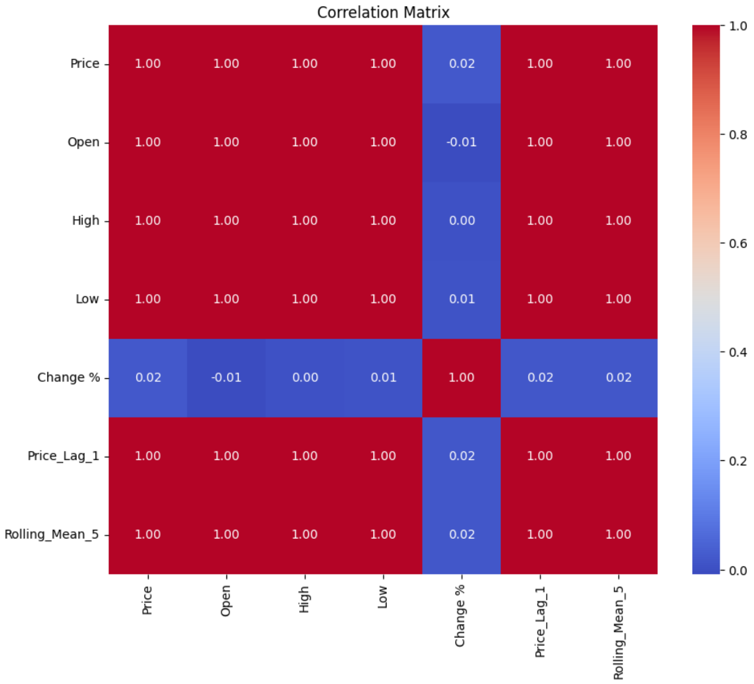


Figure 19: Correlation matrix

The above figure represents the correlation matrix of the dataset of DJIA stock prices. The majority of the variables are highly correlated to each other such as open price, day's high, price and change %. This can be due to various reasons such as institutional involvement, supply and demand fluctuations, correlation among assets, market sentiment, lack of identical market conditions and after-hours trading.

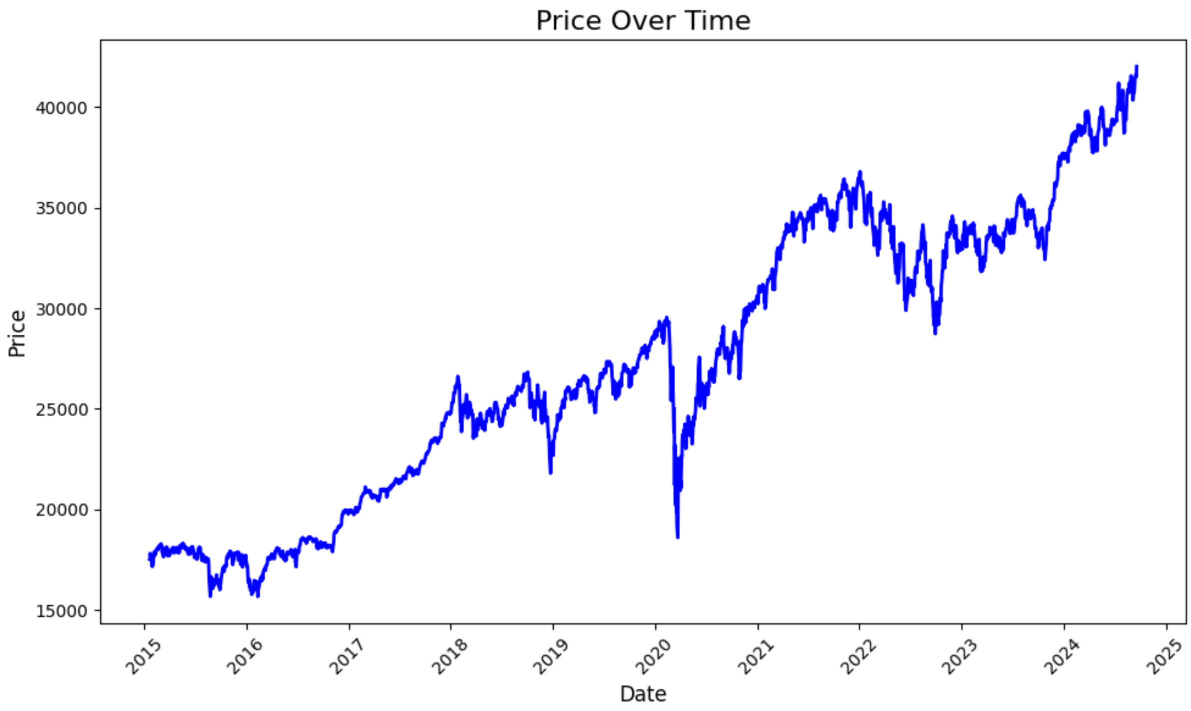


Figure 20: Price over time

(Source: Created by Author)

The above figure represents the price over time of the DJIA of the last 10 years. The price of the Dow Jones industrial average has been consistently increasing for the last 10 years. There were some market crashes such as in 2020 which was the biggest stock market crash of all time around the world which was due to the covid19 pandemic. The market recovered from it and created new all-time highs. This can be due to various reasons such as economic growth, strong GDP, optimism about interest rate cuts, company performance and stock splits.

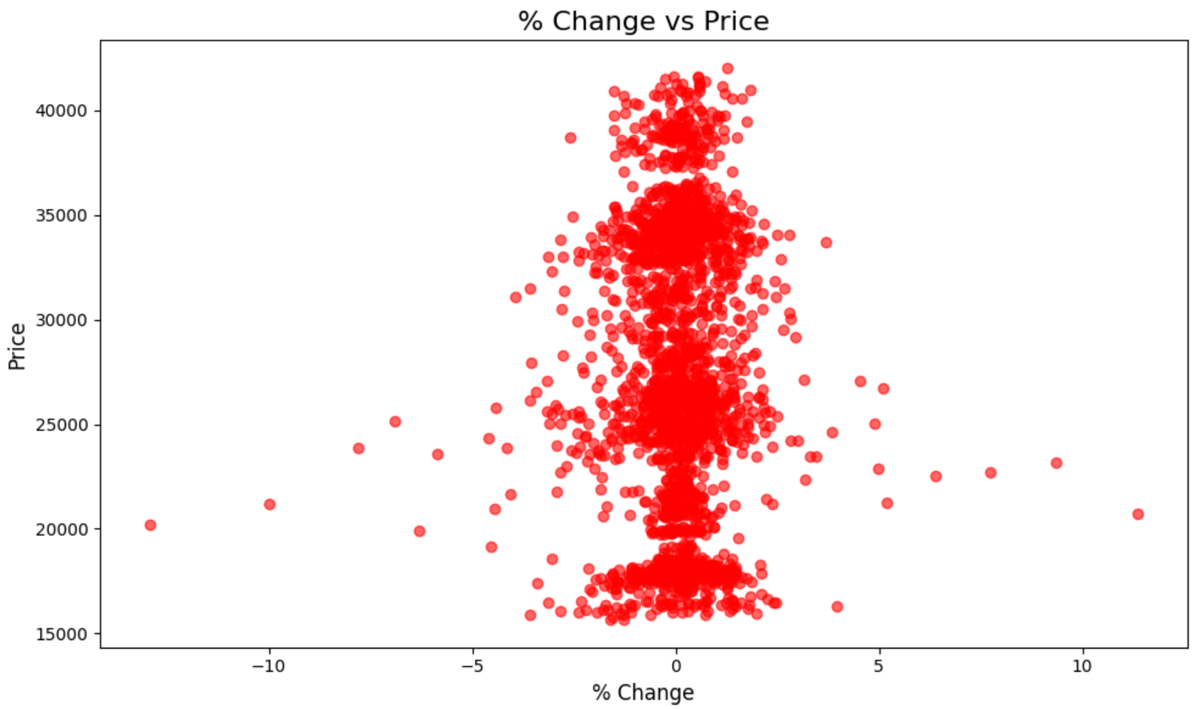


Figure 21: Percentage change versus the price of DJIA

(Source: Created by Author)

The above figure represents the percentage change in the price of DJIA in comparison with its price. It can be observed that the data points of percentage change in the price of the stock are clustered within the range of -5 to 5 irrespective of the price. This indicates that the price of the stock does not change significantly in a smaller period. the price of stock is affected by various factors such as political issues, economic concerns, earnings disappointments, a company's financial health and future profitability. Due to these factors, price does not have a major percentage change.

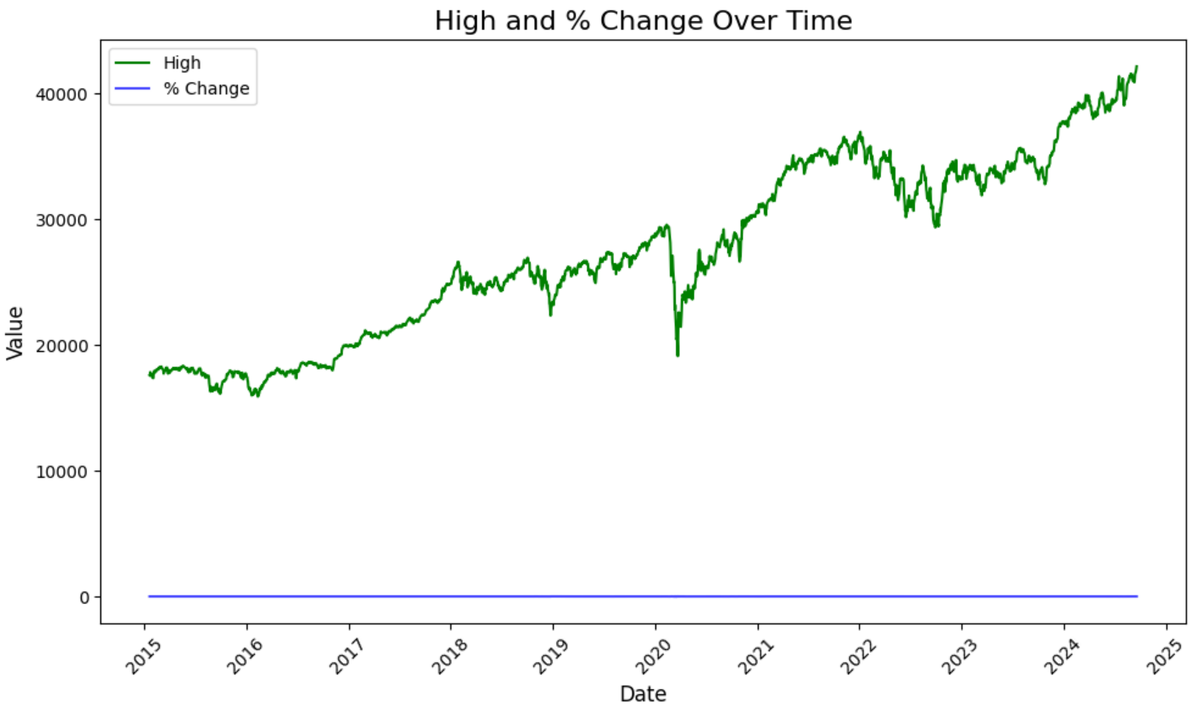


Figure 3: High and % change over time

(Source: Created by Author)

The above figure represents the day's high price and percentage change of DJIA over time. The day's high price from the last 10 years of the index is consistently increasing. Considering percentage change, it can be observed that there is no significant change in price in the smaller time frame. A stock price can be increased while the percentage change is low because the stock is already trading at a high price, meaning even a small absolute price increase represents a relatively small percentage change compared to its current value. The higher the starting price, the smaller the percentage change for the same absolute price increase.

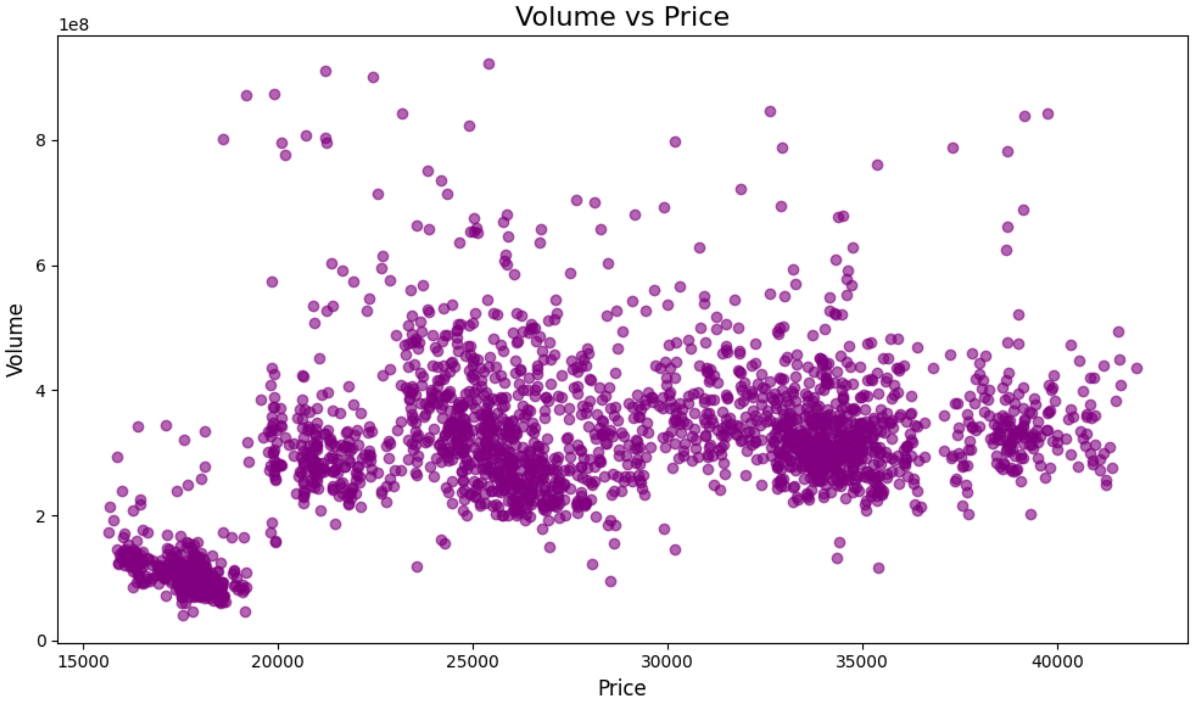


Figure 23: Volume vs price

(Source: Created by Author)

The above figure represents the volume of DJIA according to price. The volume of the index has been consistently increasing with the majority of the data points clustered within the range of 2 to 6. There can be several reasons such as a change in DJIA composition, strong GDP and optimism about interest rates. When stocks are added or removed from DJIA, trading volumes can change significantly especially on the trading day.

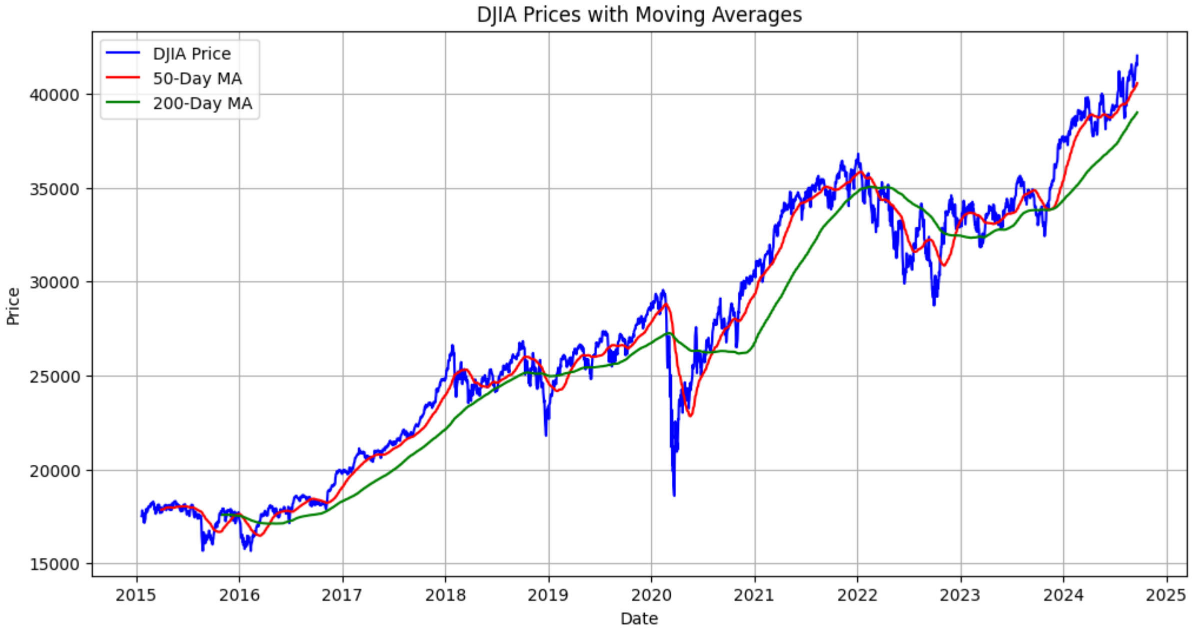


Figure 24: DJIA price with moving averages

(Source: Created by Author)

The above figure represents the price of DJIA with moving averages. A moving average (MA) is a technical indicator used in stock market analysis to smooth out price data and identify trends. It can be observed that when the 50-day MA crosses the 200-day MA, the price of AJIA follows the stock a 50-day MA from the last 10 years. This can be due to the reason that it is a trend indicator that is used in trading and coupling it with 200-day MA describes the support and resistance of the stock.

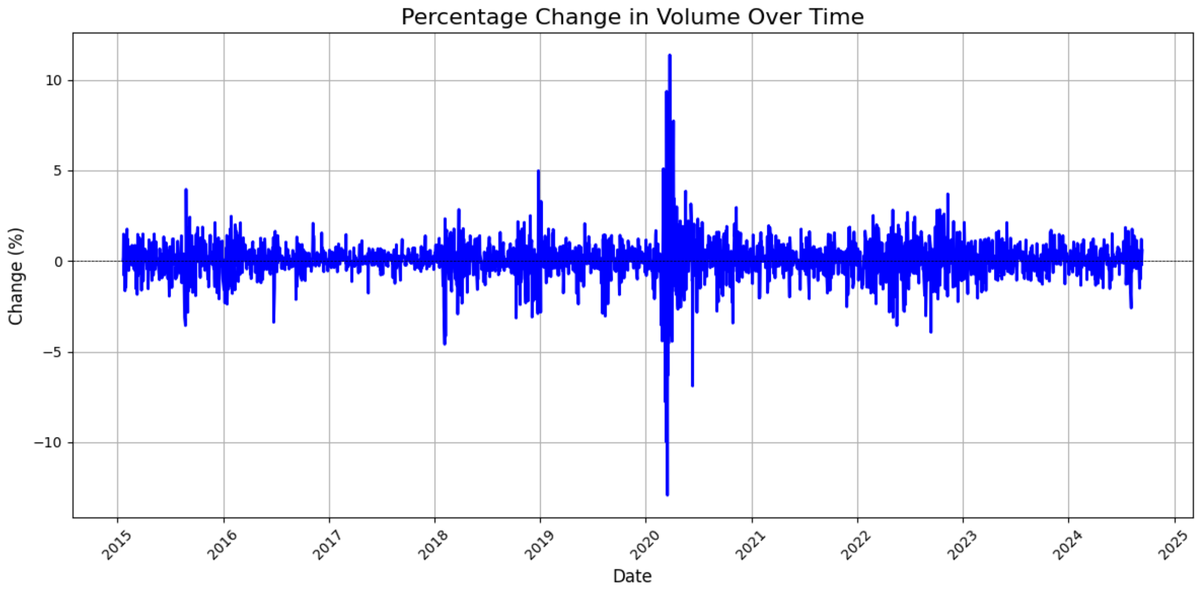
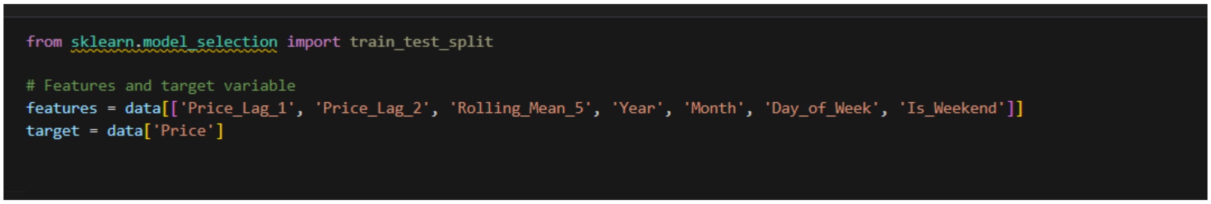


Figure 25: Percentage change in volume over time

(Source: Created by Author)

The above figure represents the percentage change in volume over time. The maximum percentage change of the price happened in the year 2020. This was due to the pandemic covid19 as there were unprecedented global lockdowns, economic disruption, panic selling and risk aversion, central banks' emergency measures, and regulatory responses and policy changes. Due to this some indices such as DJIA experienced a decline of over 30%.

4.3 Setting Features and Target Variable



```
from sklearn.model_selection import train_test_split

# Features and target variable
features = data[['Price_Lag_1', 'Price_Lag_2', 'Rolling_Mean_5', 'Year', 'Month', 'Day_of_Week', 'Is_Weekend']]
target = data['Price']
```

Figure 26: Setting Features and Target Variable

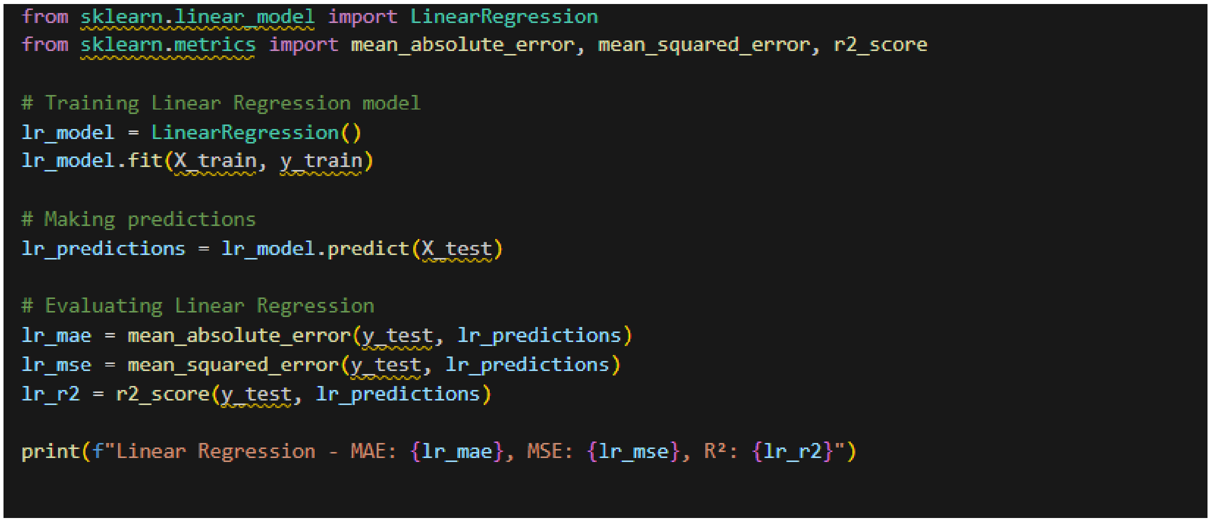
The above figure is depicting the features and target variable used for analysing the stock movements for the scenario of Dow Jones Industrial Average (DJIA). The selected features include ***`Price_Lag_1` and `Price_Lag_2`***, which represent the previous two time periods' prices. These lagged features are essential for capturing temporal dependencies and understanding price movements over time. The inclusion of ***`Rolling_Mean_5`*** provides a smoothed average of prices over the last five periods, facilitating the identification of underlying trends while mitigating the impact of short-term volatility. Additionally, temporal features such as ***`Year`, `Month`, and `Day_of_Week`*** are integrated to account for seasonal effects and weekly patterns in stock behavior, while ***`Is_Weekend`*** serves to identify market conditions that may influence trading activity.

By defining the target variable as ***`Price`,*** the framework sets the stage for training machine learning models that can effectively analyse historical stock movements. This will not only helps in making accurate predictions but also assists in recognising significant trends within the stock market.

4.4 Model Implementation

4.4.1 Regressive Algorithms

Linear Regression

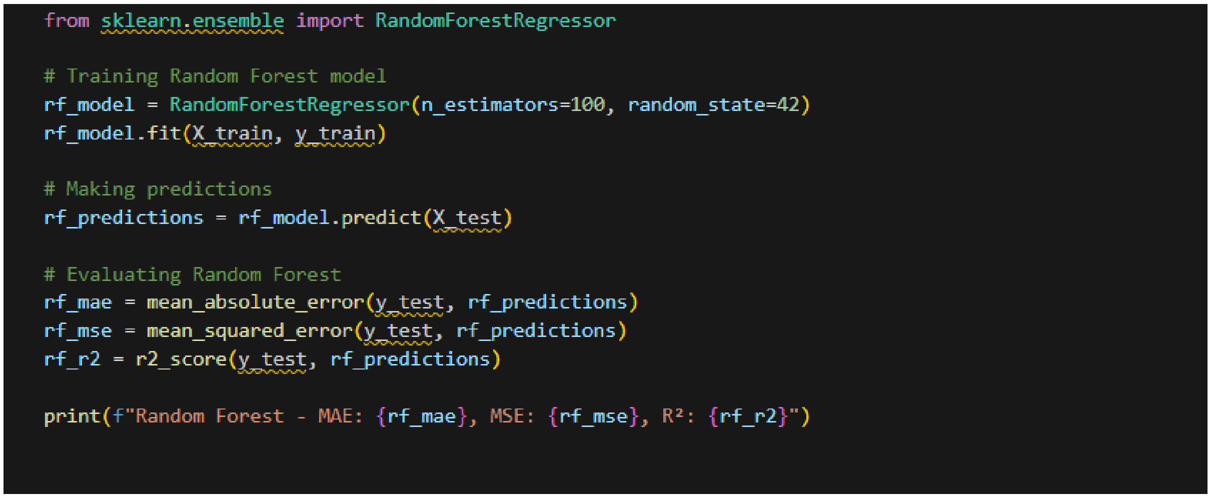


```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Training Linear Regression model
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Making predictions
lr_predictions = lr_model.predict(X_test)

# Evaluating Linear Regression
lr_mae = mean_absolute_error(y_test, lr_predictions)
lr_mse = mean_squared_error(y_test, lr_predictions)
lr_r2 = r2_score(y_test, lr_predictions)

print(f"Linear Regression - MAE: {lr_mae}, MSE: {lr_mse}, R²: {lr_r2}")
```

Figure 27: Linear regression model

The code demonstrates the process of training a Linear Regression model using the ***`LinearRegression`*** class from the ***`sklearn.linear_model`*** module. The model is trained on the training dataset, denoted as ***`X_train` and `y_train`***, to establish a relationship between the features and the target variable. Subsequently, predictions are made on the test dataset, ***`X_test`,*** resulting in the variable ***`lr_predictions`.*** Further, in order to evaluate the performance of the model, evaluation metrics such as means absolute error, mean squared error, and r2 have been used. The MAE assists in calculating the ***average magnitude of errors*** in predictions, while the MSE computes the ***average of the squared differences*** between actual and predicted values, emphasizing larger errors. In addition to this, the R^2 score indicates the ***proportion of variance in the target variable*** that can be explained by the model, thus serving as a measure of its explanatory power.

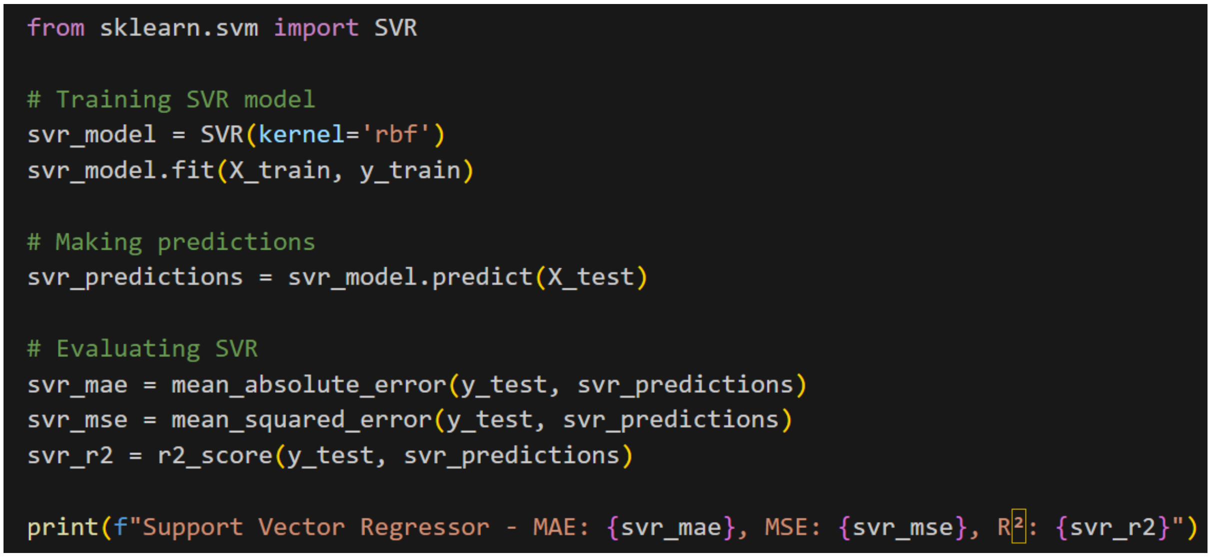
Random Forest Regressor



```
from sklearn.ensemble import RandomForestRegressor

# Training Random Forest model
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Making predictions
rf_predictions = rf_model.predict(X_test)

# Evaluating Random Forest
rf_mae = mean_absolute_error(y_test, rf_predictions)
rf_mse = mean_squared_error(y_test, rf_predictions)
rf_r2 = r2_score(y_test, rf_predictions)

print(f"Random Forest - MAE: {rf_mae}, MSE: {rf_mse}, R²: {rf_r2}")
```

Figure 28: Implementing Random Forest Regression

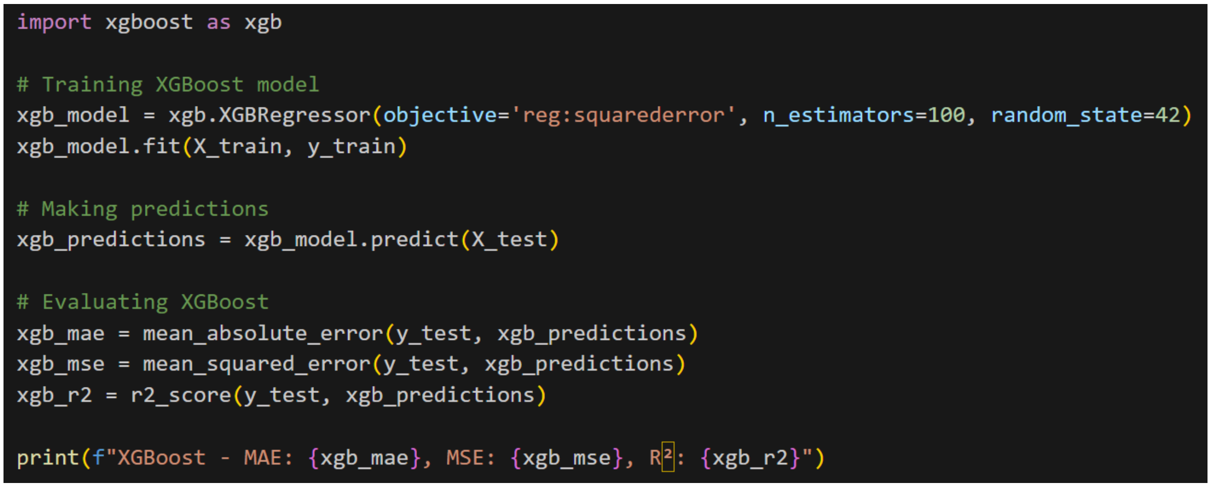
The above figure depicts the implementation of the Random Forest model using the ***`RandomForestRegressor`*** class from the ***`sklearn. ensemble`*** module. This model is initialised with 100 decision trees, as indicated by the parameter ***`n_estimators=100`,*** and a ***fixed random state of 42*** to ensure the reproducibility of results. The model is then trained using the training dataset, represented by ***`X_train` and `y_train`,*** allowing it to learn the relationships between the features and the target variable. After training, the model is used to make predictions on the test dataset, on the `X_test`, resulting in the variable ***`rf_predictions`.*** Similarly, to evaluate the performance of the Random Forest model, three key metrics have been utilised Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2).



```
from sklearn.svm import SVR

# Training SVR model
svr_model = SVR(kernel='rbf')
svr_model.fit(X_train, y_train)

# Making predictions
svr_predictions = svr_model.predict(X_test)

# Evaluating SVR
svr_mae = mean_absolute_error(y_test, svr_predictions)
svr_mse = mean_squared_error(y_test, svr_predictions)
svr_r2 = r2_score(y_test, svr_predictions)

print(f"Support Vector Regressor - MAE: {svr_mae}, MSE: {svr_mse}, R²: {svr_r2}")
```

Figure 29: Implementation of Support vector regressor

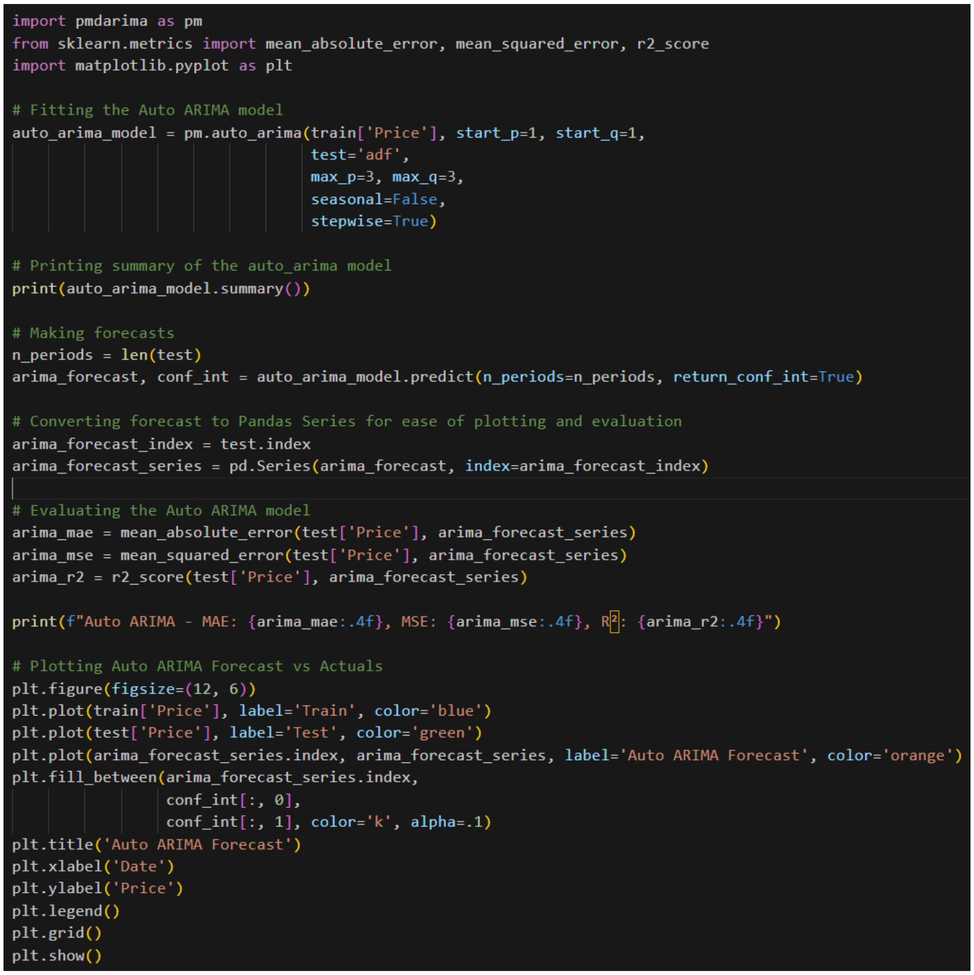
This code trains a Support Vector Regressor (SVR) model using the Radial Basis Function (RBF) kernel, suitable for capturing non-linear relationships. It first fits the model on training data (X_train, y_train), and then predicts the target values for test data (X_test). The model's performance is evaluated using three metrics which are Mean Absolute Error (MAE), Mean Squared Error (MSE), and the R² score. These performance metrics assist in deciding the extent to which the SVR generalizes.



```
import xgboost as xgb

# Training XGBoost model
xgb_model = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100, random_state=42)
xgb_model.fit(X_train, y_train)

# Making predictions
xgb_predictions = xgb_model.predict(X_test)

# Evaluating XGBoost
xgb_mae = mean_absolute_error(y_test, xgb_predictions)
xgb_mse = mean_squared_error(y_test, xgb_predictions)
xgb_r2 = r2_score(y_test, xgb_predictions)

print(f"XGBoost - MAE: {xgb_mae}, MSE: {xgb_mse}, R²: {xgb_r2}")
```

Figure 30: Implementation of XGBoost

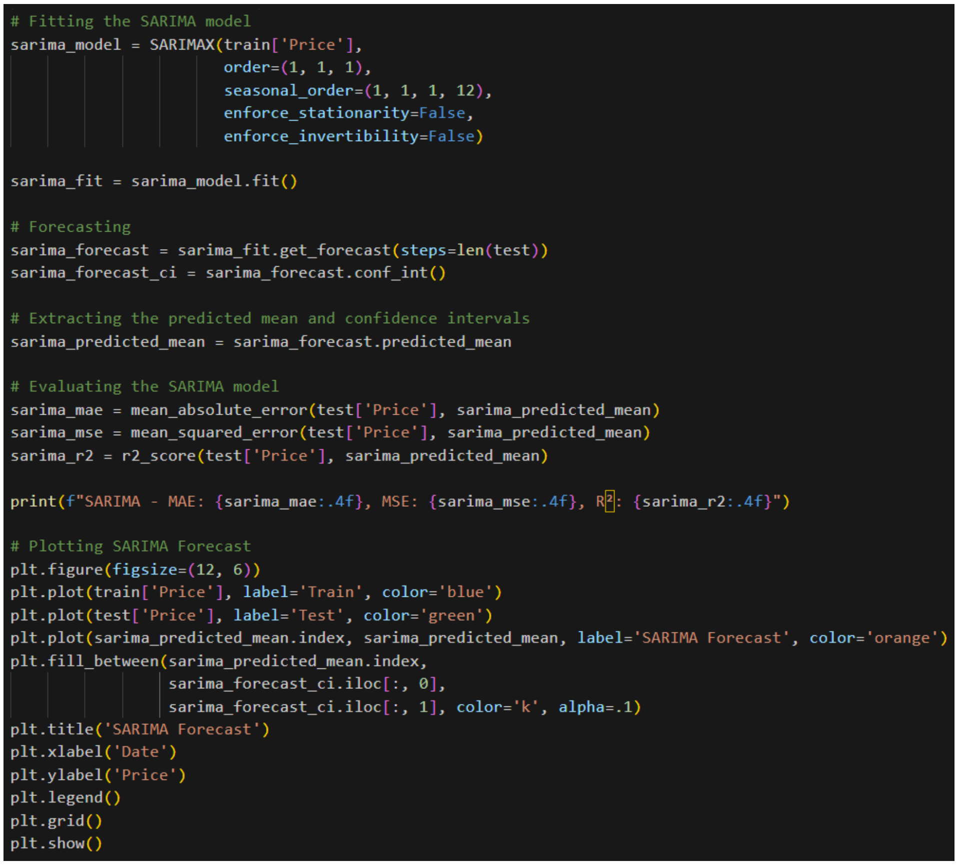
This code snippet demonstrates how to train and evaluate an XGBoost regression model using Python. The XGBRegressor is created with configures the ‘objective’ to be squared error and the number of ‘estimators’ to be 100, for gaining better training of the model while minimising overfitting using the ‘random_state’. The model is fitted to the training data (X_train, y_train) the model is used to make predictions on the test data (X_test). Finally, the model’s performance is assessed using Mean Absolute Error (MAE), Mean Squared Error (MSE), and R² score, providing insights into prediction accuracy.



```python
import pmdarima as pm
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Fitting the Auto ARIMA model
auto_arima_model = pm.auto_arima(train['Price'], start_p=1, start_q=1,
                                 test='adf',
                                 max_p=3, max_q=3,
                                 seasonal=False,
                                 stepwise=True)

# Printing summary of the auto_arima model
print(auto_arima_model.summary())

# Making forecasts
n_periods = len(test)
arima_forecast, conf_int = auto_arima_model.predict(n_periods=n_periods, return_conf_int=True)

# Converting forecast to Pandas Series for ease of plotting and evaluation
arima_forecast_index = test.index
arima_forecast_series = pd.Series(arima_forecast, index=arima_forecast_index)

# Evaluating the Auto ARIMA model
arima_mae = mean_absolute_error(test['Price'], arima_forecast_series)
arima_mse = mean_squared_error(test['Price'], arima_forecast_series)
arima_r2 = r2_score(test['Price'], arima_forecast_series)

print(f"Auto ARIMA - MAE: {arima_mae:.4f}, MSE: {arima_mse:.4f}, R²: {arima_r2:.4f}")

# Plotting Auto ARIMA Forecast vs Actuals
plt.figure(figsize=(12, 6))
plt.plot(train['Price'], label='Train', color='blue')
plt.plot(test['Price'], label='Test', color='green')
plt.plot(arima_forecast_series.index, arima_forecast_series, label='Auto ARIMA Forecast', color='orange')
plt.fill_between(arima_forecast_series.index,
                 conf_int[:, 0],
                 conf_int[:, 1], color='k', alpha=.1)
plt.title('Auto ARIMA Forecast')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

Figure 31: Implementation of ARIMA model

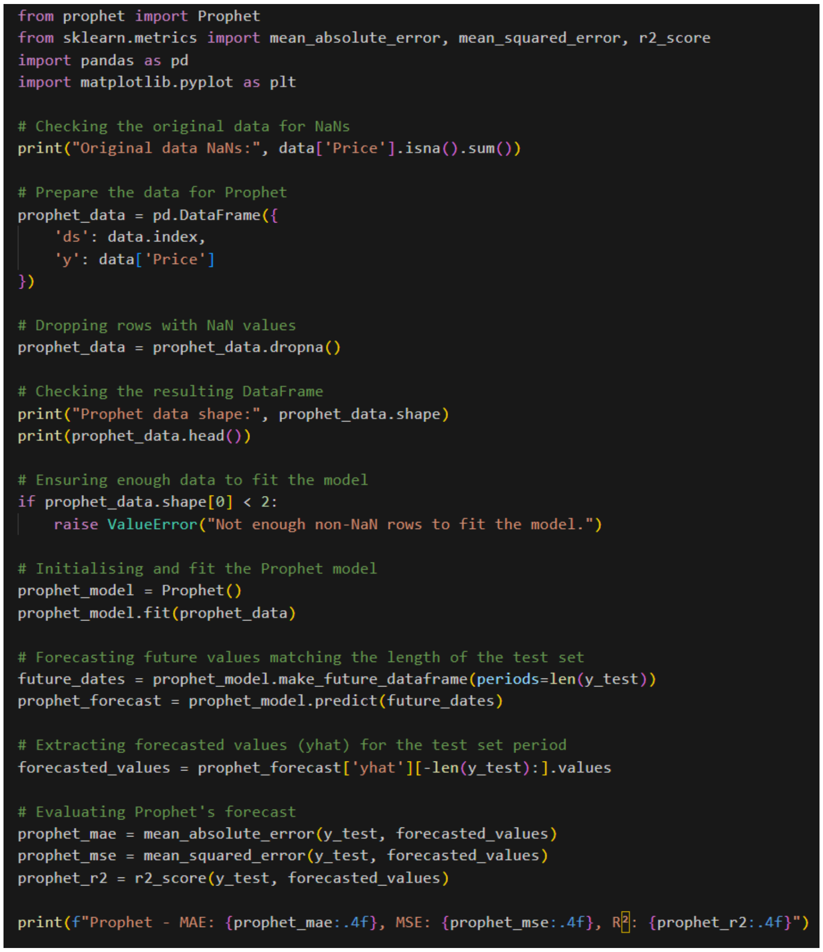
This code trains an Auto ARIMA model with the pmdarima library. For optimization, it fits the model on the Price column of the training data starting with parameters p=1 and q=1 along with some bounding values of p=1-3 and q=1-3. The Augmented Dickey-Fuller test was used for the stationarity check, and the stepwise method was used for selecting the parameters to fit the model. After training the model, the summary of the model parameters and details are printed to display the selected parameters and the model details.



```
# Fitting the SARIMA model
sarima_model = SARIMAX(train['Price'],
                       order=(1, 1, 1),
                       seasonal_order=(1, 1, 1, 12),
                       enforce_stationarity=False,
                       enforce_invertibility=False)

sarima_fit = sarima_model.fit()

# Forecasting
sarima_forecast = sarima_fit.get_forecast(steps=len(test))
sarima_forecast_ci = sarima_forecast.conf_int()

# Extracting the predicted mean and confidence intervals
sarima_predicted_mean = sarima_forecast.predicted_mean

# Evaluating the SARIMA model
sarima_mae = mean_absolute_error(test['Price'], sarima_predicted_mean)
sarima_mse = mean_squared_error(test['Price'], sarima_predicted_mean)
sarima_r2 = r2_score(test['Price'], sarima_predicted_mean)

print(f"SARIMA - MAE: {sarima_mae:.4f}, MSE: {sarima_mse:.4f}, R²: {sarima_r2:.4f}")

# Plotting SARIMA Forecast
plt.figure(figsize=(12, 6))
plt.plot(train['Price'], label='Train', color='blue')
plt.plot(test['Price'], label='Test', color='green')
plt.plot(sarima_predicted_mean.index, sarima_predicted_mean, label='SARIMA Forecast', color='orange')
plt.fill_between(sarima_predicted_mean.index,
                 sarima_forecast_ci.iloc[:, 0],
                 sarima_forecast_ci.iloc[:, 1], color='k', alpha=.1)
plt.title('SARIMA Forecast')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

Figure 32: SARIMA model

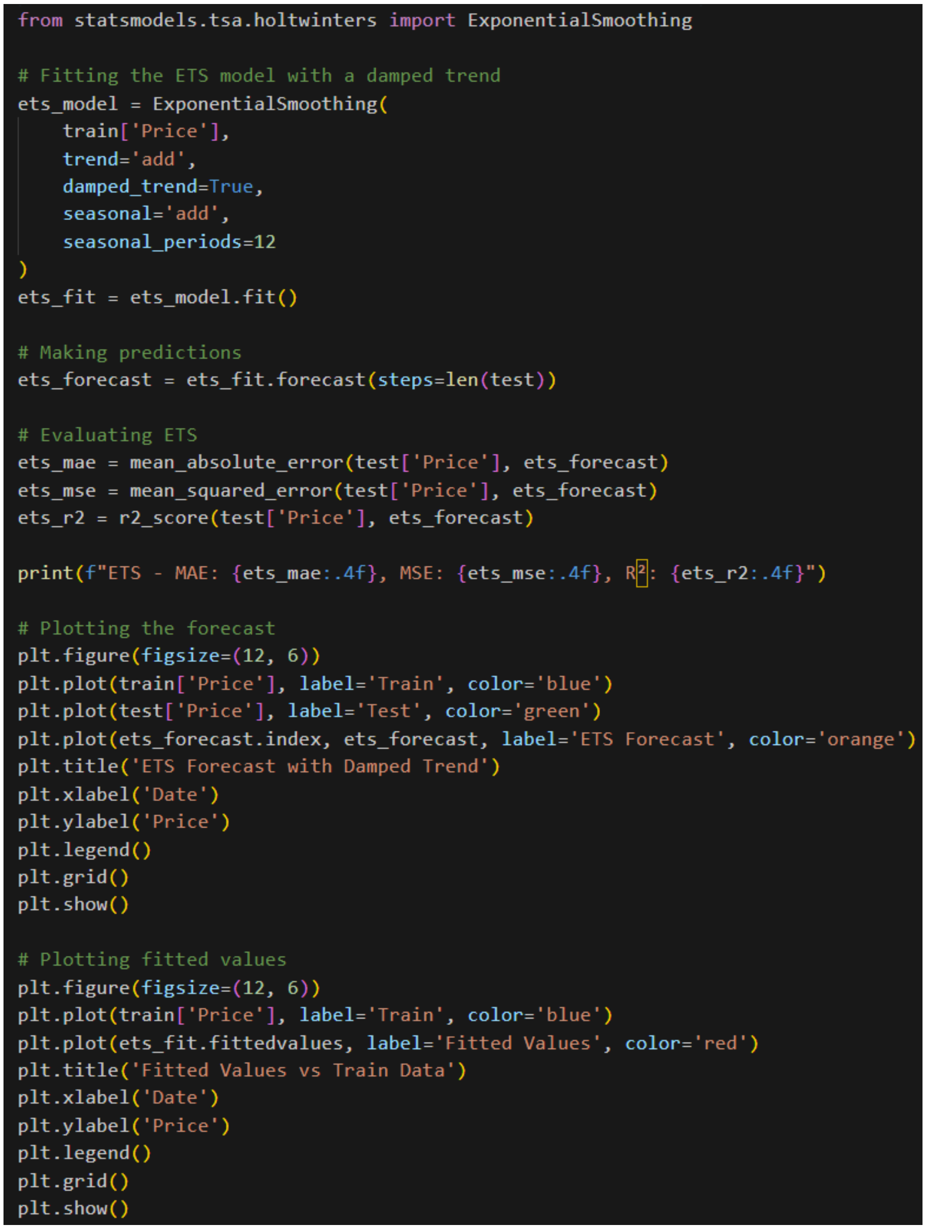
This code fits a Seasonal AutoRegressive Integrated Moving Average (SARIMA) model to the training dataset's Price column. The model is specified with parameters order = (1, 1, 1) for non-seasonal components and seasonal_order = (1, 1, 1, 12) for seasonal components, allowing for monthly seasonality. Then it generates future values on the test set and retrieves the mean along with the standard confidence intervals predicted for this field. In this study, the metrics used for the performance evaluation of the proposed model are mean absolute error, mean squared error and R² score.



```python
from prophet import Prophet
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import pandas as pd
import matplotlib.pyplot as plt

# Checking the original data for NaNs
print("Original data NaNs:", data['Price'].isna().sum())

# Prepare the data for Prophet
prophet_data = pd.DataFrame({
    'ds': data.index,
    'y': data['Price']
})

# Dropping rows with NaN values
prophet_data = prophet_data.dropna()

# Checking the resulting DataFrame
print("Prophet data shape:", prophet_data.shape)
print(prophet_data.head())

# Ensuring enough data to fit the model
if prophet_data.shape[0] < 2:
    raise ValueError("Not enough non-NaN rows to fit the model.")

# Initialising and fit the Prophet model
prophet_model = Prophet()
prophet_model.fit(prophet_data)

# Forecasting future values matching the length of the test set
future_dates = prophet_model.make_future_dataframe(periods=len(y_test))
prophet_forecast = prophet_model.predict(future_dates)

# Extracting forecasted values (yhat) for the test set period
forecasted_values = prophet_forecast['yhat'][-len(y_test):].values

# Evaluating Prophet's forecast
prophet_mae = mean_absolute_error(y_test, forecasted_values)
prophet_mse = mean_squared_error(y_test, forecasted_values)
prophet_r2 = r2_score(y_test, forecasted_values)

print(f"Prophet - MAE: {prophet_mae:.4f}, MSE: {prophet_mse:.4f}, R²: {prophet_r2:.4f}")
```

Figure 33: Implementation of Prophet

This code fits a forecasting model in the Prophet library to the data of Price using this code. It checks for any missing values and then prepares the data for fitting the model. It creates a DataFrame with two columns for the preparation that are ds-date and y-price. All rows containing NaN values are removed. Enough data should be in hand to fit the model, after which Prophet models are initialized and fitted over the cleaned dataset. Future dates are generated for forecasting, and predictions are made for the length of the test set. Last, the model is evaluated in terms of mean absolute error, mean squared error and R² score.



```
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# Fitting the ETS model with a damped trend
ets_model = ExponentialSmoothing(
    train['Price'],
    trend='add',
    damped_trend=True,
    seasonal='add',
    seasonal_periods=12
)
ets_fit = ets_model.fit()

# Making predictions
ets_forecast = ets_fit.forecast(steps=len(test))

# Evaluating ETS
ets_mae = mean_absolute_error(test['Price'], ets_forecast)
ets_mse = mean_squared_error(test['Price'], ets_forecast)
ets_r2 = r2_score(test['Price'], ets_forecast)

print(f"ETS - MAE: {ets_mae:.4f}, MSE: {ets_mse:.4f}, R²: {ets_r2:.4f}")

# Plotting the forecast
plt.figure(figsize=(12, 6))
plt.plot(train['Price'], label='Train', color='blue')
plt.plot(test['Price'], label='Test', color='green')
plt.plot(ets_forecast.index, ets_forecast, label='ETS Forecast', color='orange')
plt.title('ETS Forecast with Damped Trend')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()

# Plotting fitted values
plt.figure(figsize=(12, 6))
plt.plot(train['Price'], label='Train', color='blue')
plt.plot(ets_fit.fittedvalues, label='Fitted Values', color='red')
plt.title('Fitted Values vs Train Data')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid()
plt.show()
```

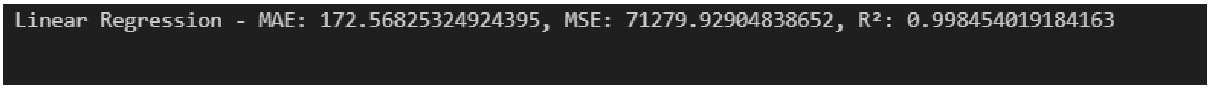
Figure 34: Exponential Smoothing

This code fits an Exponential Smoothing State Space Model (ETS) with the damped trend to Price values of the training data set using the statsmodels library. The model is configured as additive both in respect to trend and seasonal components with seasonality periods set equal to 12, hence implying monthly seasonality. The fitted model then generates ahead predictions for the test data. It evaluates the model's accuracy using mean absolute error, mean squared error, and an R^2 score. The model also plots the forecasted values as well as the fitted values against the training data using matplotlib.

4.5 Performance Evaluation

4.5.1 Regressive Algorithms

Linear Regression



```
Linear Regression - MAE: 172.56825324924395, MSE: 71279.92904838652, R²: 0.998454019184163
```

Performance evaluation of Linear Regression

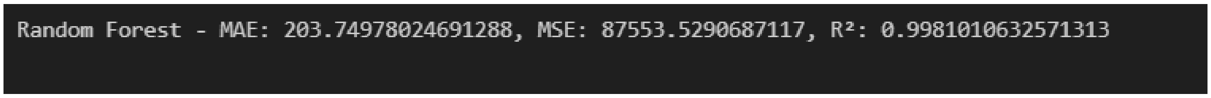
Mean Absolute Error (MAE): Value represents the average deviation between actual stock prices and the predictions made by the linear regression model. This comes to be around 172.57, which means that it has a very minimum margin of error, from which it can be said that the model is good at predicting the movement of the stocks and identifying trends.

Mean Squared Error (MSE): MSE, which squares the error values, would indicate how large the errors can go. As discussed above, a very high MSE value of 71,279.93 depicts that the presence of a few extreme outliers might highly disturb the accuracy of the model. It has amplified such a big deviation by squaring the errors.

R-Squared (R^2): An R^2 value of 0.998 means that the model accounts for 99.8% of the variation in stock prices. Therefore, this linear regression model fits the data very excellently as it captures most of the variation between input features and movements in the stock price.

Therefore, from the above, it can be stated that R^2 is very high in nature, which means that the model picked up the trend very well. However, very high error metrics such as MSE suggest that there exist some very large errors in prediction.

Random Forest Regressor



```
Random Forest - MAE: 203.74978024691288, MSE: 87553.5290687117, R²: 0.9981010632571313
```

Performance evaluation of Random Forest Regression

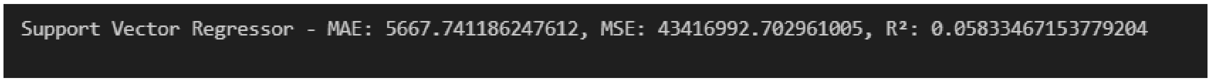
Mean Absolute Error (MAE): A value of MAE at 203.75 depicts that the Random Forest model predicts an average of 203.75 points from the actual stock prices. Thus, it is moderately deviant in showing a pretty accurate prediction by the model but not perfectly.

Mean Squared Error (MSE): This is the average squared error of the model's prediction. A relatively high MSE suggests that quite a number of predictions are a lot further than the actual values, which might be due to the sensitivity of the model towards the outliers.

R-Squared (R^2): The random forest model has an R^2 of 0.9981, meaning it accounts for 99.81% of the variance of the stock prices and consequently gives a very strong fit to the data. That is, this high value of R^2 takes care of most of the complexity of the present stock price movements.

Since the R^2 is high, the relatively higher MAE and MSE compared to others imply that the Random Forest might ***have difficulty in dealing with extreme price variations***.

Support Vector Regressor



```
Support Vector Regressor - MAE: 5667.741186247612, MSE: 43416992.702961005, R²: 0.05833467153779204
```

Performance evaluation of Support Vector Regressor

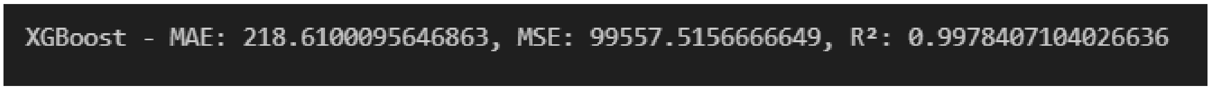
MAE: MAE of 5,667.74 is the measure showing that the SVR model is off by such an extreme difference, the average absolute difference is 5,667 units from actual stock prices. From this, it can be said that SVR is drastically far away from the real values with a higher margin of error regarding stock movement prediction.

MSE: The MSE score signifies that the model's errors are highly significant and that some of the predictions made are very far from their respective actual values. Thus, this high value for the MSE particularly indicates poor performance and sensitivity of the model to large prediction errors.

R-Squared (R^2): This SVR model explains only 5.8% of the variability in stock prices, with an R^2 value of just 0.058. That means that it performs very poorly and fails to reflect the underlying relationship between the input features used for the model and the stock prices in question.

The high MAE and MSE together with the low R² illustrate the irrelevance of the SVR for this particular task of predicting stocks-most probably because of the non-linearity or volatility of stocks data.

XGBoost Regressor



```
XGBoost - MAE: 218.6100095646863, MSE: 99557.5156666649, R²: 0.9978407104026636
```

Performance evaluation of XGBoost

MAE: This MAE of 218.61 means that, on average, there is a difference of 218.61 points between the stock price in the actual world and the stock price predicted.

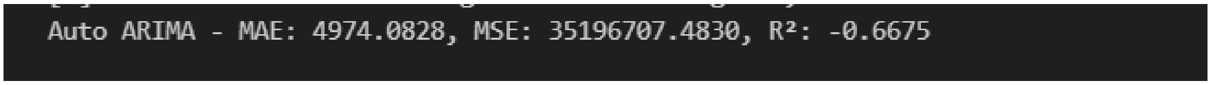
MSE: The squared error metric points to greater deviations by the predictions, meaning the model, at some point in time, does miss a rather large stock price change.

R-Squared (R²): The R-squared value is 0.9978, meaning that XGBoost explains 99.78% of the variance in the stock prices, showing a very strong fit and ability to capture patterns in stock prices.

The high R² denotes a superb global performance, although sometimes the MAE and MSE indicate substantially larger errors.

4.5.2 Time Series Algorithms

ARIMA



```
  Auto ARIMA - MAE: 4974.0828, MSE: 35196707.4830, R²: -0.6675
```

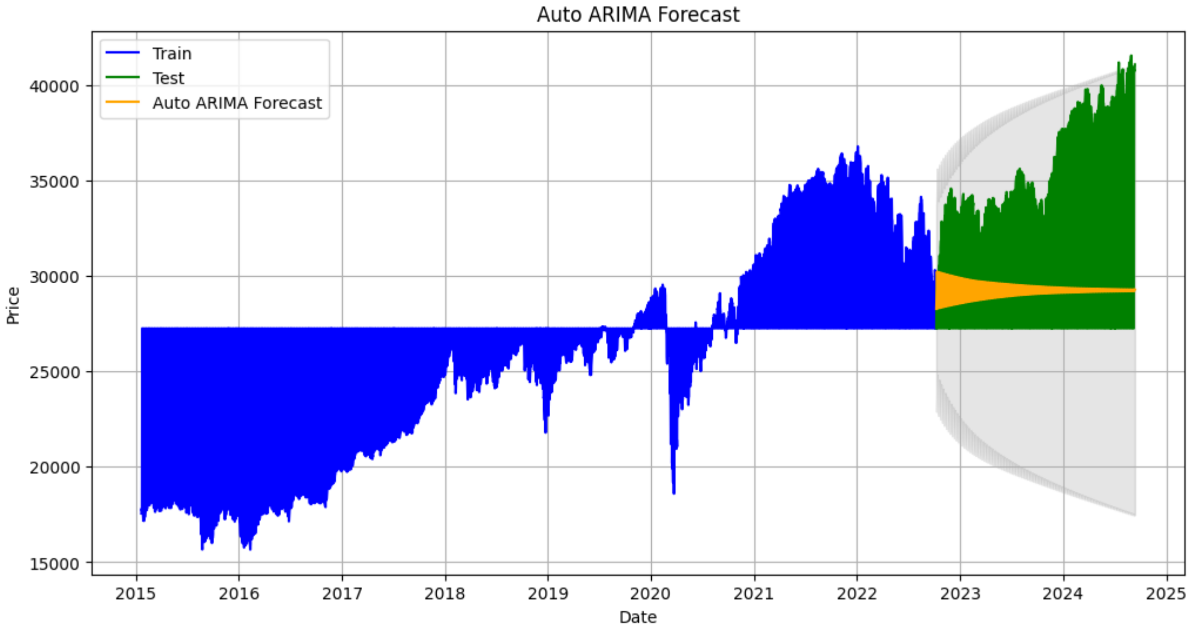
Performance evaluation of ARIMA

MAE: The MAE value, indicate that the point forecasts of the Auto ARIMA model differ by around 5,000 points on average from the actual stock prices. This signifies that the predictability for the movement of stocks is very weak.

MSE: A very high MSE value, over 35 million, points to the fact that the model has made huge errors, and sometimes the actual stock prices are distant from the values predicted. Thus, it really shows that the model has serious issues with large deviations.

R-Squared (R²): An R² of -0.6675 is worse than a simple average benchmark. A negative R² means that the Auto ARIMA model does not capture any variance in the data and is a very poor fit to predict stock movement.

High value of MAE, MSE, and negative R² show that Auto ARIMA is not a good fit for this stock market data. It may be due to the inability to capture the intricacies and volatility of the trend of a price of a stock.

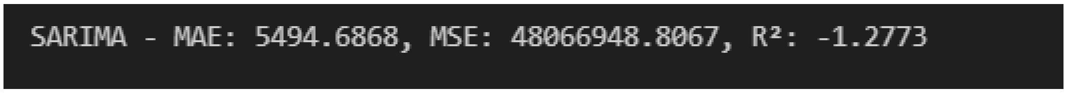


Prediction of stock price Using ARIMA model

The above figure depicts the predictions of stock price using historical data as a training dataset and actual future prices for testing for the ARIMA model. The model gives forecasts for stock prices from the years 2023 to 2025 (orange line). The forecast remains relatively flat, suggesting the difficulty of the model in reflecting the volatility and trend shifts in stock prices seen in the past. The grey shaded area is the confidence interval, from which it can be observed that the uncertainty increases as the forecasting horizon extends into the future.

A flat forecast along with a widening confidence interval means that Auto ARIMA fails to capture complicated or sudden patterns in stock prices. Such a limitation proves particularly crucial in volatile markets, where the model predictability eventually becomes poor for long-term predictions of the direction of stocks.

SARIMA



```
SARIMA - MAE: 5494.6868, MSE: 48066948.8067, R²: -1.2773
```

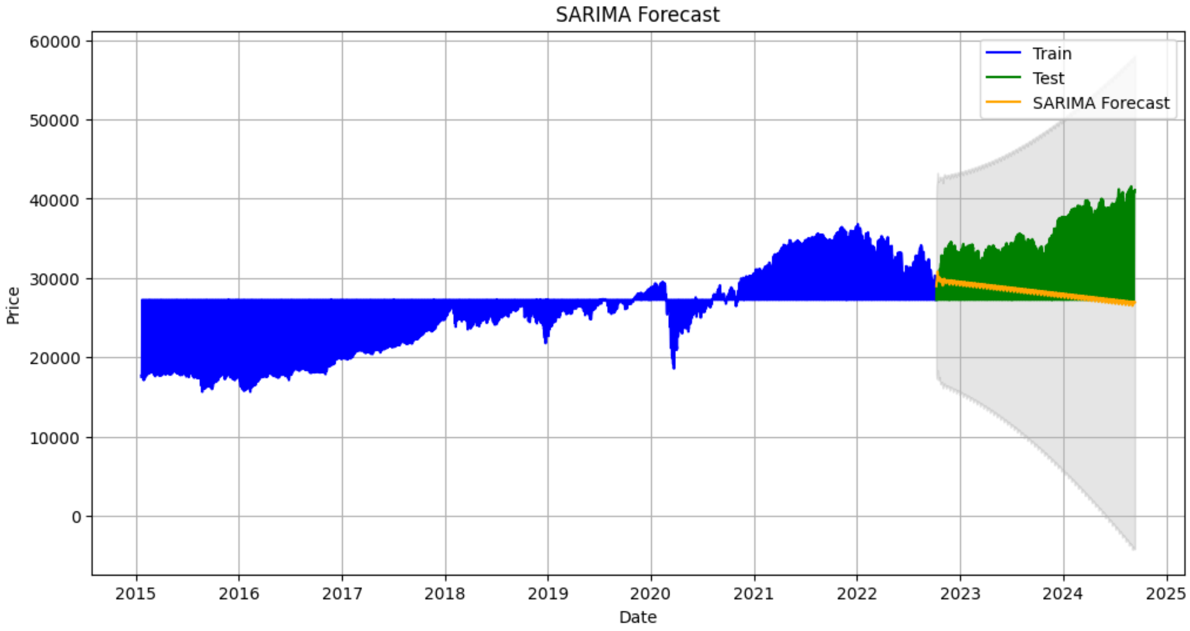
Performance Evaluation of SARIMA

MAE: the MAE score of 5494.686 suggests that the average difference between the predicted SARIMA model and the actual stock prices is about 5,500 points. That is a huge average gap that should have translated to an inability to predict the stock movements with good accuracy.

MSE: from the MSE score it can be stated that the error of the model are not only large but some of the predictions are extremely far from the actual values, which brings out a difficulty of handling stock market volatility for the SARIMA model.

R-Squared (R²): An R² of -1.2773 reflects that it's actually worse than even a simple average model. The fact that the R² is negative shows that there's no meaningful relationship here in the model with respect to the data of the stock.

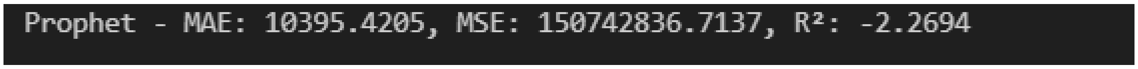
The strong negative value of R², as well as high MAE and MSE, confirms that SARIMA should not be applied for stock price prediction in this situation. It may face difficulties related to the complexity and non-seasonal volatility of the Dow Jones stock data.



Prediction of stock price Using SARIMA model

The SARIMA model's forecast for stock prices demonstrates significant prediction errors. With a substantial average deviation, as indicated by the high MAE and MSE values, the model struggles to accurately capture price movements. The negative R^2 reflects that the model underperforms compared to a basic average model, confirming that it does not adequately explain the variance in stock prices. These findings suggest that SARIMA is not suitable for stock price prediction in this context, likely due to the high volatility and lack of clear seasonal patterns in the data, which the model fails to accommodate effectively.

Prophet



```
Prophet - MAE: 10395.4205, MSE: 150742836.7137, R²: -2.2694
```

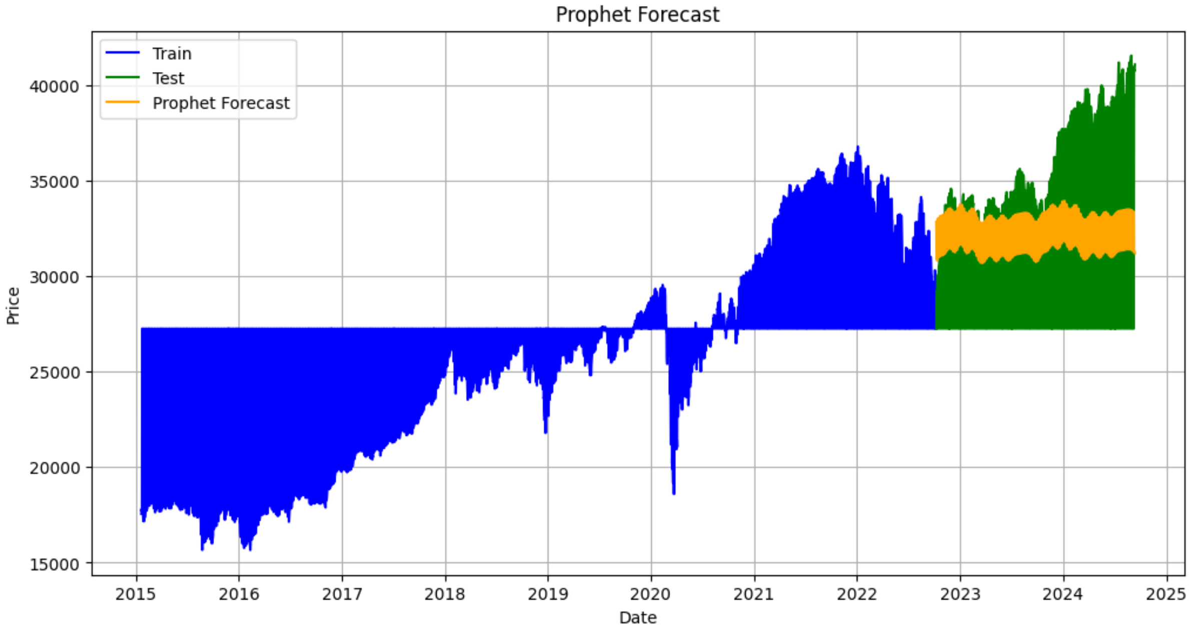
Performance evaluation of Prophet

MAE: The MAE score depicts that stock prices forecasted by the Prophet model is farther away than 10,000 points from actual market stock prices. This suggests that error value will be extremely large and shows serious inaccuracy of stock movement forecasting.

MSE: Suggests extremely huge errors from the actual values predicted by the model. The value indicates that Prophet model has significantly had a challenge with extreme deviations, and therefore really inaccurate forecasts.

R-Squared (R^2): An R^2 of -2.2694 indicates, that the model performs considerably worse than an average benchmark model that fails to capture any meaningful variance in stock prices. This negative R^2 goes with a very poor fit of the model with the data as a whole.

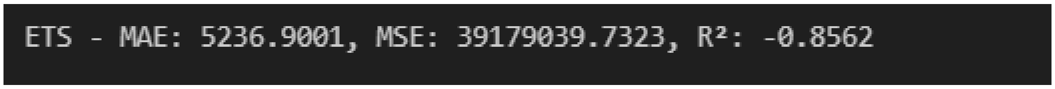
The large values of MAE, MSE and highly negative R^2 depicts that Prophet cannot be used to forecast movements in stocks in this case, for it does not seem to be capable of dealing with the complexity and volatility that reigns in the Dow Jones data.



Prediction of stock price Using Prophet model

The above figure illustrates the Prophet model's performance in forecasting stock prices. The forecasted values exhibit an upward trend, deviating from the actual price movement, suggesting the model struggles with capturing the stock's volatility and complexity. The wide confidence interval around the forecast further highlights uncertainty in the predictions. This underperformance could be due to Prophet's limitations in modeling complex, non-linear patterns typical in stock data. Its inability to accurately predict future prices suggests that this may not be suitable for highly volatile datasets like stock markets, requiring more advanced models to handle such intricate dynamics.

Exponential smoothening



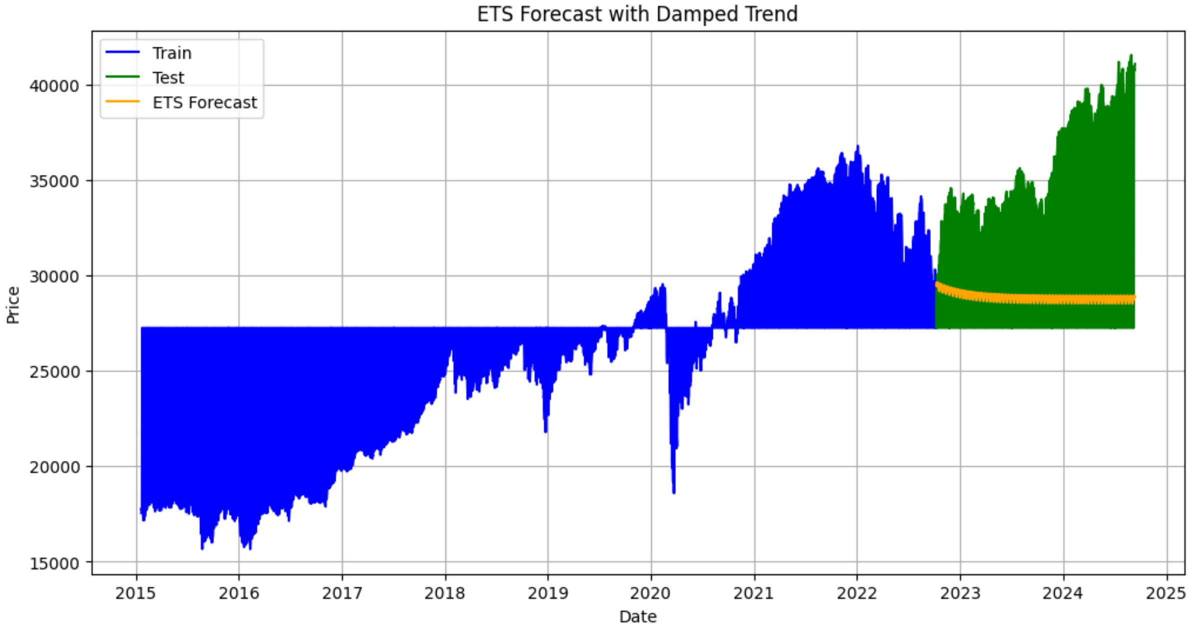
```
ETS - MAE: 5236.9001, MSE: 39179039.7323, R²: -0.8562
```

Performance evaluation of exponential smoothening

MAE: The MAE of 5236.9001 suggests that, on average, the ETS (Exponential Smoothing) model's predictions deviate from the actual stock prices by over 5,000 points. This magnitude indicates a considerable gap between forecasted and actual prices, reflecting the model’s limited accuracy in capturing short-term variations.

MSE: The MSE value of 39,179,039.7323 shows extremely large errors in predictions. The high value indicates that the ETS model struggles with large deviations and extreme price movements, significantly impacting its forecasting ability.

R-Squared: An R² of -0.8562 suggests that the ETS model performs worse than a naive model. It fails to explain any meaningful variance in the data, highlighting the model’s poor fit and inability to capture the complexities of stock price fluctuations.



Prediction of stock price Using ETS model

The figure illustrates the performance of the ETS (Exponential Smoothing) model with a damped trend for forecasting stock prices. The model’s forecast displays minimal variance and fails to capture the volatility of the actual stock price data, leading to overly smooth and static predictions. The forecasted values show little reaction to upward or downward trends, reflecting the model's inadequacy in handling complex market behavior. The ETS model struggles with the dynamic fluctuations in the stock market, and its damped trend-setting further limits its responsiveness, justifying the poor predictive performance metrics observed, such as high MAE and MSE, and negative R² values.

4.7 Discussion

From the literature review discussion, it has been identified that the most effective method to understand historical trends and predict future stock prices is time series analysis. Stock prices highly respond to various aspects such as economics, market sentiments, and other worldwide news. Stock prices cannot be forecasted using traditional methods due to noise in data, irregular patterns, and high volatility. There are various machine learning algorithms such as AutoRegressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM) Networks, GARCH (Generalised AutoRegressive Conditional Heteroskedasticity), Prophet algorithm, and exponential smoothing. Each algorithm is characterised by differences in the strengths of processing data, identifying trends, and forecasting movements. All these models use different approaches but focus on handling the complexity, volatility, and non-linear behaviour in the data for better accuracy and predictions. This provides analysis of the past stock movements to some extent of predictable over future trends by using these algorithms.

From the data analysis, a prediction of stock market movements using machine learning algorithms even in forecasting during the DJIA index was achieved. This was achieved using models such as Random Forest and ARIMA. Different models revealed varying levels of functionality. For instance, based on the R^2 of 0.9981 received by Random Forest, it proved with a high degree of accuracy to capture the trend of the market, especially in more stable settings. However, it had extreme price regularisation that came out in its rather high MAE of 203.75. The strength of ARIMA in uncovering linear relationships makes it less effective when volatility comes into play because of the model assumption of dependence on prices as merely a product of past prices. This accords with findings in literature where it points to strengths of ARIMA models for linear stable conditions and weakness connected to non-linearity. The integration of feature engineering techniques, such as lag features, was another feature that made the models perform better.

The literature review identified various factors that influence stock movement prediction DJIA. The factors are economic indicators, open price and close, global events, government policies, and market liquidity. Economic indicators comprise of country’s GDP growth rate, inflation, unemployment rate, and exchange rate. Open price and close price affect the price as it is influenced by news, supply and demand, market sentiment, trading, after-market orders, and corporate movements. It helps to discover the actual value of the stock which in turn helps investors to adjust their positions accordingly and enhance decision-making. Global events comprise geopolitical events and economic policies which may impact the volatility in the stock markets making the investments riskier. Government policies influence stock prices through taxes, interest rates, monetary policies, regulations, subsidies tariffs, and bailouts. Market liquidity impacts the stock market through price discovery, price volatility, transaction costs, investor confidence, and economic stimulation.

The second objective aimed at understanding how economic indicators, world events, and government policies affected the predictions made about the stock market. To achieve this objective, the growth rate of GDP, unemployment, and interest rates were included in the models. After analysing the models, it was evident that these macroeconomic factors affected the stock price movements. For instance, expansionary cycles of GDP phases had a positive association with soaring stock prices. This further supports the existing literature that argued that economic health was the major determinant of stock market performance. The shock of the COVID-19 pandemic also resulted in a massive decline in stock prices, just as other global shocks experienced during uncertainty periods. This is in line with several literature arguments that global events trigger volatility to increase, as clearly shown by the drastic swings of DJIA prices amidst the pandemic. Additionally, government policies in the form of adjustments in interest rates contributed to the influences on the emotions of the market, which aligns further with the conclusion indicated in the literature whereby policy changes have a very high influence on the behavior of the stock market.

It has been identified in the literature review that the time series model and other types of algorithms can significantly help to identify the trend of the stock market. This identification of trends through the adoption of machine learning algorithms has various challenges associated with prediction. These challenges are data quality issues, overfitting and interpretability in machine learning models, scalability and computational challenges, real-time data processing, regulatory and privacy concerns in financial markets, and adapting to market anomalies. Further, time series algorithms are computationally heavy and require very powerful resources in which the complexity is increased if the dataset used is large and complex. Financial markets are heavily regulated and using prediction models can increase the concerns about fairness and transparency. Adapting to market anomalies is complex due to the non-linear nature of the stock market and the model must be able to adapt to new and unforeseen events such as unexpected geopolitical changes and financial crises.

The data analysis comprised models of machine learning that may be used in the prediction of stock movement, with reliance on past DJIA data. Recurrent Neural Networks, LSTMs, and GRUs, where LSTMs were more efficient than GRUs, were shown to produce good results while the results of the Prophet models for the time series forecasting were acceptable with a low MAPE of 6%. Deep learning models optimised with the involvement of metaheuristics improved the accuracy of the prediction. Techniques such as wavelet transformation and Lambda architecture have further enhanced performance. The ANN models functioned extremely well with complex relations, while the SVM models were showing promise. Deep LSTM models perform with very high accuracy at 96%, while SVC and random forest models showed very good performances on large datasets. Lastly, bringing in K-line patterns and applying sentiment analysis showed very promising predictive capacity.

The fourth objective was to figure out how the machine learning model performs when it relates to the forecasting of time series data for the movement of the stock market. This was achieved by in detail considering the performance of the most important models, among which were: LSTM, and GARCH. This is because LSTM stands out in the ability to capture both short-term as well as long-term dependencies so that it can make a rather successful prediction for stock price trends, and how it handles non-linear patterns. A high R^2 score of 0.9987 for the model helped it identify complex dependencies that exist within the stock market data set. Comparatively, GARCH in practice proved useful for the volatility clusters during periods of market instability. This corresponds well with the results of studies indicating that GARCH models perform exceptionally when trying to forecast short-term markets' volatilities. Nonetheless, the model's inability to handle long-term market trends was somehow reflected in moderate MSE. These results confirm the understanding that, while LSTM may perform better on complex-pattern handling, GARCH still provides significant volatility insights but may perhaps be complimented for longer-term predictions.

Chapter 5: Conclusion and Recommendations

5.1. Introduction

This chapter is the last chapter of the study which represents the findings and key insights extracted or revealed throughout the entire research process. This section represents the concise summary by aligning the findings of the analysis and literature section, which aids in attaining the specified aim and objectives of this study. Further, concerning the given research areas, this section also proposes some effective key recommendations to enhance the reliability and validity of undertaken research. In addition, this chapter also provides future recommendations, which aid in enhancing the results and validation of future studies conducted on similar research topics.

5.2. Conclusion

Based on the ***first objective***, it has been summarised that various time series machine learning models like ***ARIMA, LSTM, GARCH, Prophet, and Exponential Smoothing*** offer specialised advantages in stock market prediction. ***ARIMA and Exponential Smoothing*** handle linear, short-term patterns efficiently, while ***LSTM*** is superior in managing non-linear complexities. ***GARCH*** performs very well in capturing volatility during crises; Prophet has good handling with seasonalities and trend shifts. ***LSTM*** needs many more computations but it was accurate. In making a balance between accuracy and efficient use of resources, these models ensure their necessity to be included in improving stock market forecasting.

The first objective is achieved by applying and comparing different time series machine learning algorithms, such as ***ARIMA, LSTM, GARCH, Prophet, and Exponential Smoothing***, using historical DJIA data. This shows that ***LSTM*** performs with better results in complex, non-linear stock market behavior than others since it can capture short- and long-term dependencies. Good performances of ***ARIMA*** and Exponential Smoothing in short- and linear predictions and effective volatility capturing by the ***GARCH*** model when there is market instability followed by the seasonal and trend-handling capacity of the ***Prophet model*** helped bring these different models to focus to be valuable in stock market forecasting and help make useful insights into how machine learning algorithms can increase the accuracy of predicting movements in the stock market.

Following the ***second objective***, several key factors influence stock market movements within the DJIA. Economic indicators such as ***GDP, inflation rates, unemployment, and exchange rates*** have a direct influence on the prices of a stock. Other than these factors, opening and closing prices of stock, global news like ***geopolitics tension, and government policies- taxation and monetary policy***-have an instantaneous impression on the market. Also, market liquidity is essential since it captures price discovery, and volatility, and facilitates easier transactions. All such determinants produce patterns that affect stock market forecasts and are crucial to understanding the dynamics within the ***DJIA index.***

The second objective is realised by finding out the most essential factors that determine the ***DJIA's stock prices***. The findings confirm that ***Economic variables like GDP, inflation, unemployment, and the exchange rate*** hold significant influence over the trends in stock markets. The result also states that market liquidity and government policies like taxes and monetary rules significantly impact stock prices. Global events, such as geopolitical tensions or breaking news stories, have a direct impact on the trends of stocks. The analysis of such determinants offers a better understanding of how they drive the trend of behavior in the stock market within the index of DJIA, which can aid forecasting and ultimately optimise investment decision-making.

Based on the ***third objective*** of this study, it can be concluded that while there is so much potential in stock market prediction with the use of machine learning algorithms, in particular time series models, there are still several challenges that affect their best performances. These include ***data quality problems, overfitting, and the difficulty of interpreting complex models***. Scalability and computational demands, especially for real-time data processing, also increase the obstacles. Other obstacles include regulatory and privacy concerns and adapting to the anomalies in the markets. Addressing these challenges is essential for improving the accuracy and reliability of stock market predictions using machine learning.

This objective is met by identifying the challenges that affect the usage of machine learning algorithms in stock market predictions. Issues such as ***data quality problems, overfitting, and the complexity of interpreting models*** like LSTM bring out the challenges in this analysis. Challenges also include the computationally high demand for processing real-time data and the scalings of such models. The use of machine learning algorithms, however, is complicated by regulatory and privacy issues alongside the challenge to adapt quickly to rapid shifts in market trends. Such challenges are an indication that even though the models show promise, significant improvements are needed for their practical application in stock market forecasting.

Covering the ***fourth objective***, it can be highlighted that on the basis of historical DJIA data, different models like ***LSTM, GRU, Prophet, and ANN*** are effective tools for the prediction of the stock market. It has been summarised that LSTM models handle fluctuations better while Prophet shows competitive accuracy with proper parameter tuning. Optimisation through techniques such as wavelet transformation and optimised deep frameworks improve the accuracy of the prediction process. SVM and random forest models also achieve high accuracy with large datasets.

In order to address this objective, developing and testing are conducted for various predictive models, including ***LSTM, GRU, Prophet, and ANN, with historical DJIA data***. The evaluation confirms that ***LSTM-based models*** are quite efficient in dealing with stock market volatility. On the other hand, ***Prophet*** provides ***very high accuracy*** in case the proper parameters are tuned for setting parameters. Optimisation techniques like wavelet transformation and deep learning frameworks enhance the prediction accuracy of the model as well. Additionally, models like ***SVM and Random Forest*** demonstrate ***high accuracy*** when applied to large datasets, further proving their value in forecasting stock market movements. These results confirm that time series machine learning models can be used to reinforce the capabilities of stock trend prediction.

5.3. Recommendations

Implement a continuous model monitoring system: Deploying machine learning models for the stock market requires a very frequent evaluation of their performance, especially because financial situations are characterised by constant change. It is recommended that monitoring systems be deployed automatically to track the performance of models in real time (Quiñones-Grueiro, et al., 2019). Parameters such as mean absolute error (MAE), root mean square error (RMSE), and prediction accuracy need to be continuously monitored. When performance drops due to market changes or data shifts, the system should flag the issue and trigger retraining or model adjustments. This will ensure that models stay valid and relevant across different conditions in the market.

Use ensemble learning for risk management: Instead of relying on one machine learning model, implement ensemble learning techniques, where predictions from multiple models, Random Forest, XGBoost, and more are combined. By aggregating the predictions, the ensemble approach reduces dependence on a single model's predictions, which might be over-sensitive to specific market conditions (Hamori, et al., 2018). In fact, this can enhance stability and reliability in stock market prediction and help manage risks associated with volatility in the market.

Integrating alternative data sources: Alternative data, such as social media sentiment feeds on Twitter, news headlines, and Google search trends add substantially to the predictiveness of a machine learning model. These alternative data sources may capture market sentiments and investor behavior at a close-to-real-time pace, as traditional financial indicators fail to do. Combine this alternative data with time series financial data for an overall better view of market movements and improve the predictiveness associated with them, especially during sudden market shifts or economic crises (Roh, et al., 2019).

Adaptive models for market anomalies: It's important to develop adaptive machine learning models that dynamically adapt to eventual or sudden market anomalies, like financial crises or geopolitical events. Some of these models could use meta-learning techniques where they learn how to adjust their structures or parameters based on large movements in market conditions (Meier, 2014). In that sense, a model that adjusts its learning rate or hyperparameters as it does when something as drastic as the 2008 financial crisis erupts can manage non-linearities and abrupt changes more effectively. In that regard, adaptive models do not only predict regular market behavior but perform much better when unexpected events occur.

Creating real-time feedback loops for model improvement: Develop a real-time feedback mechanism that makes it possible for the machine learning model to continually update itself with new data in the system and make predictions based on those updates. It can be achieved with algorithms of online learning, so it can adjust accordingly to new stock market data but is not fully trained again. The feedback loop could also involve incorporating expert feedback from financial analysts, enabling the model to refine its accuracy based on expert interpretations of real-world market changes. It would allow the model to stay relevant in fast-changing financial environments through an ongoing learning mechanism.

Utilising explainability as a trading strategy tool: Apart from the enhanced interpretability for decision-makers, insights provided by the XAI can be directly put to use as a tool for trading strategies. For instance, through SHAP values, identify what features of economic indicators or social sentiment spikes are actually contributing the most to stock price movement and further develop such informed trading strategies (Kalra and Mittal, 2024). The traders could rely upon the real-time explanation of the reasoning behind such a decision by the model and optimise their trading decisions by emphasising the most influencing factors involved in changes in the stock price.

5.4. Future recommendations

Expand data sources: Further research must include more data sources beyond financial numbers to expand the paper's applicability (Jiang, 2021). Other information, such as economic indicators, news from the rest of the world, and trends in specific markets. Using a wider range of data would make stock movement predictions even more accurate during uncertain times.

Investigate the impact of external shocks: Future studies should consider the effects that sudden changes within an economy can have on predicting stock markets. Most such shocks would arise from unforeseen natural disasters, downfalls of economies, or any other massive recurrences that affect individual expectations and investment strategies. Knowing how such shocks affect the behavior of the markets can help fine-tune the current models and produce other models, assuming the shocks.

Evaluate model robustness across markets: Future studies can determine whether machine learning models that are applicable in a specific stock market are also similarly useful in others. This can guide the potential application of models in various market environments (Khan, et al., 2018). Determining how well these models can be used across different markets will eventually result in providing more reliable tools for prediction globally for investors.

Develop hybrid modeling approaches: Future research in this direction can involve the incorporation of machine learning with traditional financial models. Researchers can utilise the best aspects of both domains by mixing up the techniques of machine learning with time series analysis or even econometric models. A hybrid model is most likely to be more accurate than others concerning its predictions for stocks by capturing complex patterns and historical data at the same time.

Investigate the role of behavioral economics: Future research might look at how behavioral economics affects stock market predictions, and how investors' emotions and behaviors change in response to a moving market. Adding this information to prediction models will help to understand how stock prices move during high periods of stress in the market in which emotions take a significant part of the trading decisions.

In conclusion, implementing these recommendations would enhance the accuracy and reliability of stock market predictions, which aids in better decision-making and an improved understanding of market dynamics in various conditions.

References

Abudy, M.M. (2020) Retail investors' trading and stock market liquidity. *The North American Journal of Economics and Finance*, 54, p.101281.

Achsan, B.M. (2020). *Support Vector Machine: Regression*. [Online] IT Paragon . Available at: https://medium.com/it-paragon/support-vector-machine-regression-cf65348b6345.

Adineh, A.H., Narimani, Z. and Satapathy, S.C. (2020) Importance of data preprocessing in time series prediction using SARIMA: A case study. *International Journal of Knowledge-based and Intelligent Engineering Systems*, *24*(4), pp.331-342.

Albahli, S., Irtaza, A., Nazir, T., Mehmood, A., Alkhalifah, A. and Albattah, W. (2022) A machine learning method for prediction of stock market using real-time twitter data. *Electronics*, *11*(20), p.3414.

Al-Gounmeein, R.S. and Ismail, M.T. (2021) Comparing the performances of artificial neural networks models based on autoregressive fractionally integrated moving average models. *IAENG International Journal of Computer Science*, *48*(2), pp.266-276.

Al-Thaqeb, S.A. and Algharabali, B.G. (2019) Economic policy uncertainty: A literature review. *The Journal of Economic Asymmetries*, 20, p.e00133.

Alzazah, F.S. and Cheng, X. (2020) Recent advances in stock market prediction using text mining: A survey. *E-Business-Higher Education and Intelligence Applications*, pp.1-34.

Baker, S.R., Bloom, N., Davis, S.J. and Kost, K.J. (2019) Policy news and stock market volatility (No. w25720). *National Bureau of Economic Research*.

Barrera-Animas, A.Y., Oyedele, L.O., Bilal, M., Akinosho, T.D., Delgado, J.M.D. and Akanbi, L.A. (2022) Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, *7*, p.100204.

Barrow, D., Kourentzes, N., Sandberg, R. and Niklewski, J. (2020) Automatic robust estimation for exponential smoothing: Perspectives from statistics and machine learning. *Expert Systems with Applications*, *160*, p.113637.

Belasri, Y. and Ellaia, R. (2017) Estimation of volatility and correlation with multivariate generalized autoregressive conditional heteroskedasticity models: an application to Moroccan stock markets. *International Journal of Economics and Financial Issues, 7*(2), pp.384-396.

Biktimirov, E.N. and Xu, Y. (2019) Market reactions to changes in the Dow Jones industrial average index. *International Journal of Managerial Finance*, *15*(5), pp.792-812.

Birant, D. and Işık, Z. (2019, October) Stock market forecasting using machine learning models. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)* (pp. 1-6). IEEE.

Botunac, I., Panjkota, A. and Matetic, M. (2019, January) The importance of time series data filtering for predicting the direction of stock market movement using neural networks. In *Proceedings of the 30th DAM International Symposium* (pp. 0886-0891).

Çelik, T.B., İcan, Ö. and Bulut, E. (2023) Extending machine learning prediction capabilities by explainable AI in financial time series prediction. *Applied Soft Computing*, *132*, p.109876.

Chen, W., Milosevic, Z., Rabhi, F.A. and Berry, A. (2023) Real-time analytics: Concepts, architectures and ML/AI considerations. *IEEE Access*.

Chhajer, P., Shah, M. and Kshirsagar, A. (2022) The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. *Decision Analytics Journal, 2,* p.100015.

Chung, H. and Shin, K.S. (2018) Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, *10*(10), p.3765.

Dadhich, M., Pahwa, M.S., Jain, V. and Doshi, R. (2021) Predictive models for stock market index using stochastic time series ARIMA modelling in emerging economy. In *Advances in Mechanical Engineering: Select Proceedings of CAMSE 2020* (pp. 281-290). Springer Singapore.

Damrongsakmethee, T. and Neagoe, V.E. (2020, June) Stock market prediction using a deep learning approach. In *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* (pp. 1-6). IEEE.

Dannels, S.A. (2018) Research design. In *The reviewer's guide to quantitative methods in the social sciences* (pp. 402-416). Routledge.

Dhingra, B., Batra, S., Aggarwal, V., Yadav, M. and Kumar, P. (2024) Stock market volatility: a systematic review. *Journal of Modelling in Management*, *19*(3), pp.925-952.

Dong, J., Chen, Y., Yao, B., Zhang, X. and Zeng, N. (2022) A neural network boosting regression model based on XGBoost. *Applied Soft Computing*, *125*, p.109067.

Ghania, M.U., Awaisa, M. and Muzammula, M. (2019) Stock market prediction using machine learning (ML) algorithms. *ADCAIJ: Adv Distrib Comput Artif Intell*, *8*(4), pp.97-116.

Graw, J.H., Wood, W.T. and Phrampus, B.J. (2021) Predicting global marine sediment density using the random forest regressor machine learning algorithm. *Journal of Geophysical Research: Solid Earth*, *126*(1), p.e2020JB020135.

Gudivada, V., Apon, A. and Ding, J. (2017) Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, *10*(1), pp.1-20.

Hafeez, A. (2023) *Global Stock Markets: Insights and Future Trends*. [Online] Kuvera. Available at: https://kuvera.in/blog/global-stock-markets-insights-and-future-trends/ [Accessed 21 Sep. 2024].

Hamori, S., Kawai, M., Kume, T., Murakami, Y. and Watanabe, C. (2018) Ensemble learning or deep learning? Application to default risk analysis. *Journal of Risk and Financial Management*, *11*(1), p.12.

Hansun, S. (2016) A new approach of brown's double exponential smoothing method in time series analysis. *Balkan Journal of Electrical and Computer Engineering*, *4*(2), pp.75-78.

Hao, J. and Ho, T.K. (2019) Machine learning made easy: a review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics*, *44*(3), pp.348-361.

He, Q., Liu, J., Wang, S. and Yu, J. (2020) The impact of COVID-19 on stock markets. *Economic and Political Studies*, 8(3), pp.275-288.

Hodeghatta, U.R. and Nayak, U. (2023) Time Series: Forecasting. In *Practical Business Analytics Using R and Python: Solve Business Problems Using a Data-driven Approach* (pp. 443-484). Berkeley, CA: Apress.

Hodson, T.O. (2022) Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, *2022*, pp.1-10.

Hodson, T.O., Over, T.M. and Foks, S.S. (2021) Mean squared error, deconstructed. *Journal of Advances in Modeling Earth Systems*, *13*(12), p.e2021MS002681.

Inaba, K.I. (2020) A global look into stock market comovements. *Review of World Economics*, *156*(3), pp.517-555.

Javatpoint (2024). *Linear Regression in Machine learning - Javatpoint*. [Online] www.javatpoint.com. Available at: https://www.javatpoint.com/linear-regression-in-machine-learning.

Jena, M., Behera, R.K. and Rath, S.K. (2020) Machine learning models for stock prediction using real-time streaming data. In *Biologically Inspired Techniques in Many-Criteria Decision Making: International Conference on Biologically Inspired Techniques in Many-Criteria Decision Making (BITMDM-2019)* (pp. 101-108). Springer International Publishing.

Jia, S., Hou, C. and Wang, J. (2017) Software aging analysis and prediction in a web server based on multiple linear regression algorithm. In *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)* (pp. 1452-1456). IEEE.

Jiang, W. (2021) Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, *184*, p.115537.

Jouilil, Y. (2023) Comparing the accuracy of classical and machine learning methods in time series forecasting: A case study of US inflation. *Statistics, Optimization & Information Computing*, *11*(4), pp.1041-1050.

Jyothirmayee, S., Kumar, V.D., Rao, C.S. and Shankar, R.S. (2019) Predicting stock exchange using supervised learning algorithms. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), pp.4081-4090.

Kaab, A., Bashah, A.F.A. and Wahhab, A.M.A. (2023) Evidence from Iraqi Banks on the Reporting of Sustainability Determinants and Their Impact on Market Returns. *Technium Soc. Sci. J.*, *44*, p.79.

Kalra, A. and Mittal, R. (2024) Explainable AI for Improved Financial Decision Support in Trading. In *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)* (pp. 1-6). IEEE.

Khan, U., Aadil, F., Ghazanfar, M.A., Khan, S., Metawa, N., Muhammad, K., Mehmood, I. and Nam, Y. (2018) A robust regression-based stock exchange forecasting and determination of correlation between stock markets. *Sustainability*, *10*(10), p.3702.

Khan, W., Malik, U., Ghazanfar, M.A., Azam, M.A., Alyoubi, K.H. and Alfakeeh, A.S. (2020) Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis. *Soft Computing*, *24*(15), pp.11019-11043.

Krishnan, M. (2020) Against interpretability: a critical examination of the interpretability problem in machine learning. *Philosophy & Technology*, *33*(3), pp.487-502.

Lee, N.U., Shim, J.S., Ju, Y.W. and Park, S.C. (2018) Design and implementation of the SARIMA–SVM time series analysis algorithm for the improvement of atmospheric environment forecast accuracy. *Soft Computing*, *22*, pp.4275-4281.

Lemenkova, P. (2019) Processing oceanographic data by Python libraries NumPy, SciPy and Pandas. *Aquatic Research*, *2*(2), pp.73-91.

Li, K. and Zhou, Y. (2024) Improved Financial Predicting Method Based on Time Series Long Short-Term Memory Algorithm. *Mathematics*, *12*(7), p.1074.

Li, Q., Chen, Y., Wang, J., Chen, Y. and Chen, H. (2017) Web media and stock markets: A survey and future directions from a big data perspective. *IEEE Transactions on Knowledge and Data Engineering*, *30*(2), pp.381-399.

Lin, J., Selden, G.C., Shoven, J.B. and Sialm, C. (2021) *Replicating the Dow Jones Industrial Average* (No. w28528). National Bureau of Economic Research.

Lu, R. and Lu, M. (2021) Stock trend prediction algorithm based on deep recurrent neural network. *Wireless Communications and Mobile Computing*, 2021(1), p.5694975.

Mahakud, J. and Dash, S.R. (2016) Asset pricing models, cross section of expected stock returns and financial market anomalies: A review of theories and evidences. *Journal of Management Research*, *16*(4), pp.230-249.

Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M.M. and Muhammad, K. (2020) A local and global event sentiment based efficient stock exchange forecasting using deep learning. *International Journal of Information Management*, 50, pp.432-451.

Mehtab, S. and Sen, J. (2020) A time series analysis-based stock price prediction using machine learning and deep learning models. *International Journal of Business Forecasting and Marketing Intelligence*, *6*(4), pp.272-335.

Meier, C. (2014) Adaptive market efficiency: review of recent empirical evidence on the persistence of stock market anomalies. *Review of Integrative Business and Economics Research*, *3*(2), p.268.

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A. and Salwana, E. (2020) Deep learning for stock market prediction. *Entropy*, *22*(8), p.840.

Naik, P. and Reddy, Y.V. (2021) Stock market liquidity: A literature review. *Sage Open*, 11(1), p.2158244020985529.

Nassif, A.B., Talib, M.A., Nasir, Q. and Dakalbab, F.M. (2021) Machine learning for anomaly detection: A systematic review. *Ieee Access*, *9*, pp.78658-78700.

Nenu, E.A., Vintilă, G. and Gherghina, Ş.C. (2018) The impact of capital structure on risk and firm performance: Empirical evidence for the Bucharest Stock Exchange listed companies. *International Journal of Financial Studies*, 6(2), p.41.

Nti, I.K., Adekoya, A.F. and Weyori, B.A. (2020) A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, 53(4), pp.3007-3057.

Nyoni, T. (2018) Modeling and forecasting inflation in zimbabwe: a generalized autoregressive conditionally heteroskedastic (GARCH) approach.

Özen, E. and Tetik, M. (2019) Did developed and developing stock markets react similarly to Dow Jones during 2008 crisis? *Frontiers in Applied Mathematics and Statistics*, *5*, p.49.

Pardo, F.D.M. and López, R.C. (2020) Mitigating overfitting on financial datasets with generative adversarial networks. *The Journal of Financial Data Science*, *2*(1), pp.76-85.'

Phan, D.H.B. and Narayan, P.K. (2021) Country responses and the reaction of the stock market to COVID-19—A preliminary exposition. In *Research on Pandemics* (pp. 6-18). Routledge.

Potla, R.T. (2022) Scalable Machine Learning Algorithms for Big Data Analytics: Challenges and Opportunities. *Journal of Artificial Intelligence Research*, *2*(2), pp.124-141.

Quantinsti (2017). *Using ARIMA Model for Forecasting Stock Returns*. [online] QuantInsti. Available at: https://blog.quantinsti.com/forecasting-stock-returns-using-arima-model/.

Quiñones-Grueiro, M., Prieto-Moreno, A., Verde, C. and Llanes-Santiago, O. (2019) Data-driven monitoring of multimode continuous processes: A review. *Chemometrics and Intelligent Laboratory Systems*, *189*, pp.56-71.

Rahul, Sarangi, S., Kedia, P. and Monika (2020) Analysis of various approaches for stock market prediction. *Journal of Statistics and Management Systems*, *23*(2), pp.285-293.

Rask, J.K., Madsen, F.P., Battle, N., Macedo, H.D. and Larsen, P.G. (2021) Visual studio code vdm support. In *Proceedings of the 18th International Overture Workshop* (pp. 35-49).

Roelofs, R., Shankar, V., Recht, B., Fridovich-Keil, S., Hardt, M., Miller, J. and Schmidt, L. (2019) A meta-analysis of overfitting in machine learning. *Advances in Neural Information Processing Systems*, *32*.

Roh, Y., Heo, G. and Whang, S.E. (2019) A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, *33*(4), pp.1328-1347.

Rubio, L., Palacio Pinedo, A., Mejía Castaño, A. and Ramos, F. (2023) Forecasting volatility by using wavelet transform, ARIMA and GARCH models. *Eurasian Economic Review*, *13*(3), pp.803-830.

Saetia, K. and Yokrattanasak, J. (2022) Stock movement prediction using machine learning based on technical indicators and Google trend searches in Thailand. *International Journal of Financial Studies*, 11(1), p.5.

Sanz, J.L. and Zhu, Y. (2021) Toward scalable artificial intelligence in finance. In *2021 IEEE International Conference on Services Computing (SCC)* (pp. 460-469). IEEE.

Schaffer, A.L., Dobbins, T.A. and Pearson, S.A. (2021) Interrupted time series analysis using autoregressive integrated moving average (ARIMA) models: a guide for evaluating large-scale health interventions. *BMC medical research methodology*, *21,* pp.1-12.

Shah, D., Isah, H. and Zulkernine, F. (2019) Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, 7(2), p.26.

Sheth, D. and Shah, M. (2023) Predicting stock market using machine learning: best and accurate way to know future stock prices. *International Journal of System Assurance Engineering and Management*, 14(1), pp.1-18.

Srinath, K.R. (2017) Python–the fastest growing programming language. *International Research Journal of Engineering and Technology*, *4*(12), pp.354-357.

Sunki, A., SatyaKumar, C., Narayana, G.S., Koppera, V. and Hakeem, M. (2024) Time series forecasting of stock market using ARIMA, LSTM and FB prophet. In *MATEC Web of Conferences* (Vol. 392, p. 01163). EDP Sciences.

Sutrick, K. (2017) Teaching R² in Regression. *Business Education Innovation Journal*, *9*(1).

Syed, A.M. and Bajwa, I.A. (2018) Earnings announcements, stock price reaction and market efficiency–the case of Saudi Arabia. *International Journal of Islamic and Middle Eastern Finance and Management*, 11(3), pp.416-431.

Tan, J., Chen, Y. and Jiao, S. (2023) Visual Studio Code in Introductory Computer Science Course: An Experience Report. *arXiv preprint arXiv:2303.10174*.

Verma, S., Sahu, S.P. and Sahu, T.P. (2023) Stock market forecasting with different input indicators using machine learning and deep learning techniques: A review. *Engineering Letters*, 31(1).

Voigt, P. and Von dem Bussche, A. (2017) The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, *10*(3152676), pp.10-5555.

Wall Street (2024) *Generalized Autoregressive Conditional Heteroskedasticity*. [online] Wallstreetmojo.com. Available at: https://www.wallstreetmojo.com/generalized-autoregressive-conditional-heteroskedasticity/ [Accessed 23 Sep. 2024].

Wang, J., He, J., Feng, C., Feng, L. and Li, Y. (2021) Stock index prediction and uncertainty analysis using multi-scale nonlinear ensemble paradigm of optimal feature extraction, two-stage deep learning and Gaussian process regression. *Applied Soft Computing*, 113, p.107898.

Wen, M., Li, P., Zhang, L. and Chen, Y. (2019) Stock market trend prediction using high-order information of time series. *Ieee Access*, *7*, pp.28299-28308.

Weng, B., Ahmed, M.A. and Megahed, F.M. (2017) Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications*, *79*, pp.153-163.

Xiao, D. and Su, J. (2022) Research on stock price time series prediction based on deep learning and autoregressive integrated moving average. *Scientific Programming*, *2022*(1), p.4758698.

Xu, G., Xu, C.Z. and Jiang, S. (2016) Prophet: Scheduling executors with time-varying resource demands on data-parallel computation frameworks. In *2016 IEEE International Conference on Autonomic Computing (ICAC)* (pp. 45-54). IEEE.

Xue, D.M. and Hua, Z.Q. (2016) ARIMA based time series forecasting model. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, *9*(2), pp.93-98.

Yadav, A., Jha, C.K. and Sharan, A. (2020) Optimizing LSTM for time series prediction in the Indian stock market. *Procedia Computer Science*, *167*, pp.2091-2100.

Ying, X. (2019) An overview of overfitting and its solutions. In *Journal of physics: Conference series* (Vol. 1168, p. 022022). IOP Publishing.

Yusof, U.K., Khalid, M.N.A., Hussain, A. and Shamsudin, H. (2020) Financial time series forecasting using prophet. In *International Conference of Reliable Information and Communication Technology* (pp. 485-495). Cham: Springer International Publishing.

Yusof, U.K., Khalid, M.N.A., Hussain, A. and Shamsudin, H. (2020, December) Financial time series forecasting using prophet. In *International Conference of Reliable Information and Communication Technology* (pp. 485-495). Cham: Springer International Publishing.

Zakhidov, G. (2024) Economic indicators: tools for analyzing market trends and predicting future performance. *International Multidisciplinary Journal of Universal Scientific Prospectives*, 2(3), pp.23-29.

Zemkoho, A. (2022) A basic time series forecasting course with python. In *Operations Research Forum* (Vol. 4, No. 1, p. 2). Cham: Springer International Publishing.

Zhang, F. and O'Donnell, L.J. (2020) Support vector regression. In *Machine learning* (pp. 123-140). Academic Press.

Zolfaghari, M. and Gholami, S. (2021) A hybrid approach of adaptive wavelet transform, long short-term memory and ARIMA-GARCH family models for the stock index prediction. *Expert Systems with Applications*, *182*, p.115149.