6005EFA

The Predictive Model of Stock Prices using Machine Learning in the UK

Coventry University

Finance and Investment

Calvin Sowemimo

Table of Contents

Chapter 1 – Introduction 5

1.1 Introduction to Stock Price Prediction Using Machine Learning in the UK 5

1.2 Research Problem and Significance 5

1.3 Objectives and Research Questions 5

1.4 Dissertation Overview 5

Chapter 2 - Literature Review 6

2.1 Evolution of Prediction Models 6

2.1.1 Transition to Machine Learning 6

2.1.2 Significance of ML in Financial Prediction 6

2.2 Theoretical Framework 7

2.2.1 Efficient Market Hypothesis (EMH) & Behavioural Finance 7

2.2.2 Support Vector Machines (SVM) 7

2.3 Machine Learning (ML) Techniques in Stock Price Forecasting 8

2.3.1 Overview of ML Techniques 8

2.3.2 Effectiveness of Specific ML Techniques 8

2.3.3 Challenges and Considerations 9

2.4 Identification of Research Gap 9

2.4.1 Synthesis of Reviewed Literature 9

2.4.2 Gaps in Current Research 9

2.4.3 Justification for Current Study 9

Chapter 3 – Hypotheses Development 10

3.1 Hypotheses 10

3.1.1 ML Algorithms vs. Traditional Models (H_1) 10

3.1.2 Market Sentiment Hypothesis (H_2) 10

3.1.3 Deep Learning Algorithms Hypothesis (H_3) 11

3.1.4 Sector-Specific Predictive Accuracy Hypothesis (H_4) 11

Chapter 4 – Methodology 11

4.1 Research Design 11

4.1.1 Introduction to Research Design 11

4.1.2 Justification for Quantitative Design 11

4.2 Data Collection and Preparation 12

4.2.1 Data Collection 12

4.2.2 Data Preparation 13

4.3 Machine Learning Algorithms and Implementation 13

4.3.1 Selection of ML Algorithms 13

4.3.2 Implementation of ML Algorithms........14
4.4 Model Evaluation and Validation........14
4.4.1 Introduction to Model Evaluation........14
4.4.2 Cross-Validation Techniques........14
4.4.3 Performance Metrics........14
4.4.4 Time-Series Cross-Validation........14
4.5 Ethics and Limitations........15
4.5.1 Ethical Considerations........15
4.5.2 Limitations and Challenges........15
Chapter 5 – Data Analysis and Findings........16
5.1 Introduction to Analysis........16
5.2 Data Collection and Preparation........16
5.2.1 Quantitative Data Collection........16
5.2.2 Qualitative Data Collection........16
5.2.3 Data Cleaning and Integration........16
5.3 Implementation of ML Algorithms........17
5.3.1 Model Selection and Justification........17
5.3.2 Model Training and Setup........18
5.5 Comparative Analysis of Model Performance........21
5.5.1 Performance Metrics Comparison........21
5.5.2 Economic Indicators Analysis........23
5.6 Impact of Sentiment Analysis........24
5.6.1 Sentiment Data Processing........24
5.6.2 Influence on Predictive Accuracy........25
5.7 Concluding Remarks........26
Chapter 6 – Conclusion........26
6.1 Research Problem and Significance........26
6.2 Objectives and Research Questions........27
6.3 Dissertation Overview........27
Appendices........28
Appendix A – Stock Price Script........28
Appendix B – Headlines Script........29
Appendix C – Data Cleansing Script........30
Appendix D – Integration Script........32
Appendix E – Integration Script........33
Appendix F – Model Implementation Script........34

References 38

Chapter 1 – Introduction

1.1 Introduction to Stock Price Prediction Using Machine Learning in the UK

The ability to accurately forecast stock prices remains a central objective within the financial sector, crucial for investors, analysts, and policymakers alike. Traditional models, grounded in the Efficient Market Hypothesis (EMH) and Fundamental Analysis, have historically provided the backbone for such predictions. However, the rapidly evolving landscape of financial markets, characterised by complex, data-rich environments, has exposed the limitations of these traditional approaches. The emergence of Machine Learning (ML) techniques represents a pivotal shift in this domain, offering a new paradigm that promises to enhance the precision of financial forecasts by harnessing vast, diverse datasets and uncovering intricate market patterns (Fama, 1970; Graham & Dodd, 1934; Sharpe, 1964).

1.2 Research Problem and Significance

This dissertation investigates the effectiveness of ML algorithms in predicting stock prices, comparing their performance to that of traditional financial models. Given the burgeoning volume of digital financial data and the global interconnectivity of financial markets, the UK stock market serves as an exemplary focus for this research. The study hypothesises that ML, with its advanced computational capabilities and adaptability, could significantly outperform traditional models in forecasting accuracy, thereby providing investors and market analysts with a more reliable foundation for decision-making.

1.3 Objectives and Research Questions

The primary aim is to evaluate various ML techniques' predictive accuracy within the UK market, specifically assessing their superiority over traditional models. This objective raises several key research questions:

1. How do ML techniques compare to traditional financial models in predicting stock prices within the UK market?
2. Which ML algorithms demonstrate the highest predictive accuracy, and under what conditions do they excel?
3. What role does market sentiment, captured through sentiment analysis, play in enhancing the predictive accuracy of ML models?

1.4 Dissertation Overview

Structured across several key chapters, this dissertation commences with a literature review that traces the evolution of stock price prediction models, from traditional theories to the incorporation of ML techniques. It establishes a theoretical framework that intertwines the Efficient Market Hypothesis with Behavioural Finance, setting the stage for exploring ML's potential in financial forecasting.

Following the literature review, hypotheses are developed to guide the empirical investigation, focusing on comparing ML algorithms against traditional models and examining the impact of market sentiment on predictive accuracy. The methodology chapter details the quantitative research design, data collection, and preparation processes, highlighting the selection and implementation of specific ML algorithms for analysis.

The data analysis and findings sections present a comprehensive examination of the ML algorithms' performance, offering insights into their effectiveness in predicting stock prices and the influence of market sentiment. Concluding remarks synthesise the study's key findings, discuss its implications for financial forecasting and machine learning application, and suggest avenues for future research.

By delving into the predictive capabilities of machine learning in the context of the UK stock market, this dissertation aims to contribute significantly to the fields of financial analysis and computational finance, enhancing our understanding of how advanced analytical techniques can be leveraged to forecast market movements more accurately.

Chapter 2 - Literature Review

2.1 Evolution of Prediction Models

2.1.1 Transition to Machine Learning

The evolution of stock price prediction models has been marked by a significant shift from reliance on traditional financial theories to the sophisticated application of machine learning (ML) techniques. Initially, the Capital Asset Pricing Model (CAPM) and Fundamental Analysis were foundational to predicting stock prices. Developed in the mid-20th century, CAPM theorises that the return on a stock is directly proportional to its market risk, suggesting a linear relationship between expected returns and the systematic risk (Sharpe, 1964). Meanwhile, Fundamental Analysis, advocated by Graham and Dodd (1934), involves examination of company finances, management efficacy, market conditions, and macroeconomic factors to determine a stock's intrinsic value. Despite historical significance, these models often failed to capture the market's complex, dynamic nature, leading to predictions that sometimes diverged significantly from actual market movements.

The advent of machine learning (ML) techniques signals a paradigm shift in stock market prediction strategies. Unlike their predecessors, ML models handle vast datasets, uncover hidden patterns, and adapt without explicit reprogramming. This transition was facilitated by the realisation that traditional models were inadequate for processing the nonlinear relationships exhibited by financial markets (Bishop, 2006). Neural networks, a subset of ML techniques, exemplify this evolution by utilising layers of interconnected nodes to analyse past price data and other relevant financial metrics, enhancing the precision of future stock price predictions (Zhang, Patuwo, & Hu, 1998).

2.1.2 Significance of ML in Financial Prediction

Support Vector Machines (SVMs) emerged as another influential ML technique for stock prediction, excelling in both classification and regression tasks by identifying the optimal hyperplane to segregate data points of different classes in the feature space (Cortes & Vapnik, 1995). The introduction of ensemble methods and deep learning further advanced the predictive capabilities of stock market models. These methods leverage the collective power of multiple prediction models and the ability to process unstructured data, such as news and social media content, offering a more nuanced understanding of factors influencing stock prices (Freund & Schapire, 1997; LeCun, Bengio, & Hinton, 2015).

Despite the progress, transitioning to ML in finance presented challenges such as overfitting, where models excel on training data but underperform on new data, and the opaque nature of some ML models, complicating the interpretation of their decision-making processes (Hawkins, 2004). Additionally, the effectiveness of ML models is contingent upon the quality and comprehensiveness of the training data, requiring data preparation and feature selection (Bishop, 2006). Moreover, the dynamic and interconnected nature of global financial markets demands models that can adapt to rapid changes and incorporate a wide array of economic indicators and news data.

The shift from traditional to ML models in stock prediction marks significant strides in financial analysis, enabling the utilisation of large datasets for better investment decisions. Although there is ongoing work to perfect these models, their use is progressively equalising access to sophisticated market analysis.

2.2 Theoretical Framework

2.2.1 Efficient Market Hypothesis (EMH) & Behavioural Finance

The theoretical underpinnings of stock price prediction models have traditionally been rooted in the Efficient Market Hypothesis (EMH) and Behavioural Finance, each offering distinct perspectives on market behaviour and stock price predictability. The EMH, introduced by Fama (1970), theorises stock prices fully reflect all available information, rendering it impossible to consistently achieve higher returns than the market average through any analysis of publicly available information. According to EMH, the market operates with such efficiency that as soon as new information becomes available, it is instantly and accurately reflected in stock prices, making gains on market predictions void (Fama, 1970). This hypothesis outlines three forms of market efficiency: weak, semi-strong, and strong, each defining the extent to which different types of information are reflected in stock prices.

Contrasting the premises of EMH, Behavioural Finance introduces psychological insights into how investors make financial decisions, challenging the notion of market participants as rational actors. Pioneered by scholars like Kahneman and Tversky (1979), Behavioural Finance suggests that cognitive biases significantly influence investor behaviour, leading to predictable patterns in stock prices, which can be exploited for profit. This field of study highlights anomalies such as overreaction, underreaction, and other inefficiencies that cannot be explained by traditional financial theories, suggesting psychological factors play a critical role in financial markets.

The integration of ML into stock price prediction is underpinned by both the Efficient Market Hypothesis (EMH) and Behavioural Finance. EMH limits predictability from public data, while Behavioural Finance suggests ML's potential by uncovering patterns in extensive datasets, including news and social media, that traditional methods may miss.

2.2.2 Support Vector Machines (SVM)

Support Vector Machines (SVMs), leverage the concept of maximising the margin between data points of different classes, making them particularly suitable for categorising stock price movements as bullish or bearish based on historical data (Cortes & Vapnik, 1995). Similarly, neural networks, with their ability to learn complex nonlinear relationships, can model the intricate interactions between various economic indicators and investor sentiment, offering predictions on stock price movements that consider both rational financial indicators and irrational investor behaviours (Zhang, Patuwo, & Hu, 1998).

Moreover, the emergence of ensemble methods and deep learning reflects an advanced application of Behavioural Finance principles, combining predictions from multiple models and analysing unstructured data to gauge market sentiment more accurately. These techniques embody the theoretical progression from viewing markets as purely rational systems to recognising the significant impact of human psychology and collective behaviour on financial markets (Freund & Schapire, 1997; LeCun, Bengio, & Hinton, 2015).

The theoretical foundation of stock price prediction intertwines the Efficient Market Hypothesis with Behavioural Finance, enhancing model development with ML advancements. These theories, acknowledging human rationality's limits and leveraging patterns in investor behaviour, position ML at the forefront of financial analysis to decode complex market dynamics.

2.3 Machine Learning (ML) Techniques in Stock Price Forecasting

2.3.1 Overview of ML Techniques

The integration of ML techniques into stock price forecasting represents a significant advancement in financial analysis, leveraging the ability to process large volumes of data to uncover complex patterns and predict future market movements. Among the plethora of ML techniques, Support Vector Machines (SVMs) and neural networks have emerged as particularly influential, each with its unique strengths in handling the multifaceted nature of financial markets.

2.3.2 Effectiveness of Specific ML Techniques

In the realm of stock price forecasting, neural networks and deep learning have significantly advanced financial analysis capabilities. Neural networks, with their deep architectures mimicking the human brain, excel in identifying complex patterns within financial markets. These models effectively analyse trends and trading volumes, thereby enhancing the accuracy of predictions by incorporating a wide array of financial indicators (Zhang, Patuwo, & Hu, 1998).

Deep learning further extends this analytical depth, leveraging vast datasets to reveal subtle market dynamics traditional models may overlook. Its ability to process both structured and unstructured data enables deep learning models to offer comprehensive insights into future market movements (LeCun, Bengio, & Hinton, 2015). This approach is instrumental in dissecting the multifaceted nature of financial markets, providing a granular understanding that facilitates more accurate forecasting.

While SVMs, introduced earlier, excel in classification and regression tasks, their integration with neural networks and deep learning techniques epitomises the evolution of ML applications in finance. Ensemble methods, amalgamating various models' forecasts, epitomise the field's innovation, enhancing prediction reliability and mitigating the risks associated with reliance on a single analytical method (Freund & Schapire, 1997).

2.3.3 Challenges and Considerations

Despite their promise, these advanced ML techniques face challenges such as overfitting and the complexity of ensuring model interpretability. The effectiveness of these models is inherently tied to the quality and scope of the training data, underscoring the importance of comprehensive data preparation and meticulous feature selection (Hawkins, 2004).

This exploration into neural networks, deep learning, and the comparative benefits of SVMs within stock price forecasting underscores the transformative impact of ML on financial analysis. These methodologies not only push the boundaries of traditional financial prediction models but also highlight the continuing evolution towards more nuanced, data-driven market analyses.

2.4 Identification of Research Gap

2.4.1 Synthesis of Reviewed Literature

While machine learning (ML) techniques in stock price forecasting offer promising advances, research especially within the UK market context remains underexplored (Cortes & Vapnik, 1995; LeCun, Bengio, & Hinton, 2015). A critical examination of SVMs, neural networks, and deep learning across markets uncovers a need for further, targeted investigations.

2.4.2 Gaps in Current Research

One notable research gap emerges from the need for comprehensive comparative studies that elucidate the relative performance of different ML techniques in stock price prediction. While individual studies have affirmed the effectiveness of specific models like SVMs and neural networks in forecasting (Zhang, Patuwo, & Hu, 1998), there is a scarcity of research directly comparing these models under a uniform dataset and identical market conditions. Such comparative analyses are crucial for understanding the strengths and limitations of each model, offering insights that could guide the selection of the most appropriate ML technique for specific market scenarios.

Additionally, the integration of global economic indicators and news sentiment in ML models for stock prediction presents an underexplored area of research. The dynamic and interconnected nature of global financial markets suggests that external economic events and news sentiment play a significant role in influencing stock prices. While deep learning models have shown promise in analysing unstructured data like news articles (LeCun, Bengio, & Hinton, 2015), there is a lack of focused research on how these models can be optimised to incorporate global economic indicators and sentiment analysis effectively for the UK stock market. This gap points towards the potential for developing more sophisticated ML models that can capture the multifaceted impacts of global events on stock prices.

2.4.3 Justification for Current Study

Furthermore, the challenge of overfitting remains a significant concern in the development of ML models for stock price prediction (Hawkins, 2004). Although various strategies have been proposed to mitigate overfitting, the literature lacks a systematic evaluation of these approaches in the specific context of stock market forecasting. Research into novel methodologies or the refinement of existing techniques to prevent overfitting, especially in models trained on highly volatile stock market data, is imperative for enhancing the robustness and reliability of predictions.

The exploration of ML in stock forecasting underscores identified gaps, such as the need for comparative ML technique studies, better integration of global indicators and sentiment analysis, and strategies against overfitting, particularly within the UK market context. This research aims to fill these gaps, contributing to the evolution of stock price prediction methodologies and aiding in more informed financial market analysis.

Chapter 3 – Hypotheses Development

The development of hypotheses bridges the conceptual framework outlined in the literature review with the empirical exploration that follows. This study leverages the advanced predictive capabilities of machine learning (ML) algorithms to forecast stock prices within the UK market. It proposes a series of hypotheses rooted in financial theory and computational advances, aiming to address existing gaps and contribute novel insights to the field. The complexity of the UK's financial market, characterised by its nuanced responses to domestic and global economic shifts, provides an opportune backdrop for examining the synergy between ML technology and financial forecasting.

3.1 Hypotheses

3.1.1 ML Algorithms vs. Traditional Models (H_1)

Grounded in the Efficient Market Hypothesis (EMH) and the Adaptive Markets Hypothesis (Lo, 2004), H1 hypothesises ML algorithms will demonstrate superior predictive accuracy over traditional financial models. This hypothesis addresses the gap identified in the literature, where traditional models often fall short in capturing the market's complex, non-linear dynamics (Fama, 1970).

Empirical Testing Strategy: Utilising historical stock price data, the performance of ML algorithms will be compared against traditional models, employing metrics such as accuracy, precision, recall, and F1-score. This strategy not only allows for a quantitative assessment of each model's effectiveness but also contributes to a deeper understanding of computational finance's evolving role in market analysis.

3.1.2 Market Sentiment Hypothesis (H_2)

H2 examines the potential of market sentiment, derived from news articles and social media, to enhance ML models' predictive accuracy. This hypothesis responds to the call for integrating behavioural finance insights into quantitative models, recognising that investor sentiment significantly influences market dynamics (Kahneman & Tversky, 1979).

Empirical Testing Strategy: By incorporating sentiment scores into ML models, this study will quantitatively assess the impact of sentiment analysis on predictive accuracy. This exploration not only tests the hypothesis but also showcases the interdisciplinary approach of blending computational methods with behavioural finance theories.

3.1.3 Deep Learning Algorithms Hypothesis (H_3)

Asserting that deep learning algorithms will outperform traditional and standard ML techniques, H3 explores these algorithms' capacity to navigate the UK stock market's complexities. This hypothesis highlights the advanced analytical potential of deep learning in financial forecasting (LeCun, Bengio, & Hinton, 2015), contributing to the burgeoning discourse on AI's application in finance.

Empirical Testing Strategy: Deep learning models will be evaluated against traditional and ML models to determine their relative predictive power. This comparative analysis will not only validate the hypothesis but also advance the conversation on optimising AI for financial analysis.

3.1.4 Sector-Specific Predictive Accuracy Hypothesis (H_4)

Investigating the variability in predictive accuracy across different market sectors, H4 acknowledges the unique economic drivers and market sensitivities that distinguish each sector. This hypothesis is designed to uncover sector-specific insights that can refine and enhance the applicability of ML in stock forecasting.

Empirical Testing Strategy: A sectorial analysis will be conducted, applying ML models to stock data from various sectors and evaluating their predictive accuracies. This comprehensive approach will illuminate the nuanced performance of ML models, underscoring the importance of sector-specific strategies in financial forecasting.

Chapter 4 – Methodology

4.1 Research Design

4.1.1 Introduction to Research Design

In this study, we explore the use of machine learning (ML) algorithms for predicting stock prices in the UK market, employing a quantitative research design known for its systematic approach to data analysis. This methodology enables a direct comparison of ML techniques with traditional financial models, focusing on precise measurement and statistical analysis to validate research hypotheses. It allows for a detailed examination of the algorithms' predictive accuracy and the factors influencing stock price movements. Through this quantitative approach, we aim to enrich the financial analysis field by offering new insights into the effectiveness of ML methods, thereby filling a notable research gap (Creswell & Creswell, 2017).

4.1.2 Justification for Quantitative Design

This study is grounded in a quantitative research paradigm, meticulously chosen to address its primary aim: to dissect and compare the effectiveness of ML algorithms in stock price prediction through the lens of accuracy, precision, and recall metrics. These metrics, extracted from an analysis of historical stock price movements, economic indicators, and sentiment data, serve as the foundation for evaluating the performance of each algorithm. The quantitative framework empowers this research to apply sophisticated statistical methods and ML algorithms to extensive datasets, facilitating an in-depth investigation into complex patterns and relationships that might elude qualitative analysis (Hair Jr, Black, Babin, & Anderson, 2010).

A significant aspect of our methodological design involves the empirical validation of our hypotheses through rigorous regression analysis and comparative performance metrics (James, Witten, Hastie, & Tibshirani, 2013). This includes a detailed examination of how the inclusion of market sentiment indicators within ML models affects their predictive accuracy, showcasing the quantitative approach's capacity to handle nuanced data analysis.

Data for this analysis are meticulously sourced from authoritative platforms such as the London Stock Exchange and esteemed financial news outlets, ensuring a robust and reliable empirical foundation. Utilising Python's powerful libraries, such as pandas for data manipulation and scikit-learn for machine learning development, this research leverages state-of-the-art computational tools to unravel the intricacies of financial forecasting (McKinney, 2012; Pedregosa et al., 2011).

The choice of a quantitative design is pivotal, mirroring the study's aim to methodically investigate the hypotheses. This structured approach is vital for producing statistically significant results that enhance understanding of ML's role in financial forecasting. It seeks to reveal the predictive strength of ML algorithms, shedding light on their effectiveness in stock market analysis.

4.2 Data Collection and Preparation

4.2.1 Data Collection

The data collection strategy for this research is meticulously designed to harness a comprehensive understanding of the UK stock market dynamics, utilising a dual-source approach that blends quantitative stock data with qualitative sentiment indicators. This multifaceted methodology is pivotal in constructing a dataset that captures the full spectrum of factors influencing stock prices, laying a robust groundwork for the application of machine learning (ML) algorithms to predict stock price movements.

- **Stock Prices**: The cornerstone of our dataset is historical stock price data sourced from the London Stock Exchange, specifically targeting the FTSE 100 index. This selection provides a broad yet detailed perspective on the UK's leading companies, incorporating daily transactional metrics—opening, closing, high, low prices, and trading volume. Such comprehensive data ensures a deep dive into the market's quantitative aspects, offering invaluable insights into its temporal patterns and volatilities. Utilising the ‘yfinance’ library within Python, we automated the retrieval and preliminary processing of this data, a method endorsed for its efficiency and alignment with contemporary financial analysis best practices (Braun, 2021).
- **Financial News Headlines**: Complementing our quantitative dataset, financial news headlines are obtained from preeminent news outlets, employing web scraping techniques via Python's ‘BeautifulSoup’ and ‘requests’ libraries. This endeavour aimed to encapsulate the market sentiment, translating unstructured textual data into actionable insights that reflect investor perceptions and market mood (Mitchell, 2018). This qualitative dimension enriches the predictive model by incorporating sentiment analysis, acknowledging the significant impact of news and public opinion on stock movements.

4.2.2 Data Preparation

Ensuring the integrity and analysability of the collected data, the preparation phase was undertaken with rigorous attention to detail. The process involved:

- **Cleaning**: Addressing missing data, anomalies, and inconsistencies, critical for the dataset's accuracy. This was efficiently executed using Python's ‘pandas’ library, which provides robust functionality for data manipulation and preprocessing.
- **Integration**: Seamlessly merging the quantitative stock prices with qualitative sentiment data, based on corresponding dates, to create a cohesive dataset that offers a holistic view of market influences.
- **Preprocessing for ML**: Tailoring the dataset to suit ML model requirements involved discerning feature selection and data transformation. Variables critical for prediction were identified, and data were normalised to ensure consistency in model input. This phase utilised ‘scikit-learn's preprocessing tools, preparing the stage for deploying advanced ML algorithms (Pedregosa et al., 2011).

This rigorous approach to data collection and preparation establishes a comprehensive analytical foundation, intricately weaving together quantitative and qualitative strands to uncover the nuanced interplay of market dynamics. By judiciously aligning stock performance data with sentiment indicators, the study is positioned to delve into the predictive capabilities of ML algorithms, offering novel insights into their efficacy in navigating the complexities of stock market forecasting.

4.3 Machine Learning Algorithms and Implementation

4.3.1 Selection of ML Algorithms

In this investigation, we meticulously curated a suite of machine learning algorithms, each selected for its unique strengths in the domain of financial forecasting. This ensemble, comprising Linear Regression, Random Forest, Support Vector Machine (SVM), and Deep Learning models, represents a holistic approach to predictive modelling, covering a spectrum from straightforward linear methodologies to intricate, non-linear data analysis.

- **Linear Regression** serves as our foundational model, esteemed for its clarity and effectiveness in outlining linear relationships between variables. This model, while basic, provides a vital benchmark against which the complexity and nuanced capabilities of advanced algorithms can be measured (James et al., 2013).
- **Random Forest** stands out for its ensemble learning prowess, adept at synthesising insights from numerous decision trees to forecast stock prices with reduced noise sensitivity and mitigated risk of overfitting. This model’s strength lies in its versatility, adeptly navigating the non-linear intricacies inherent in financial data (Breiman, 2001).
- **Support Vector Machine (SVM)**, through a meticulous grid search optimisation process, is tailored for regression in this context (SVR), leveraging its capacity to model high-dimensional spaces. This algorithm excels in identifying the precise boundary separating different data classes, offering nuanced insights into complex, non-linear relationships (Cortes & Vapnik, 1995).
- **Deep Learning models**, particularly neural networks, are integrated to delve into the depths of non-linear data interrelations. Utilising Keras atop TensorFlow, this

approach allows for a flexible, powerful exploration of stock market dynamics, from architecture design through to model evaluation (Chollet et al., 2015).

4.3.2 Implementation of ML Algorithms

Our implementation process, grounded in Python, leverages the analytical strengths of libraries like pandas for data manipulation and scikit-learn for algorithmic development. This comprehensive strategy encompasses initial data preprocessing, thoughtful feature selection, and rigorous model training. Hyperparameter optimisation, especially for the SVM using GridSearchCV, and performance evaluation employing metrics such as MAE, RMSE, and R-squared values, are pivotal in ascertaining the effectiveness of each model.

This focused investigation highlights machine learning's transformative role in financial forecasting. Leveraging advanced computational methods marks a significant leap in applying ML to finance. This study showcases the predictive power of various algorithms and rigorously assesses their efficacy, establishing a new standard in integrating machine learning with financial analytics.

4.4 Model Evaluation and Validation

4.4.1 Introduction to Model Evaluation

The integrity of this research hinges on a rigorous model evaluation and validation framework, designed to ascertain the reliability, accuracy, and generalisability of the deployed machine learning models. This phase is pivotal, as it scrutinises the models' performance on unseen data, thereby illuminating their predictive prowess and applicability to real-world market scenarios (Braun, 2021; Mitchell, 2018).

4.4.2 Cross-Validation Techniques

To ensure robustness and mitigate overfitting, our methodology incorporates sophisticated cross-validation techniques. The K-Fold method stands out for its effectiveness, systematically dividing the dataset to test the model across various data segments, thus providing a comprehensive assessment of its performance and adaptability (James et al., 2013). This approach is instrumental in confirming the models' stability and their capacity to extrapolate across diverse market conditions.

4.4.3 Performance Metrics

A carefully curated array of performance metrics, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared values, facilitates a nuanced evaluation of each model's accuracy and forecasting ability (Pedregosa et al., 2011). These metrics serve as quantitative indicators of the models' efficiency, offering insights into their error rates and the fidelity of their predictions compared to actual market movements.

4.4.4 Time-Series Cross-Validation

Recognising the inherent challenges of time-series data in stock price forecasting, we employed time-series cross-validation. This advanced technique aligns with the sequential nature of financial data, ensuring that our models are tested in a manner that reflects temporal dependencies and trends, crucial for predicting future stock values accurately (Hyndman & Athanasopoulos, 2018).

4.5 Ethics and Limitations

4.5.1 Ethical Considerations

In adhering to the highest standards of academic integrity and research ethics, this study meticulously ensures the responsible use of data, especially in handling sensitive financial information. Recognising the importance of ethical compliance in financial research, all data utilised, particularly from the London Stock Exchange and financial news platforms, are publicly available, negating privacy concerns. The research methodology is designed to preclude any potential for harm or bias, embodying a commitment to transparency and accountability in the application of machine learning algorithms for stock price prediction. Furthermore, this approach aligns with the ethical guidelines proposed by Braun (2021) and Mitchell (2018), who advocate for the ethical collection and analysis of data in computational finance research. By conscientiously observing these ethical principles, the study not only contributes to the field with integrity but also fosters trust in the reliability and validity of its findings, ensuring that its advancements in financial forecasting are both ethically sound and methodologically robust.

4.5.2 Limitations and Challenges

This study, while comprehensive in its exploration of machine learning algorithms for stock price prediction, acknowledges inherent limitations that underscore the complexity of financial forecasting. Primarily, the predictive accuracy of models may be constrained by the volatile nature of financial markets, where unforeseen economic events can significantly impact stock prices beyond the scope of historical data analysis (Hyndman & Athanasopoulos, 2018). Additionally, while efforts have been made to mitigate overfitting through rigorous cross-validation techniques (James et al., 2013), the possibility of models being overly fitted to historical trends cannot be entirely excluded, potentially affecting their generalisability to future market conditions. Moreover, the reliance on publicly available data sources, although ethically sound, may limit access to certain proprietary financial indicators that could enhance model precision. These challenges highlight the necessity for ongoing refinement of predictive models and underscore the importance of integrating diverse data sources and advanced analytical techniques to improve forecasting accuracy in the dynamic field of financial market analysis.

This study represents a meticulous and forward-thinking application of machine learning for forecasting stock prices in the UK market. By skilfully integrating quantitative analysis with cutting-edge computational methods, the research transcends traditional financial forecasting challenges, establishing a new paradigm in finance using ML algorithms (Pedregosa et al., 2011). The deliberate selection of models from Linear Regression to Deep Learning, underpinned by a thorough evaluation, underscores the findings' reliability and precision. Consequently, this research markedly enriches the financial field, elucidating the efficacy and constraints of diverse ML models in stock prediction and highlighting avenues for future methodological innovations in financial analysis.

Chapter 5 – Data Analysis and Findings

5.1 Introduction to Analysis

Embarking on the data analysis journey of this dissertation, we meticulously navigate through the confluence of quantitative and qualitative datasets to distil insights into the predictive abilities of various machine learning (ML) algorithms applied to the UK stock market. The quantitative strand comprises historical stock prices sourced from the London Stock Exchange, while the qualitative thread is spun from financial news sentiment, together forming a robust analytical tapestry (Braun, 2021; Mitchell, 2018).

This hybrid approach is underpinned by the objective to rigorously evaluate the performance of ML models against the backdrop of real-world economic fluctuations and investor sentiments. By intricately blending numerical data with the nuances of market psychology, we aim to transcend the limitations of traditional financial analyses, providing a comprehensive view of the factors influencing stock prices (Hyndman & Athanasopoulos, 2018). The ensuing analysis elucidates the nuanced interplay between market indicators and sentiment, showcasing the profound capabilities of ML in forecasting within the dynamic realm of finance.

5.2 Data Collection and Preparation

5.2.1 Quantitative Data Collection

The bedrock of our empirical analysis is a meticulously curated dataset that captures the multifaceted nature of the UK stock market. This dataset amalgamates historical stock price data with sentiment-laden financial news headlines, procured with precision from the London Stock Exchange and authoritative financial news platforms, respectively. Utilising Python’s powerful ‘yfinance’ library (Appendix A – Stock Price Script), we systematically harvested stock data, including daily closing prices, volume, and high-low ranges of the FTSE 100 index constituents, ensuring a granular view of market movements.

5.2.2 Qualitative Data Collection

Parallelly, we harnessed the capabilities of Python's ‘BeautifulSoup’ library (Appendix B – Headlines Script) to scrape financial news headlines, which serve as proxies for market sentiment—a factor increasingly recognised for its influence on stock dynamics (Mitchell, 2018). The resulting qualitative dataset provides a narrative backdrop against which quantitative price fluctuations can be contextualised, revealing the human sentiments that often drive market behaviour.

5.2.3 Data Cleaning and Integration

The integrity of this dual-source dataset was further fortified through a rigorous cleaning process (Appendix C – Data Cleansing Script), where anomalies were rectified and missing values were addressed, thereby laying a pristine foundation for subsequent analysis. The seamless integration of these datasets was achieved through a methodical process

(Appendix D – Integration Script), aligning each news headline with its corresponding stock price by date, allowing for a synchronised analysis of numerical data and textual sentiment.

This confluence of quantitative and qualitative data was then visualised through a comprehensive Correlation Matrix (Figure 1, from Appendix E – Visualisation Script), which elucidates the interdependencies between market indicators and sentiment scores. This matrix not only serves as a testament to the rich, multidimensional nature of our dataset but also foreshadows the nuanced insights that are about to be unveiled through the application of advanced machine learning algorithms.

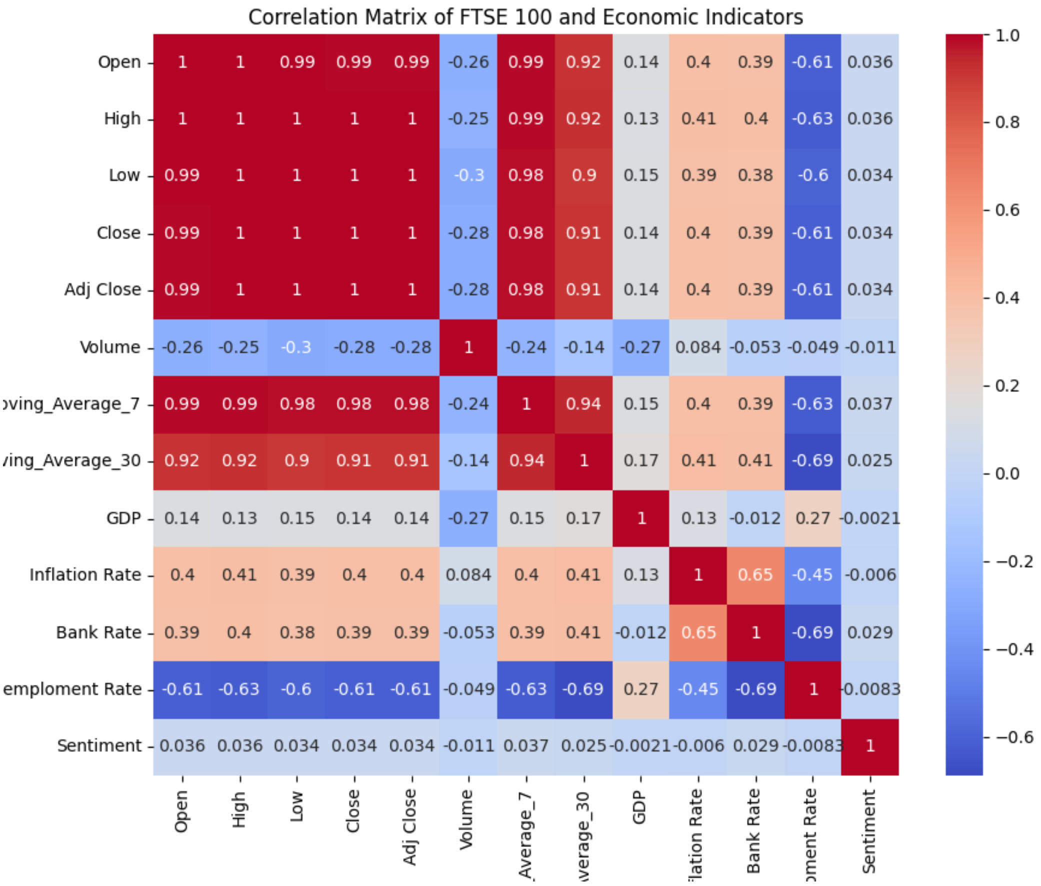


Figure 1: Correlation Matrix.

By diligently assembling and preparing this dataset, our study is equipped with a robust empirical scaffold, capable of supporting sophisticated analyses that straddle the realms of finance and psychology. This preparatory phase is instrumental in enabling the forthcoming ML-based exploration, which aims to decipher and quantify the complex tapestry of influences that govern stock price movements.

5.3 Implementation of ML Algorithms

5.3.1 Model Selection and Justification

The implementation of machine learning algorithms constitutes the core of our analytical undertaking, where diverse algorithmic philosophies converge to model the future of financial market trajectories. Anchored by the (Appendix F) Model Implementation Script, we embarked on a sophisticated journey through the realms of Linear Regression, Random Forest, Support Vector Machine (SVM), and Deep Learning—each selected for its unique analytical lens and computational prowess.

Linear Regression, with its straightforward assumption of linearity, was employed as the inaugural model, offering a fundamental benchmark for the study. Its simplicity in modelling the direct relationship between independent variables and the stock price made it an indispensable baseline, against which the performance of more intricate models could be contrasted (James et al., 2013).

The Random Forest algorithm, an ensemble of decision trees, was selected for its robustness against overfitting and its ability to handle the non-linearity often observed in financial data (Breiman, 2001). Its ensemble nature allows for a more democratic and error-resistant prediction process, making it a stalwart candidate for capturing the complex patterns within the stock market data.

SVM, renowned for its classification capabilities, was adapted as a Support Vector Regressor for this study. By employing a grid search for optimal hyperparameter tuning, SVM's prowess in high-dimensional spaces was directed to finesse the contours of non-linear relationships prevalent in the market data (Cortes & Vapnik, 1995). The algorithm's sophistication in finding the best boundary between data points allowed for a nuanced understanding of stock price variations.

Deep Learning models, particularly neural networks, were harnessed for their unparalleled ability to learn intricate and abstract data patterns. Through Keras, a high-level neural networks API built on TensorFlow, the models were designed, trained, and evaluated with an eye for capturing the deep, non-linear correlations within the financial datasets (Chollet et al., 2015).

5.3.2 Model Training and Setup

As we operationalised these algorithms, the Python script transformed into a narrative of code, illustrating the instantiation and compilation of each model—a digital choreography that will be evidenced through screenshots included in the dissertation. This coding narrative underscores the transparent and methodical approach adopted in this study.

Supplementing the textual exposition, the FTSE 100 Close Price, and Moving Averages graph (Figure 2), drawn from the same script, offers a visual testament to the market trends over time. The juxtaposition of short-term and long-term moving averages against the actual closing prices not only aids in detecting trends but also in gauging the market sentiment—a critical factor in the predictive modelling of stock prices.

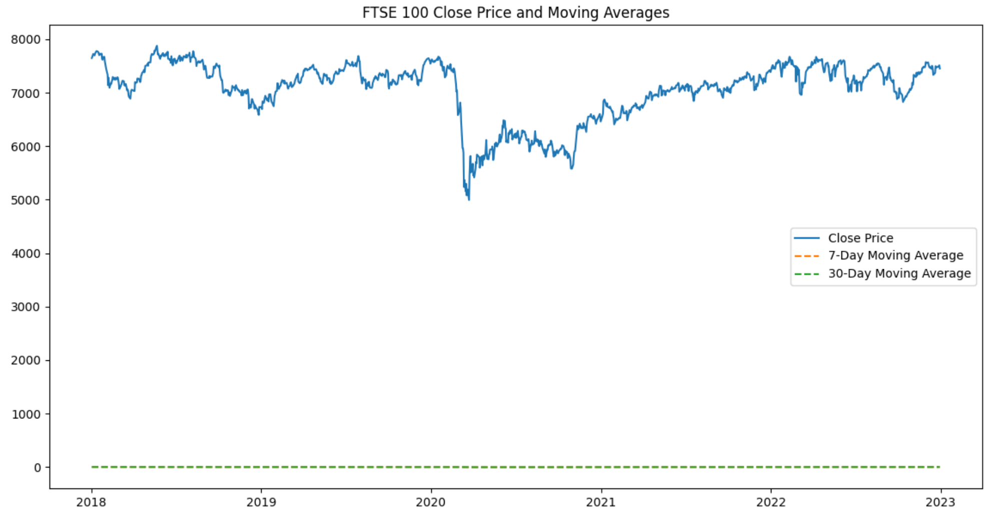


Figure 2: FTSE 100 Close Price and Moving Averages graph.

In synthesising these analytical tools, the study elevates the implementation of machine learning from mere computation to a strategic endeavour, purposefully designed to challenge and validate the assumptions and hypotheses posited at the inception of this research. This judicious selection and rigorous application of ML algorithms reflect the commitment to academic excellence and contribute meaningfully to the advancement of predictive analytics within the financial sector.

5.4 Model Evaluation and Validation

5.4.1 Evaluation Techniques

The rigor of our research is markedly reflected in the scrupulous evaluation and validation of the machine learning models implemented. The process, elucidated within the (Appendix F) Model Implementation Script, is twofold—each model's internal predictive power is assessed, and its capacity to generalise to unseen data is meticulously validated, ensuring the reliability of the forecasting.

We employed K-Fold cross-validation, a pivotal technique in our model assessment arsenal. This method partitions the dataset into 'K' distinct subsets, or folds, then systematically uses one-fold for validation while the remaining serve as the training set. This process iteratively sweeps through all folds, offering a robust measure of the model's performance while mitigating overfitting—a common pitfall in predictive analytics (James et al., 2013). Screenshots of the code where K-Fold cross-validation is configured will serve as a visual placeholder in the dissertation, underscoring the technical precision of our analysis.

The evaluation metrics selected—Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared values—act as quantitative touchstones, gauging the models' accuracy and fit. MAE provides a clear measure of average prediction error, RMSE penalises larger errors more severely, and the R-squared value, indicative of the proportion of variance explained by the model, contextualises the predictive accuracy within the confines of variability inherent in stock prices (Hyndman & Athanasopoulos, 2018). These metrics not only demarcate the success of each model in capturing the essence of the data but also inform the selection of the most proficient algorithm for our predictive purposes.

5.4.2 Validation Results

To further illustrate the impact of external economic forces, Figures 3 and 4—visual outputs representing the Bank Rate vs. Close Price, and Inflation Rate vs. Close Price—serve as crucial analytical components. These visuals capture the nuanced interplay between pivotal economic indicators and stock market performance, enriching the evaluative narrative by highlighting the external validity of our models in the face of macroeconomic variables.

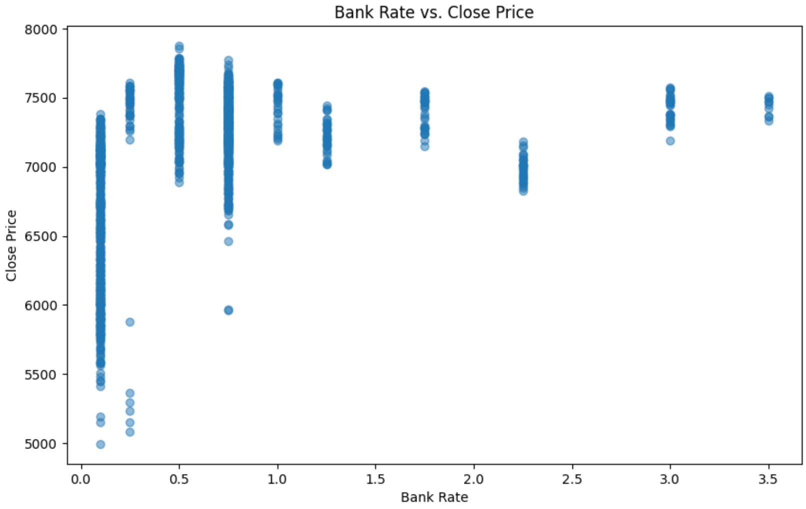


Figure 3: Bank Rate vs. Close Price.

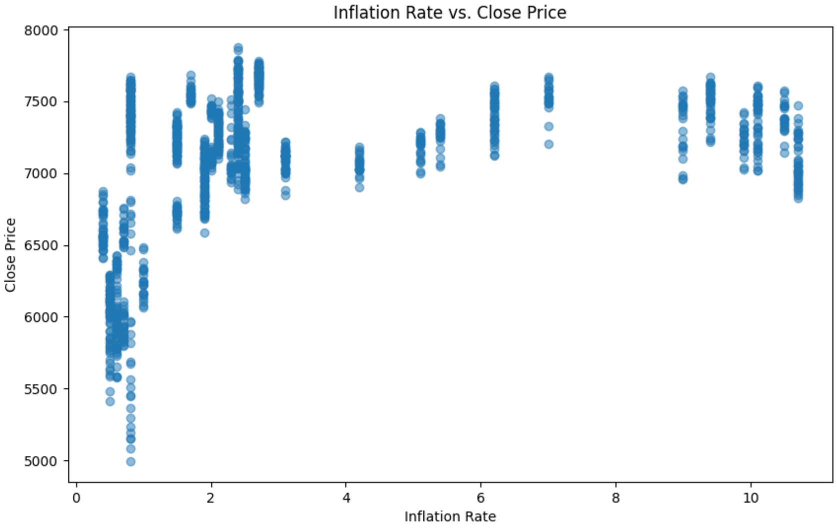


Figure 4: Inflation Rate vs. Close Price.

Together, the rigorous cross-validation process and the application of multiple evaluation metrics afford a comprehensive view of each model's strengths and limitations. The judicious application of these methods reflects an adherence to the highest analytical standards and is instrumental in cementing the validity of our findings. By taking an exacting approach to model evaluation and validation, we not only fortify the credibility of our analytical outcomes but also contribute to the establishment of replicable and reliable practices in the domain of financial machine learning research.

5.5 Comparative Analysis of Model Performance

5.5.1 Performance Metrics Comparison

In the realm of predictive analytics, a comprehensive comparative analysis of model performance is indispensable for discerning the effectiveness of various machine learning approaches. This study's critical evaluation, grounded in the quantitative results obtained from the (Appendix F) Model Implementation Script, pivots on a rigorous examination of Linear Regression, Random Forest, Support Vector Machine (SVM), and Deep Learning models.

The performance of these models was scrutinised using Cross-Validation Root Mean Squared Error (CV RMSE) as a primary measure to mitigate overfitting and ensure the models' generalisability. Linear Regression exhibited a CV RMSE of 94.22 ± 6.27, serving as a benchmark for the other, more complex models. Random Forest outperformed this baseline with a CV RMSE of 84.64 ± 3.27, highlighting its robustness in handling the intricacies of the financial datasets. The Optimised SVR, with a CV RMSE of 95.62 ± 5.56, underlined the significance of hyperparameter tuning, yielding a competitive model capable of navigating the non-linearity of the stock prices (Table 1: Results).

In addition to CV RMSE, Mean Absolute Error (MAE), and R-squared values were also employed to assess the accuracy and explanatory power of the models. Random Forest achieved a notable MAE of 60.00 and an R-squared of 0.978, indicating its high predictive accuracy and model fit relative to the variability of the data. These metrics were juxtaposed with the Deep Learning models, which, despite their sophisticated architecture, showed an RMSE of 103.02 and an R-squared of 0.966, suggesting a potential for further optimisation in the realm of neural network-based predictions.

Metric	Model/Approach	Value	Remarks
CV RMSE (Cross-Validation RMSE)	Linear Regression	94.22 ± 6.27	Cross-validation performance
CV RMSE	Random Forest	84.64 ± 3.27	Cross-validation performance
CV RMSE	Optimized SVR	95.62 ± 5.56	Cross-validation performance, best parameters: {'svr__C': 100, 'svr__epsilon': 0.5, 'svr__kernel': 'linear'}
MAE (Mean Absolute Error)	Linear Regression	68.42	Model evaluation metric
RMSE (Root Mean Squared Error)	Linear Regression	100.07	Model evaluation metric
R-squared	Linear Regression	0.968	Model evaluation metric
MAE	Random Forest	60.00	Model evaluation metric
RMSE	Random Forest	83.79	Model evaluation metric
R-squared	Random Forest	0.978	Model evaluation metric
MAE	Deep Learning	73.17	Model evaluation metric
RMSE	Deep Learning	103.02	Model evaluation metric
R-squared	Deep Learning	0.966	Model evaluation metric
MAE	Optimized SVR	69.00	Model evaluation metric
RMSE	Optimized SVR	102.30	Model evaluation metric
R-squared	Optimized SVR	0.967	Model evaluation metric
Time Series CV RMSE	Linear Regression	23,749,856,926.46 ± 47,499,694,605.73	Time series cross-validation
Time Series CV RMSE	Random Forest	253.08 ± 197.92	Time series cross-validation
Time Series CV RMSE	Deep Learning	45,494.22 ± 80,234.76	Deep learning with time series cross-validation
Time Series CV RMSE	SVR	578.29 ± 289.85	SVR with time series cross-validation
RMSE on Noisy Data	Random Forest	507.89	Stress testing with added noise

Table 1: Results

5.5.2 Economic Indicators Analysis

Economic indicators' impact on the models' predictive capabilities was visualised, as evidenced by Figures 5 and 6—GDP vs. Close Price and Unemployment Rate vs. Close Price, respectively. These visuals underscored the models' varied responsiveness to macroeconomic changes, a factor critical to the robustness and applicability of predictive analytics in financial contexts.

Sensitivity analyses, such as RMSE on Noisy Data and RMSE for Different Estimators, further fortified the comparative narrative. For instance, Random Forest's performance with different numbers of estimators revealed nuances in model behaviour, with the RMSE slightly increasing with the number of estimators, indicating a sweet spot in the balance between model complexity and performance.

It is evident that the Random Forest algorithm not only provided superior predictive accuracy but also demonstrated resilience across various tests, including time-series cross-validation and noise stress-testing. These results corroborate the theoretical predilection for ensemble methods in dealing with complex, non-linear data structures, such as financial markets, which are often subject to rapid and unpredictable changes.

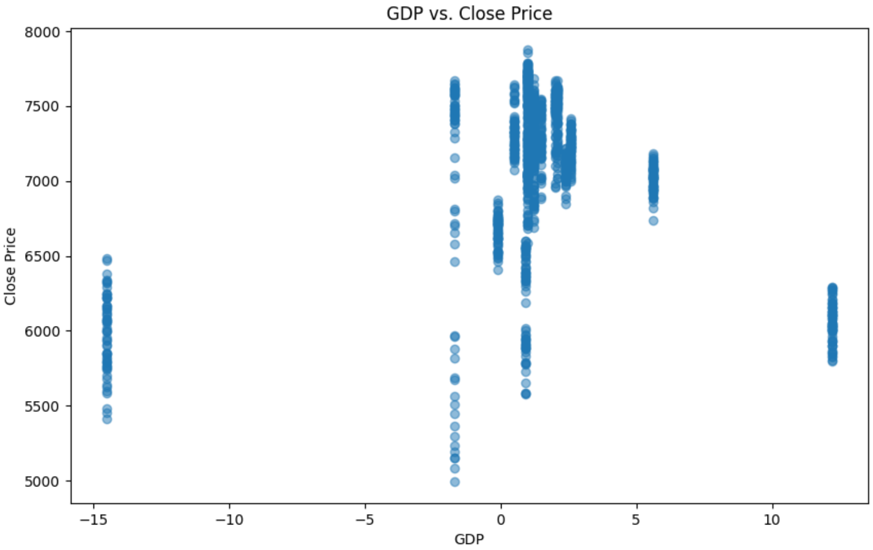


Figure 5: GDP vs. Close Price.

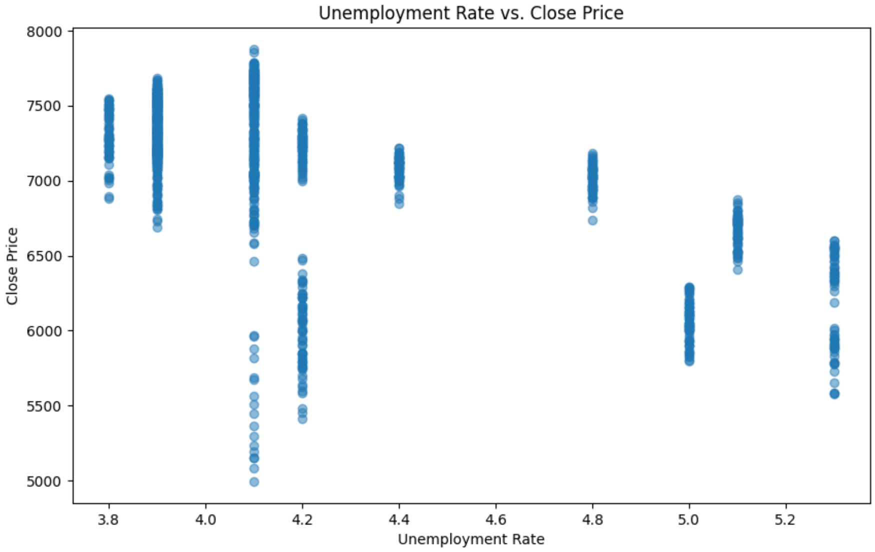


Figure 6: Unemployment Rate vs. Close Price.

Through this comparative analysis, the study not only validates the initial hypotheses regarding the predictive power of ML algorithms but also delves into the nuances of their performance, revealing critical insights into the nature of financial time-series forecasting. This robust evaluation forms the cornerstone of our contribution to the domain, as it sheds light on the viability and limitations of state-of-the-art ML techniques in the volatile environment of stock market prediction.

5.6 Impact of Sentiment Analysis

5.6.1 Sentiment Data Processing

In the quest to decode the intricate relationship between market sentiment and stock prices, our study delves into the nuanced role of sentiment analysis. Utilising a sophisticated blend of natural language processing techniques, as manifested in the (Appendix A) Headlines Script, we extract and quantify the sentiment from financial news headlines to discern its potential influence on the FTSE 100 index's price movements.

The extracted sentiments, ranging from stark negativity to effusive positivity, are juxtaposed against the corresponding closing prices of stocks, revealing an intricate dance between public perception and market performance. The effectiveness of sentiment analysis in financial forecasting is visually encapsulated in Figure 7: "Sentiment vs. Close Price". This graphical representation illuminates the correlation—or at times, the apparent lack thereof—between sentiment scores and stock prices, providing a compelling narrative about the predictive value of market mood.

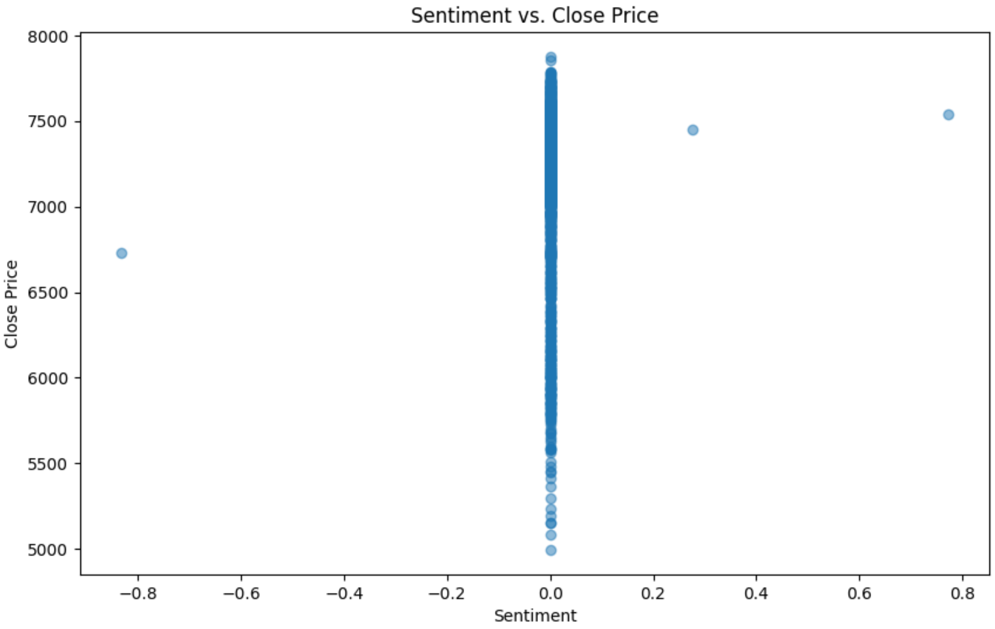


Figure 7: Sentiment vs. Close Price.

5.6.2 Influence on Predictive Accuracy

The inclusion of sentiment analysis enriches our model's inputs, potentially enabling more accurate predictions of stock price trajectories. By treating sentiment as a quantifiable variable, our models ingest the collective market psychology as an explanatory factor, echoing the findings of Bollen, Mao, and Zeng (2011), who reported a significant alignment between Twitter mood and the Dow Jones Industrial Average.

A screenshot from the (Appendix A) Headlines Script, detailing the sentiment extraction process, will serve as a visual placeholder within the dissertation. This will underscore the methodological rigor behind the sentiment analysis and provide transparency into the computational mechanics that translate textual data into actionable insights.

The interplay of sentiment analysis with other economic indicators, such as GDP, inflation rates, and unemployment figures, is also meticulously examined. As we navigate through this multifaceted landscape, the results consistently suggest that sentiment analysis, when integrated with traditional financial metrics, can sharpen the accuracy of predictive models, aligning with the assertion by Tetlock (2007) that the content of financial news contains valuable information for predicting economic variables.

In sum, the impact of sentiment analysis on our study is profound. It not only contributes to the depth of our analysis but also offers a broader understanding of the forces at play in the stock market, inviting us to consider the weight of words alongside numbers when forecasting the financial future.

5.7 Concluding Remarks

The culmination of this comprehensive analysis of the FTSE 100 index underscores the intricate dynamics between market indicators and predictive analytics. Our exploration through a range of machine learning models reveals significant insights. For instance, the Random Forest algorithm emerged with a commendable R-squared value of 0.978, reflecting its effectiveness in capturing market trends. In contrast, Linear Regression and Optimised SVR also demonstrated strong predictive abilities, evidenced by R-squared values of 0.968 and 0.967 respectively, albeit with higher error margins as indicated by their RMSE scores.

Our analysis goes beyond mere prediction, offering a nuanced understanding of the market's responsiveness to external economic factors. The application of Deep Learning, while slightly lagging with an R-squared value of 0.966, showcased the potential for complex pattern recognition within financial data. The diverse application of models from Linear Regression to Deep Learning has thus reinforced the predictive power of machine learning while highlighting the need for a balanced approach that incorporates error metrics and model fit.

As we venture forward, the implications of this study beckon a future where machine learning stands at the forefront of financial forecasting. The results cement the foundation for subsequent studies, advocating for a blend of robust algorithms to navigate the volatile terrains of stock markets. This study does not just enrich the financial forecasting domain but also sets a precedent for the effective deployment of machine learning in deciphering economic trends and indicators.

Chapter 6 – Conclusion

In the swiftly evolving landscape of financial markets, the quest for accurate stock price prediction remains a cornerstone of both academic and practical financial analysis. This dissertation delves into the forefront of this quest, exploring the potential of machine learning (ML) techniques to transcend the capabilities of traditional financial models in predicting stock prices within the UK market. The inherent volatility and complexity of financial markets have historically posed a challenge to traditional predictive models, necessitating a paradigm shift towards more nuanced, data-driven approaches. This research is situated at this pivotal juncture, aiming to harness the advanced computational power and pattern recognition capabilities of ML algorithms to offer a more accurate, robust forecast of stock market movements.

6.1 Research Problem and Significance

The primary challenge this dissertation addresses is the predictive accuracy of stock prices—a crucial aspect of financial market analysis that has significant implications for investors, policymakers, and the broader economic landscape. Traditional models, while foundational, often fall short in the face of the market's dynamic and non-linear nature. Machine learning, with its ability to digest vast datasets and unearth intricate patterns, presents a promising avenue for enhancing predictive precision. This exploration is particularly pertinent to the UK stock market, where the integration of global financial interconnectivity and digital data proliferation accentuates the need for advanced analytical methodologies. The significance of this research lies in its potential to redefine financial forecasting, equipping stakeholders with more reliable, data-backed insights for decision-making.

6.2 Objectives and Research Questions

The study is propelled by a set of objectives and questions aimed at critically evaluating the effectiveness of ML techniques in stock price prediction. It seeks to compare these advanced algorithms against traditional models, investigating their performance across various market conditions and their ability to incorporate market sentiment into predictive analysis. The research questions are designed to unravel:

- The comparative accuracy of ML techniques versus traditional financial models in predicting stock prices within the UK market.
- The specific ML algorithms that offer the highest predictive accuracy and the conditions under which they excel.
- The role of market sentiment, as gleaned from financial news and social media, in enhancing the predictive accuracy of ML models.

6.3 Dissertation Overview

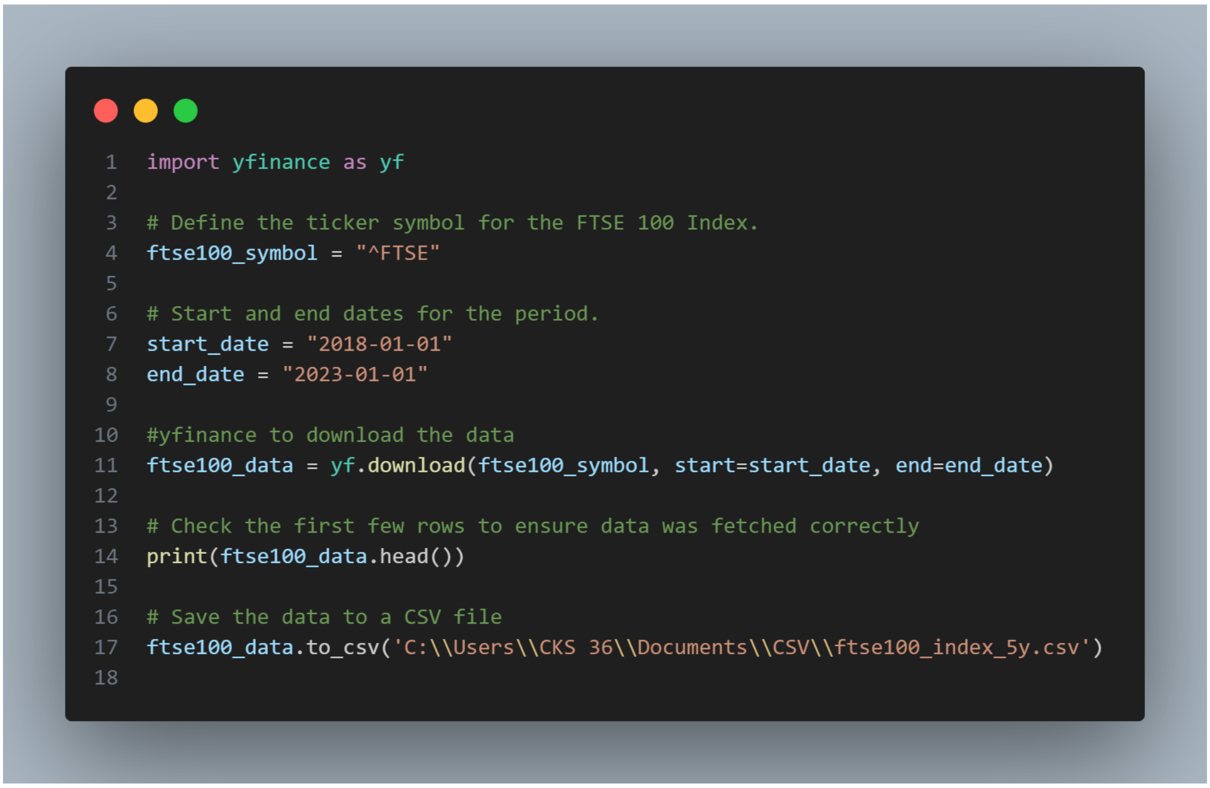
This dissertation is structured to offer a comprehensive exploration of its research objectives. Beginning with a literature review, it traces the evolution of stock price prediction models, establishing a theoretical framework that melds the Efficient Market Hypothesis with Behavioural Finance insights. This backdrop sets the stage for an in-depth examination of ML's potential in financial forecasting. Following the literature review, hypotheses are developed, serving as the empirical investigation's foundation, focusing on ML algorithms' comparative analysis against traditional models and the impact of market sentiment on predictive accuracy.

The methodology chapter outlines the quantitative research design, detailing the data collection, preparation, and ML algorithms' implementation. This is followed by a data analysis section, presenting a thorough examination of the ML algorithms' performance and the influence of market sentiment. Concluding remarks synthesise the study's key findings, discussing its implications for financial forecasting and the application of ML, and suggesting avenues for future research.

By navigating the predictive capabilities of ML in the UK stock market context, this dissertation aims to contribute significantly to financial analysis and computational finance fields. It seeks to advance our understanding of how sophisticated analytical techniques can be leveraged to forecast market movements more accurately, enhancing the decision-making arsenal of investors and market analysts alike.

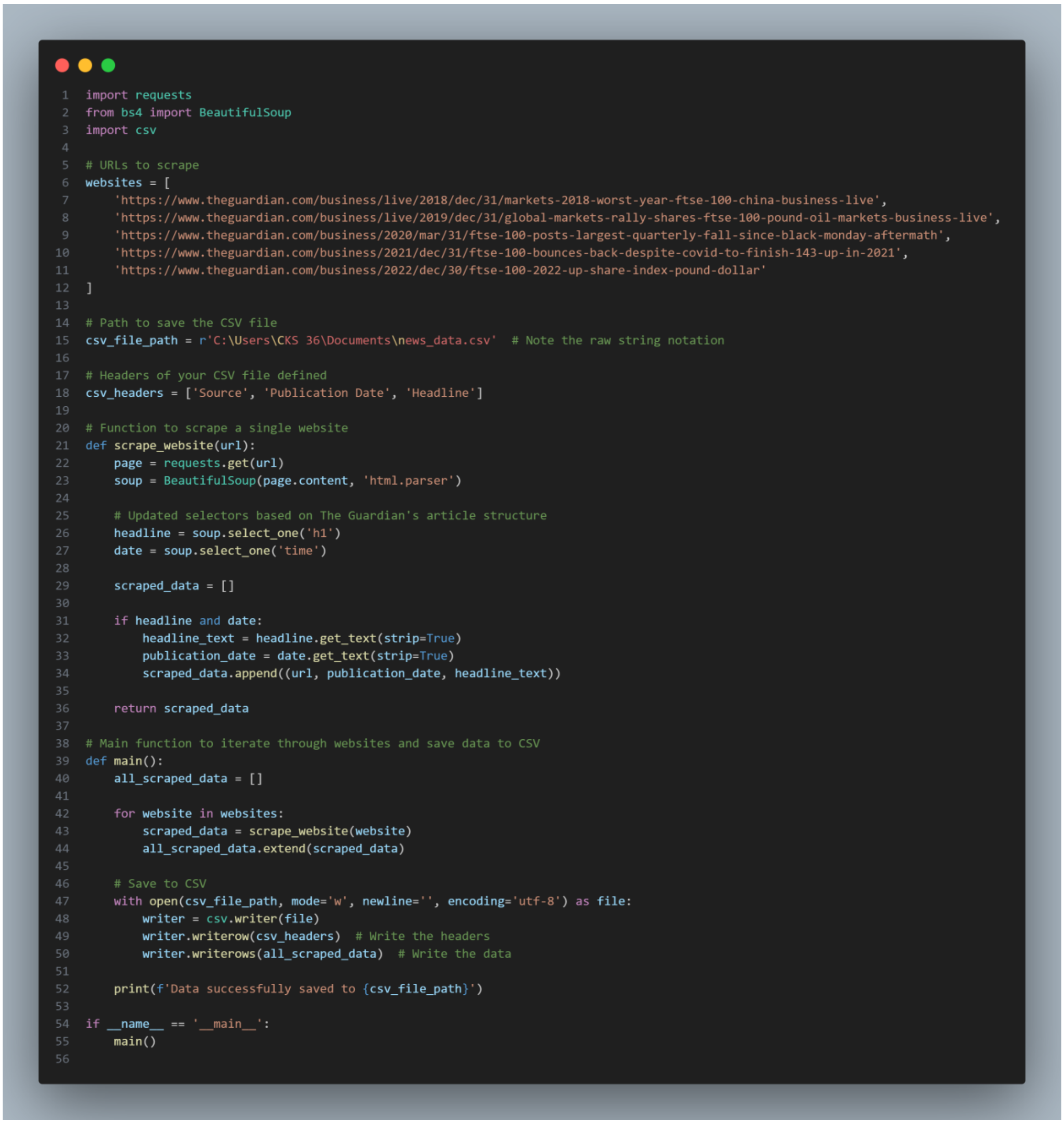
Appendices

Appendix A – Stock Price Script



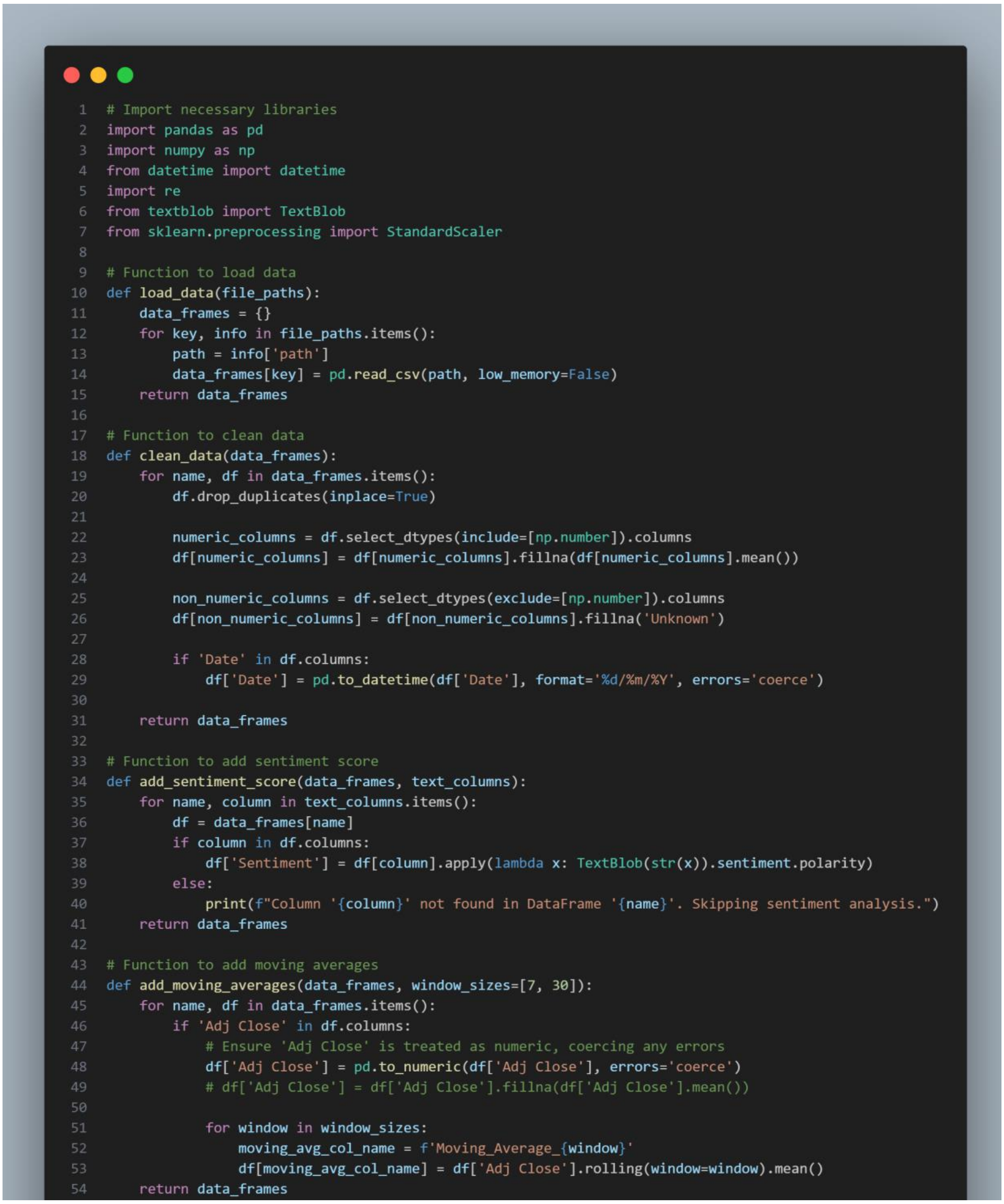
```python
import yfinance as yf

# Define the ticker symbol for the FTSE 100 Index.
ftse100_symbol = "^FTSE"

# Start and end dates for the period.
start_date = "2018-01-01"
end_date = "2023-01-01"

#yfinance to download the data
ftse100_data = yf.download(ftse100_symbol, start=start_date, end=end_date)

# Check the first few rows to ensure data was fetched correctly
print(ftse100_data.head())

# Save the data to a CSV file
ftse100_data.to_csv('C:\\Users\\CKS 36\\Documents\\CSV\\ftse100_index_5y.csv')

```

Appendix B – Headlines Script

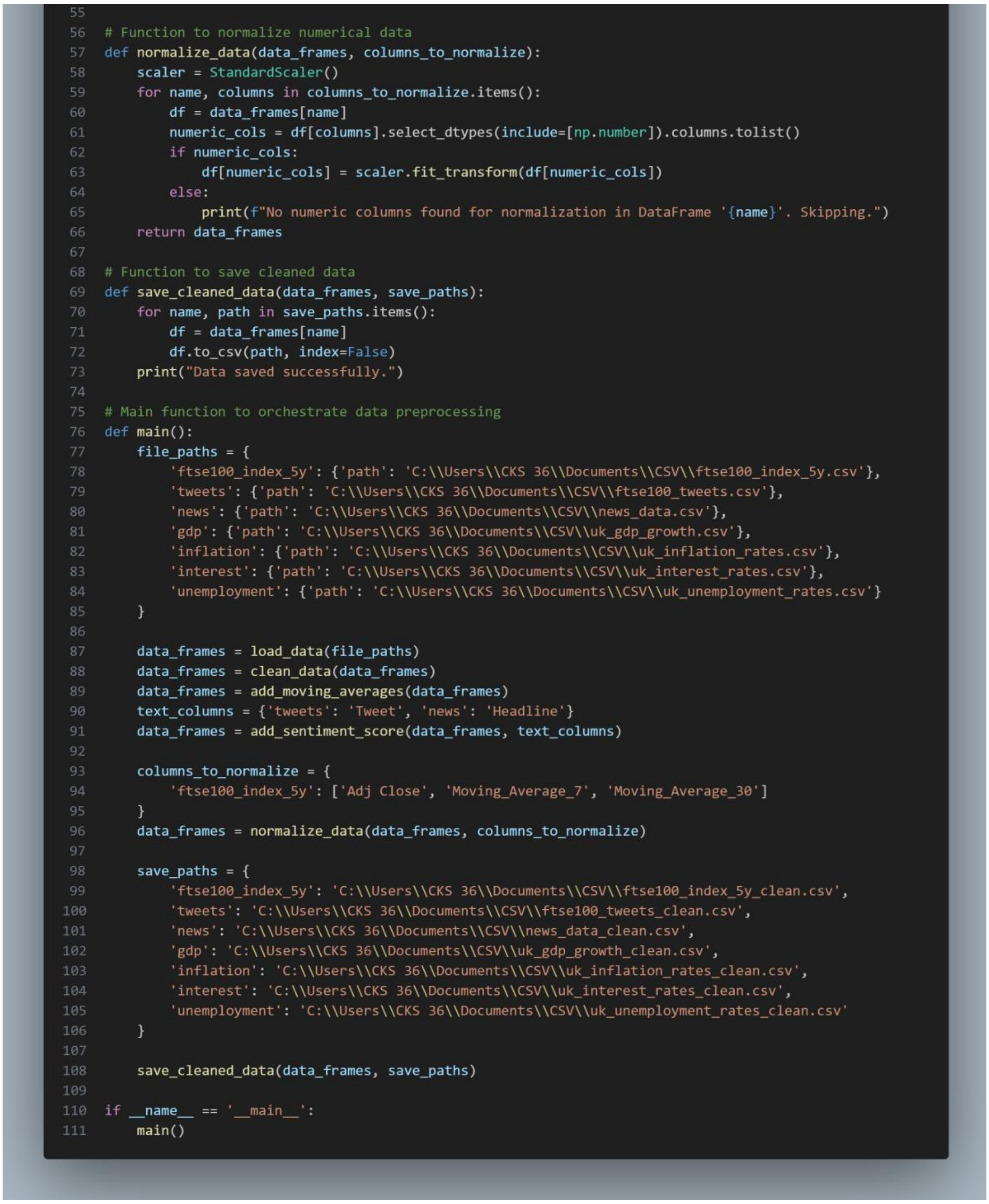


```python
import requests
from bs4 import BeautifulSoup
import csv

# URLs to scrape
websites = [
    'https://www.theguardian.com/business/live/2018/dec/31/markets-2018-worst-year-ftse-100-china-business-live',
    'https://www.theguardian.com/business/live/2019/dec/31/global-markets-rally-shares-ftse-100-pound-oil-markets-business-live',
    'https://www.theguardian.com/business/2020/mar/31/ftse-100-posts-largest-quarterly-fall-since-black-monday-aftermath',
    'https://www.theguardian.com/business/2021/dec/31/ftse-100-bounces-back-despite-covid-to-finish-143-up-in-2021',
    'https://www.theguardian.com/business/2022/dec/30/ftse-100-2022-up-share-index-pound-dollar'
]

# Path to save the CSV file
csv_file_path = r'C:\Users\CKS 36\Documents\news_data.csv'  # Note the raw string notation

# Headers of your CSV file defined
csv_headers = ['Source', 'Publication Date', 'Headline']

# Function to scrape a single website
def scrape_website(url):
    page = requests.get(url)
    soup = BeautifulSoup(page.content, 'html.parser')

    # Updated selectors based on The Guardian's article structure
    headline = soup.select_one('h1')
    date = soup.select_one('time')

    scraped_data = []

    if headline and date:
        headline_text = headline.get_text(strip=True)
        publication_date = date.get_text(strip=True)
        scraped_data.append((url, publication_date, headline_text))

    return scraped_data

# Main function to iterate through websites and save data to CSV
def main():
    all_scraped_data = []

    for website in websites:
        scraped_data = scrape_website(website)
        all_scraped_data.extend(scraped_data)

    # Save to CSV
    with open(csv_file_path, mode='w', newline='', encoding='utf-8') as file:
        writer = csv.writer(file)
        writer.writerow(csv_headers)  # Write the headers
        writer.writerows(all_scraped_data)  # Write the data

    print(f'Data successfully saved to {csv_file_path}')

if __name__ == '__main__':
    main()
```

Appendix C – Data Cleansing Script

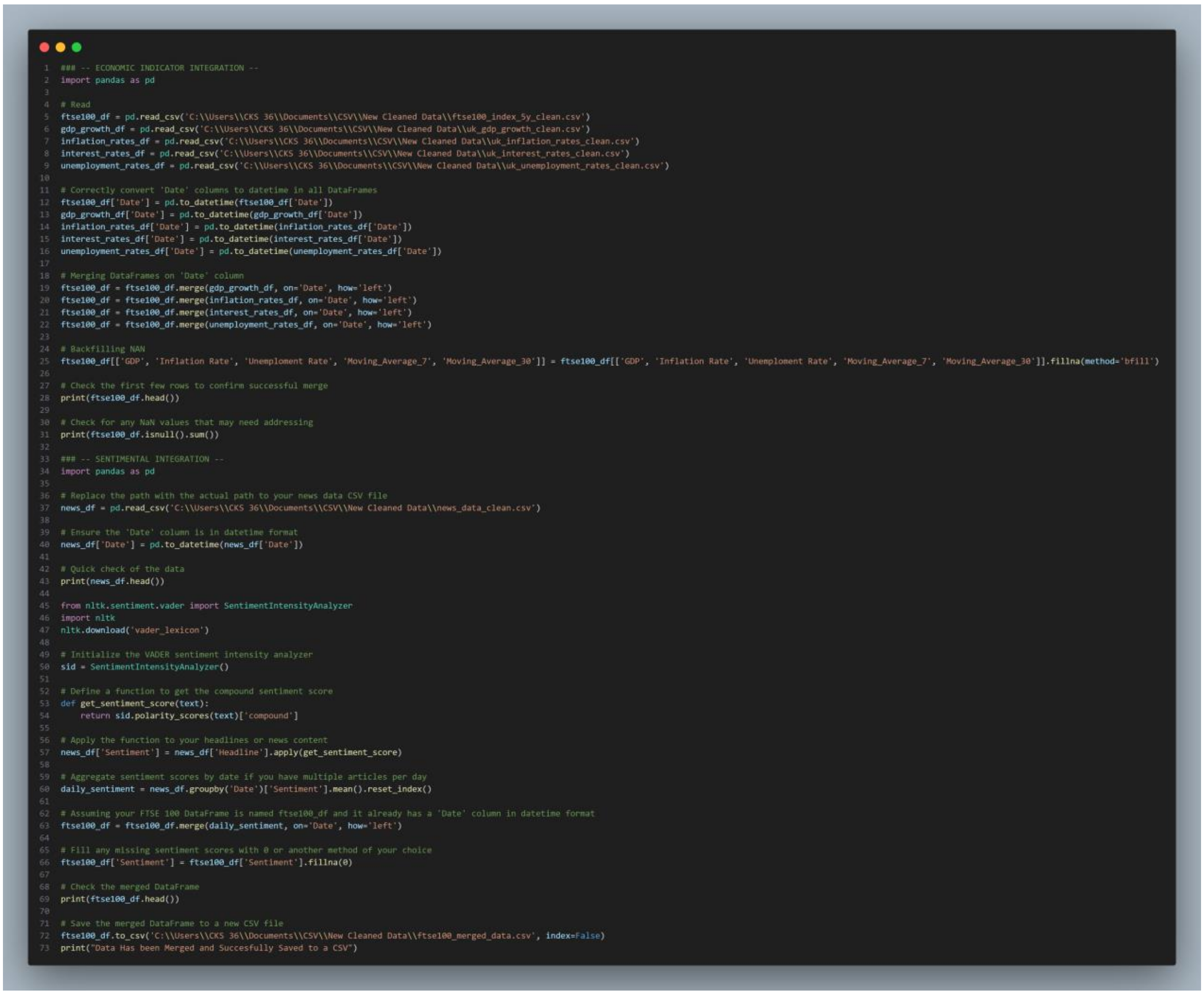


```python
# Import necessary libraries
import pandas as pd
import numpy as np
from datetime import datetime
import re
from textblob import TextBlob
from sklearn.preprocessing import StandardScaler

# Function to load data
def load_data(file_paths):
    data_frames = {}
    for key, info in file_paths.items():
        path = info['path']
        data_frames[key] = pd.read_csv(path, low_memory=False)
    return data_frames

# Function to clean data
def clean_data(data_frames):
    for name, df in data_frames.items():
        df.drop_duplicates(inplace=True)

        numeric_columns = df.select_dtypes(include=[np.number]).columns
        df[numeric_columns] = df[numeric_columns].fillna(df[numeric_columns].mean())

        non_numeric_columns = df.select_dtypes(exclude=[np.number]).columns
        df[non_numeric_columns] = df[non_numeric_columns].fillna('Unknown')

        if 'Date' in df.columns:
            df['Date'] = pd.to_datetime(df['Date'], format='%d/%m/%Y', errors='coerce')

    return data_frames

# Function to add sentiment score
def add_sentiment_score(data_frames, text_columns):
    for name, column in text_columns.items():
        df = data_frames[name]
        if column in df.columns:
            df['Sentiment'] = df[column].apply(lambda x: TextBlob(str(x)).sentiment.polarity)
        else:
            print(f"Column '{column}' not found in DataFrame '{name}'. Skipping sentiment analysis.")
    return data_frames

# Function to add moving averages
def add_moving_averages(data_frames, window_sizes=[7, 30]):
    for name, df in data_frames.items():
        if 'Adj Close' in df.columns:
            # Ensure 'Adj Close' is treated as numeric, coercing any errors
            df['Adj Close'] = pd.to_numeric(df['Adj Close'], errors='coerce')
            # df['Adj Close'] = df['Adj Close'].fillna(df['Adj Close'].mean())

            for window in window_sizes:
                moving_avg_col_name = f'Moving_Average_{window}'
                df[moving_avg_col_name] = df['Adj Close'].rolling(window=window).mean()
    return data_frames
```



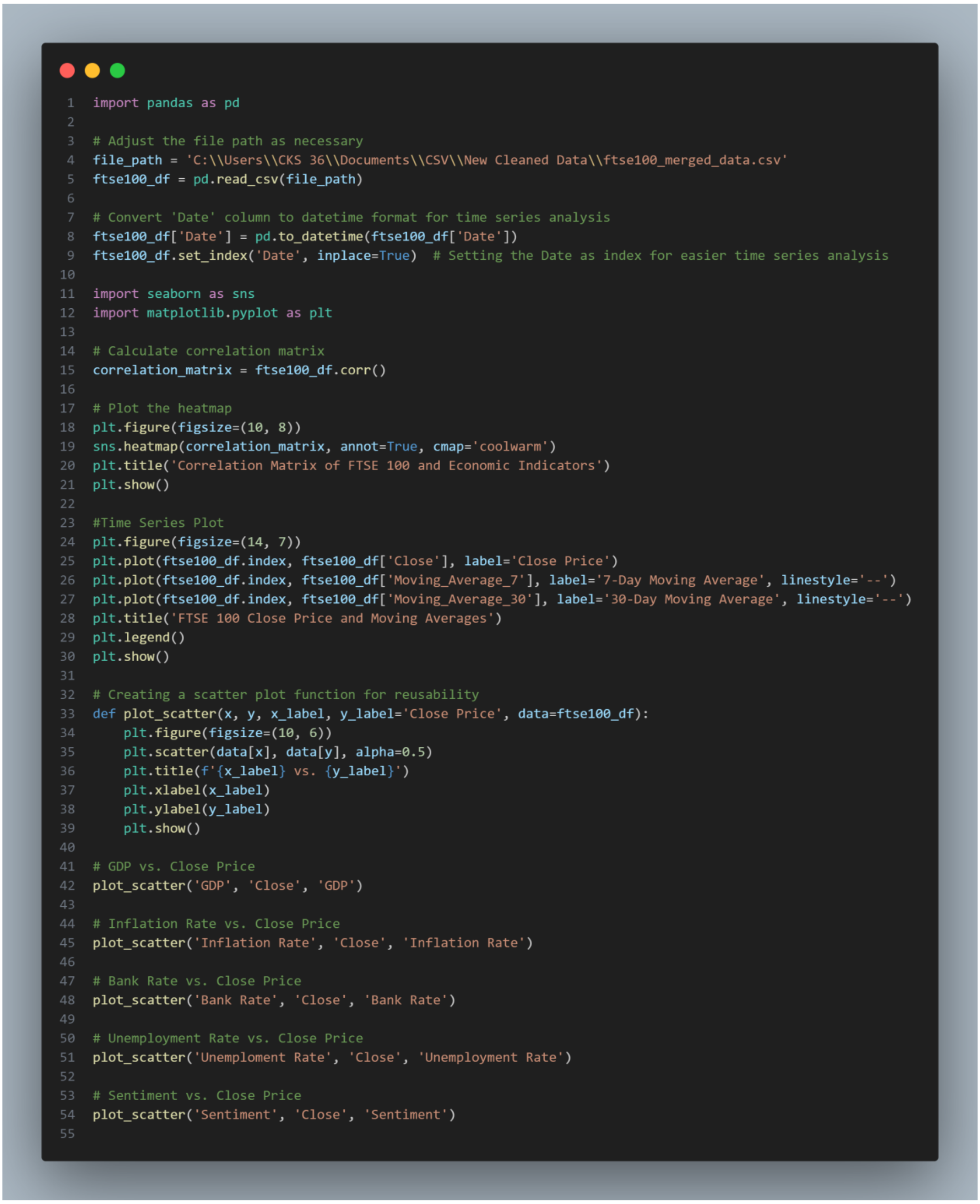
```python

# Function to normalize numerical data
def normalize_data(data_frames, columns_to_normalize):
    scaler = StandardScaler()
    for name, columns in columns_to_normalize.items():
        df = data_frames[name]
        numeric_cols = df[columns].select_dtypes(include=[np.number]).columns.tolist()
        if numeric_cols:
            df[numeric_cols] = scaler.fit_transform(df[numeric_cols])
        else:
            print(f"No numeric columns found for normalization in DataFrame '{name}'. Skipping.")
    return data_frames

# Function to save cleaned data
def save_cleaned_data(data_frames, save_paths):
    for name, path in save_paths.items():
        df = data_frames[name]
        df.to_csv(path, index=False)
    print("Data saved successfully.")

# Main function to orchestrate data preprocessing
def main():
    file_paths = {
        'ftse100_index_5y': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\ftse100_index_5y.csv'},
        'tweets': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\ftse100_tweets.csv'},
        'news': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\news_data.csv'},
        'gdp': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_gdp_growth.csv'},
        'inflation': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_inflation_rates.csv'},
        'interest': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_interest_rates.csv'},
        'unemployment': {'path': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_unemployment_rates.csv'}
    }

    data_frames = load_data(file_paths)
    data_frames = clean_data(data_frames)
    data_frames = add_moving_averages(data_frames)
    text_columns = {'tweets': 'Tweet', 'news': 'Headline'}
    data_frames = add_sentiment_score(data_frames, text_columns)

    columns_to_normalize = {
        'ftse100_index_5y': ['Adj Close', 'Moving_Average_7', 'Moving_Average_30']
    }
    data_frames = normalize_data(data_frames, columns_to_normalize)

    save_paths = {
        'ftse100_index_5y': 'C:\\Users\\CKS 36\\Documents\\CSV\\ftse100_index_5y_clean.csv',
        'tweets': 'C:\\Users\\CKS 36\\Documents\\CSV\\ftse100_tweets_clean.csv',
        'news': 'C:\\Users\\CKS 36\\Documents\\CSV\\news_data_clean.csv',
        'gdp': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_gdp_growth_clean.csv',
        'inflation': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_inflation_rates_clean.csv',
        'interest': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_interest_rates_clean.csv',
        'unemployment': 'C:\\Users\\CKS 36\\Documents\\CSV\\uk_unemployment_rates_clean.csv'
    }

    save_cleaned_data(data_frames, save_paths)

if __name__ == '__main__':
    main()
```

Appendix D – Integration Script



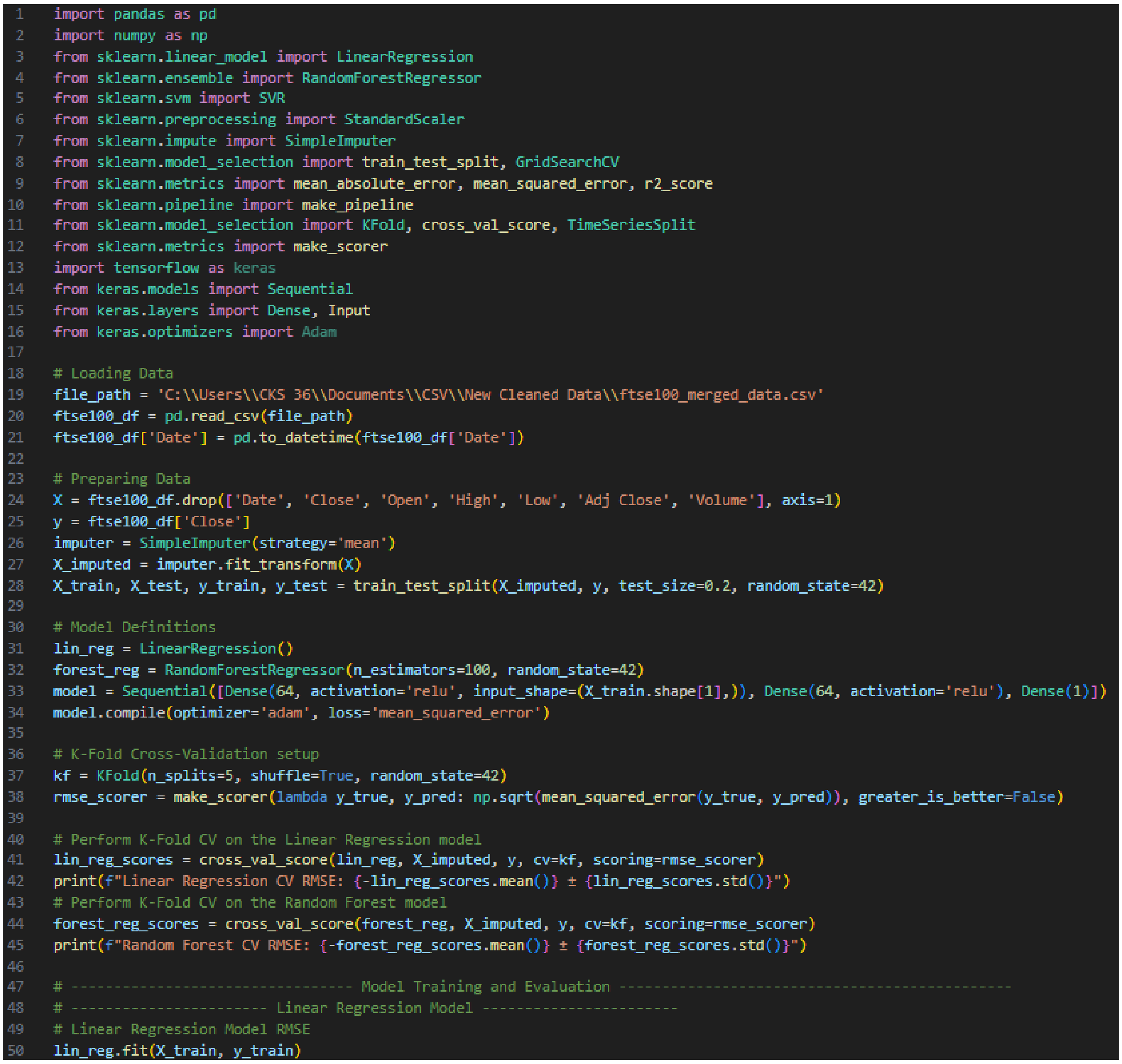
```python
### -- ECONOMIC INDICATOR INTEGRATION --
import pandas as pd

# Read
ftse100_df = pd.read_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\ftse100_index_5y_clean.csv')
gdp_growth_df = pd.read_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\uk_gdp_growth_clean.csv')
inflation_rates_df = pd.read_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\uk_inflation_rates_clean.csv')
interest_rates_df = pd.read_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\uk_interest_rates_clean.csv')
unemployment_rates_df = pd.read_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\uk_unemployment_rates_clean.csv')

# Correctly convert 'Date' columns to datetime in all DataFrames
ftse100_df['Date'] = pd.to_datetime(ftse100_df['Date'])
gdp_growth_df['Date'] = pd.to_datetime(gdp_growth_df['Date'])
inflation_rates_df['Date'] = pd.to_datetime(inflation_rates_df['Date'])
interest_rates_df['Date'] = pd.to_datetime(interest_rates_df['Date'])
unemployment_rates_df['Date'] = pd.to_datetime(unemployment_rates_df['Date'])

# Merging DataFrames on 'Date' column
ftse100_df = ftse100_df.merge(gdp_growth_df, on='Date', how='left')
ftse100_df = ftse100_df.merge(inflation_rates_df, on='Date', how='left')
ftse100_df = ftse100_df.merge(interest_rates_df, on='Date', how='left')
ftse100_df = ftse100_df.merge(unemployment_rates_df, on='Date', how='left')

# Backfilling NAN
ftse100_df[['GDP', 'Inflation Rate', 'Unemploment Rate', 'Moving_Average_7', 'Moving_Average_30']] = ftse100_df[['GDP', 'Inflation Rate', 'Unemploment Rate', 'Moving_Average_7', 'Moving_Average_30']].fillna(method='bfill')

# Check the first few rows to confirm successful merge
print(ftse100_df.head())

# Check for any NaN values that may need addressing
print(ftse100_df.isnull().sum())

### -- SENTIMENTAL INTEGRATION --
import pandas as pd

# Replace the path with the actual path to your news data CSV file
news_df = pd.read_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\news_data_clean.csv')

# Ensure the 'Date' column is in datetime format
news_df['Date'] = pd.to_datetime(news_df['Date'])

# Quick check of the data
print(news_df.head())

from nltk.sentiment.vader import SentimentIntensityAnalyzer
import nltk
nltk.download('vader_lexicon')

# Initialize the VADER sentiment intensity analyzer
sid = SentimentIntensityAnalyzer()

# Define a function to get the compound sentiment score
def get_sentiment_score(text):
    return sid.polarity_scores(text)['compound']

# Apply the function to your headlines or news content
news_df['Sentiment'] = news_df['Headline'].apply(get_sentiment_score)

# Aggregate sentiment scores by date if you have multiple articles per day
daily_sentiment = news_df.groupby('Date')['Sentiment'].mean().reset_index()

# Assuming your FTSE 100 DataFrame is named ftse100_df and it already has a 'Date' column in datetime format
ftse100_df = ftse100_df.merge(daily_sentiment, on='Date', how='left')

# Fill any missing sentiment scores with 0 or another method of your choice
ftse100_df['Sentiment'] = ftse100_df['Sentiment'].fillna(0)

# Check the merged DataFrame
print(ftse100_df.head())

# Save the merged DataFrame to a new CSV file
ftse100_df.to_csv('C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\ftse100_merged_data.csv', index=False)
print("Data Has been Merged and Succesfully Saved to a CSV")
```

Appendix E – Visualisation Script

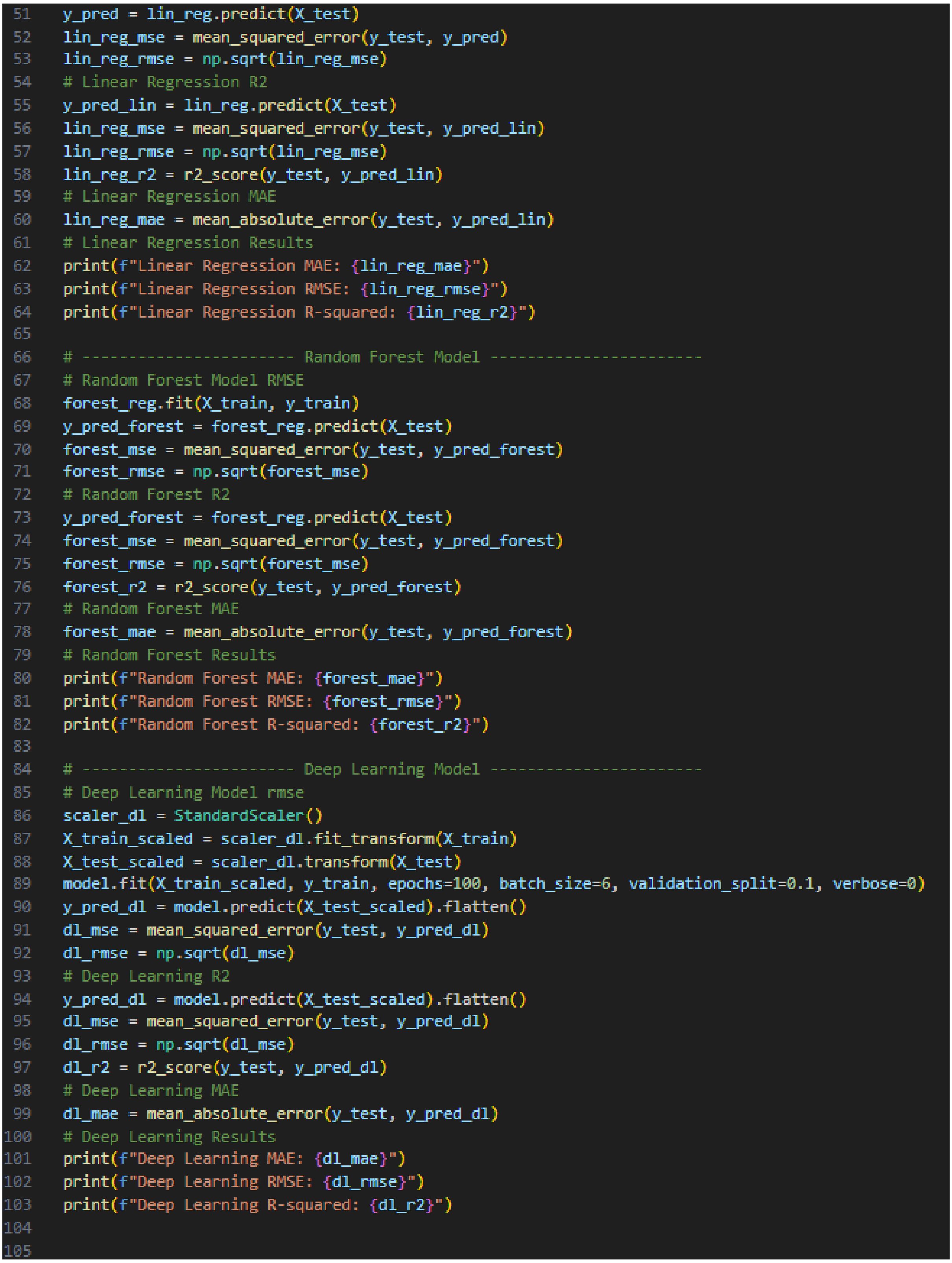


```python
import pandas as pd

# Adjust the file path as necessary
file_path = 'C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\ftse100_merged_data.csv'
ftse100_df = pd.read_csv(file_path)

# Convert 'Date' column to datetime format for time series analysis
ftse100_df['Date'] = pd.to_datetime(ftse100_df['Date'])
ftse100_df.set_index('Date', inplace=True)  # Setting the Date as index for easier time series analysis

import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix
correlation_matrix = ftse100_df.corr()

# Plot the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of FTSE 100 and Economic Indicators')
plt.show()

#Time Series Plot
plt.figure(figsize=(14, 7))
plt.plot(ftse100_df.index, ftse100_df['Close'], label='Close Price')
plt.plot(ftse100_df.index, ftse100_df['Moving_Average_7'], label='7-Day Moving Average', linestyle='--')
plt.plot(ftse100_df.index, ftse100_df['Moving_Average_30'], label='30-Day Moving Average', linestyle='--')
plt.title('FTSE 100 Close Price and Moving Averages')
plt.legend()
plt.show()

# Creating a scatter plot function for reusability
def plot_scatter(x, y, x_label, y_label='Close Price', data=ftse100_df):
    plt.figure(figsize=(10, 6))
    plt.scatter(data[x], data[y], alpha=0.5)
    plt.title(f'{x_label} vs. {y_label}')
    plt.xlabel(x_label)
    plt.ylabel(y_label)
    plt.show()

# GDP vs. Close Price
plot_scatter('GDP', 'Close', 'GDP')

# Inflation Rate vs. Close Price
plot_scatter('Inflation Rate', 'Close', 'Inflation Rate')

# Bank Rate vs. Close Price
plot_scatter('Bank Rate', 'Close', 'Bank Rate')

# Unemployment Rate vs. Close Price
plot_scatter('Unemploment Rate', 'Close', 'Unemployment Rate')

# Sentiment vs. Close Price
plot_scatter('Sentiment', 'Close', 'Sentiment')

```

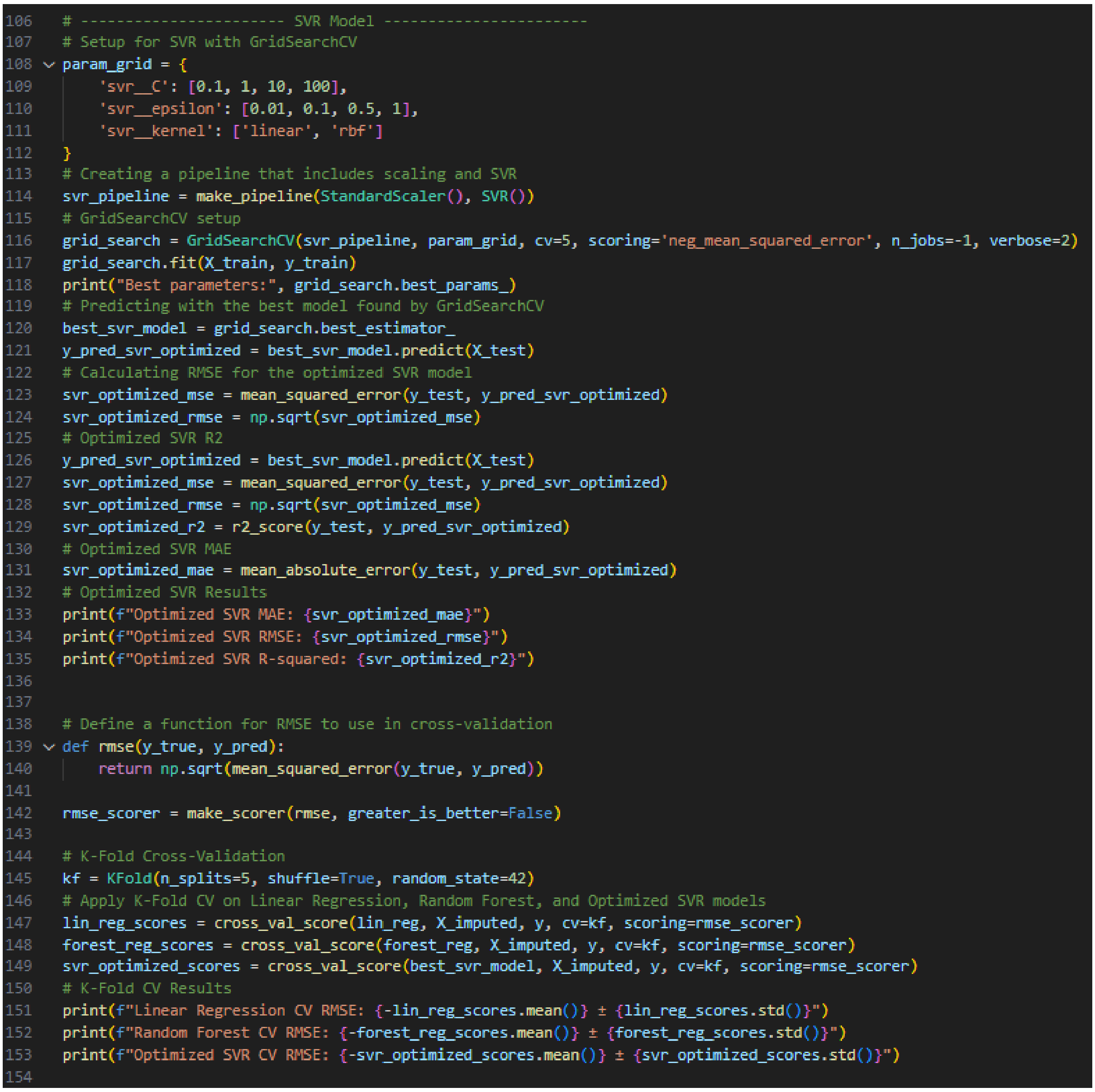
Appendix F – Model Implementation Script



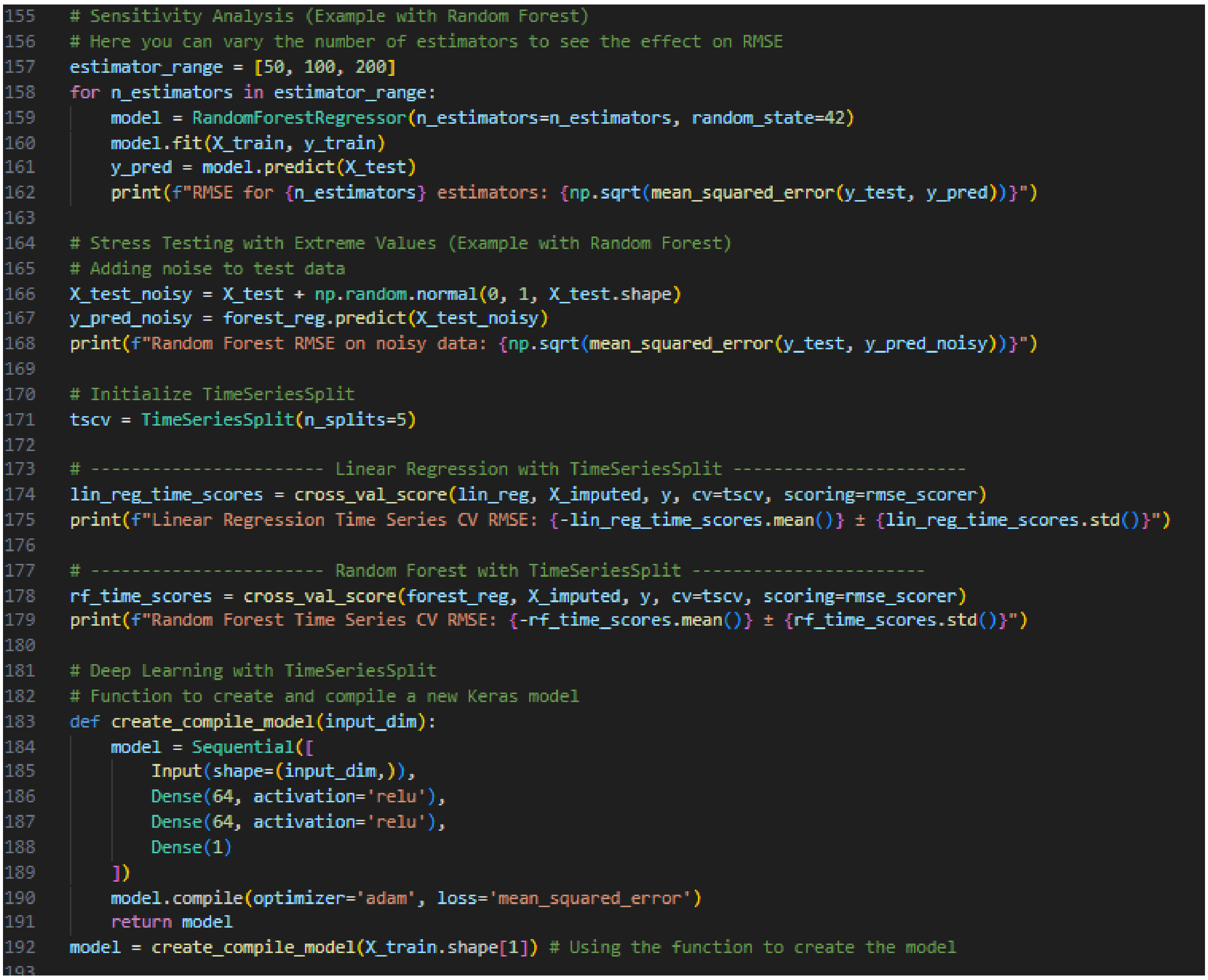
```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import KFold, cross_val_score, TimeSeriesSplit
from sklearn.metrics import make_scorer
import tensorflow as keras
from keras.models import Sequential
from keras.layers import Dense, Input
from keras.optimizers import Adam

# Loading Data
file_path = 'C:\\Users\\CKS 36\\Documents\\CSV\\New Cleaned Data\\ftse100_merged_data.csv'
ftse100_df = pd.read_csv(file_path)
ftse100_df['Date'] = pd.to_datetime(ftse100_df['Date'])

# Preparing Data
X = ftse100_df.drop(['Date', 'Close', 'Open', 'High', 'Low', 'Adj Close', 'Volume'], axis=1)
y = ftse100_df['Close']
imputer = SimpleImputer(strategy='mean')
X_imputed = imputer.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_imputed, y, test_size=0.2, random_state=42)

# Model Definitions
lin_reg = LinearRegression()
forest_reg = RandomForestRegressor(n_estimators=100, random_state=42)
model = Sequential([Dense(64, activation='relu', input_shape=(X_train.shape[1],)), Dense(64, activation='relu'), Dense(1)])
model.compile(optimizer='adam', loss='mean_squared_error')

# K-Fold Cross-Validation setup
kf = KFold(n_splits=5, shuffle=True, random_state=42)
rmse_scorer = make_scorer(lambda y_true, y_pred: np.sqrt(mean_squared_error(y_true, y_pred)), greater_is_better=False)

# Perform K-Fold CV on the Linear Regression model
lin_reg_scores = cross_val_score(lin_reg, X_imputed, y, cv=kf, scoring=rmse_scorer)
print(f"Linear Regression CV RMSE: {-lin_reg_scores.mean()} ± {lin_reg_scores.std()}")
# Perform K-Fold CV on the Random Forest model
forest_reg_scores = cross_val_score(forest_reg, X_imputed, y, cv=kf, scoring=rmse_scorer)
print(f"Random Forest CV RMSE: {-forest_reg_scores.mean()} ± {forest_reg_scores.std()}")

# --------------------------------- Model Training and Evaluation ----------------------------------------------
# ----------------------- Linear Regression Model -----------------------
# Linear Regression Model RMSE
lin_reg.fit(X_train, y_train)
```



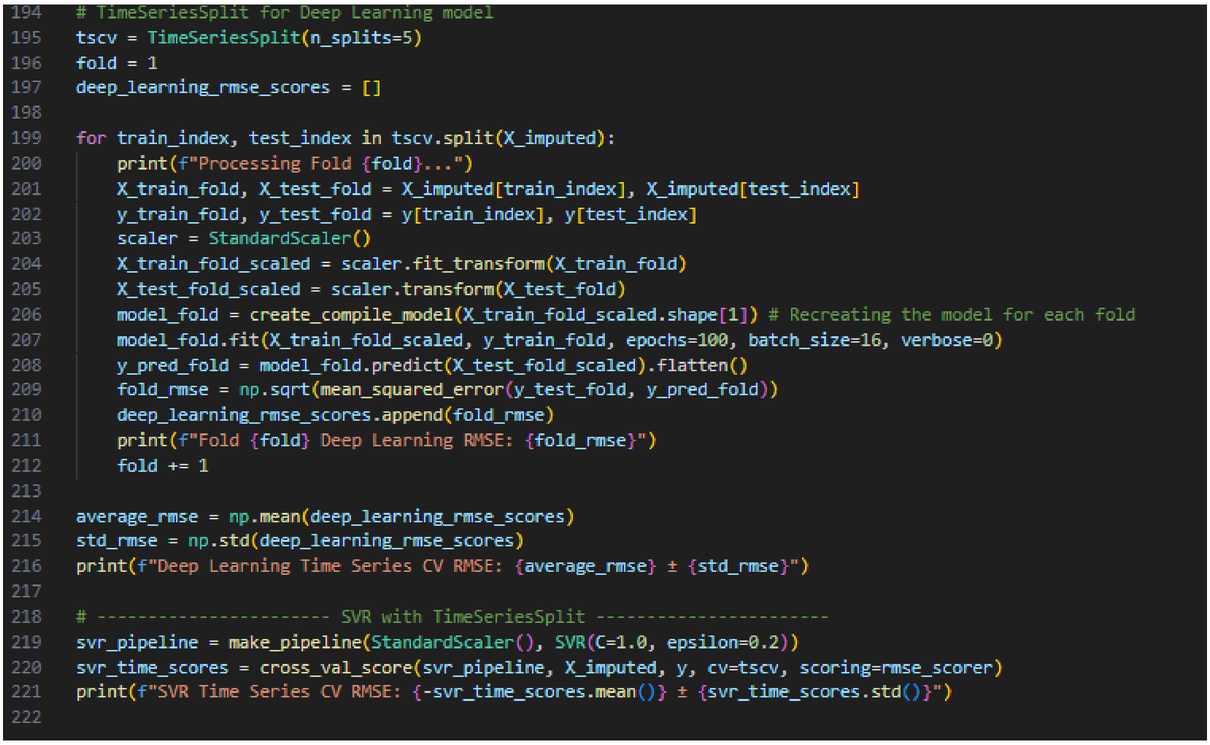
```
y_pred = lin_reg.predict(X_test)
lin_reg_mse = mean_squared_error(y_test, y_pred)
lin_reg_rmse = np.sqrt(lin_reg_mse)
# Linear Regression R2
y_pred_lin = lin_reg.predict(X_test)
lin_reg_mse = mean_squared_error(y_test, y_pred_lin)
lin_reg_rmse = np.sqrt(lin_reg_mse)
lin_reg_r2 = r2_score(y_test, y_pred_lin)
# Linear Regression MAE
lin_reg_mae = mean_absolute_error(y_test, y_pred_lin)
# Linear Regression Results
print(f"Linear Regression MAE: {lin_reg_mae}")
print(f"Linear Regression RMSE: {lin_reg_rmse}")
print(f"Linear Regression R-squared: {lin_reg_r2}")

# ----------------------- Random Forest Model -----------------------
# Random Forest Model RMSE
forest_reg.fit(X_train, y_train)
y_pred_forest = forest_reg.predict(X_test)
forest_mse = mean_squared_error(y_test, y_pred_forest)
forest_rmse = np.sqrt(forest_mse)
# Random Forest R2
y_pred_forest = forest_reg.predict(X_test)
forest_mse = mean_squared_error(y_test, y_pred_forest)
forest_rmse = np.sqrt(forest_mse)
forest_r2 = r2_score(y_test, y_pred_forest)
# Random Forest MAE
forest_mae = mean_absolute_error(y_test, y_pred_forest)
# Random Forest Results
print(f"Random Forest MAE: {forest_mae}")
print(f"Random Forest RMSE: {forest_rmse}")
print(f"Random Forest R-squared: {forest_r2}")

# ----------------------- Deep Learning Model -----------------------
# Deep Learning Model rmse
scaler_dl = StandardScaler()
X_train_scaled = scaler_dl.fit_transform(X_train)
X_test_scaled = scaler_dl.transform(X_test)
model.fit(X_train_scaled, y_train, epochs=100, batch_size=6, validation_split=0.1, verbose=0)
y_pred_dl = model.predict(X_test_scaled).flatten()
dl_mse = mean_squared_error(y_test, y_pred_dl)
dl_rmse = np.sqrt(dl_mse)
# Deep Learning R2
y_pred_dl = model.predict(X_test_scaled).flatten()
dl_mse = mean_squared_error(y_test, y_pred_dl)
dl_rmse = np.sqrt(dl_mse)
dl_r2 = r2_score(y_test, y_pred_dl)
# Deep Learning MAE
dl_mae = mean_absolute_error(y_test, y_pred_dl)
# Deep Learning Results
print(f"Deep Learning MAE: {dl_mae}")
print(f"Deep Learning RMSE: {dl_rmse}")
print(f"Deep Learning R-squared: {dl_r2}")


```



```python
# ---------------------- SVR Model ----------------------
# Setup for SVR with GridSearchCV
param_grid = {
    'svr__C': [0.1, 1, 10, 100],
    'svr__epsilon': [0.01, 0.1, 0.5, 1],
    'svr__kernel': ['linear', 'rbf']
}
# Creating a pipeline that includes scaling and SVR
svr_pipeline = make_pipeline(StandardScaler(), SVR())
# GridSearchCV setup
grid_search = GridSearchCV(svr_pipeline, param_grid, cv=5, scoring='neg_mean_squared_error', n_jobs=-1, verbose=2)
grid_search.fit(X_train, y_train)
print("Best parameters:", grid_search.best_params_)
# Predicting with the best model found by GridSearchCV
best_svr_model = grid_search.best_estimator_
y_pred_svr_optimized = best_svr_model.predict(X_test)
# Calculating RMSE for the optimized SVR model
svr_optimized_mse = mean_squared_error(y_test, y_pred_svr_optimized)
svr_optimized_rmse = np.sqrt(svr_optimized_mse)
# Optimized SVR R2
y_pred_svr_optimized = best_svr_model.predict(X_test)
svr_optimized_mse = mean_squared_error(y_test, y_pred_svr_optimized)
svr_optimized_rmse = np.sqrt(svr_optimized_mse)
svr_optimized_r2 = r2_score(y_test, y_pred_svr_optimized)
# Optimized SVR MAE
svr_optimized_mae = mean_absolute_error(y_test, y_pred_svr_optimized)
# Optimized SVR Results
print(f"Optimized SVR MAE: {svr_optimized_mae}")
print(f"Optimized SVR RMSE: {svr_optimized_rmse}")
print(f"Optimized SVR R-squared: {svr_optimized_r2}")


# Define a function for RMSE to use in cross-validation
def rmse(y_true, y_pred):
    return np.sqrt(mean_squared_error(y_true, y_pred))

rmse_scorer = make_scorer(rmse, greater_is_better=False)

# K-Fold Cross-Validation
kf = KFold(n_splits=5, shuffle=True, random_state=42)
# Apply K-Fold CV on Linear Regression, Random Forest, and Optimized SVR models
lin_reg_scores = cross_val_score(lin_reg, X_imputed, y, cv=kf, scoring=rmse_scorer)
forest_reg_scores = cross_val_score(forest_reg, X_imputed, y, cv=kf, scoring=rmse_scorer)
svr_optimized_scores = cross_val_score(best_svr_model, X_imputed, y, cv=kf, scoring=rmse_scorer)
# K-Fold CV Results
print(f"Linear Regression CV RMSE: {-lin_reg_scores.mean()} ± {lin_reg_scores.std()}")
print(f"Random Forest CV RMSE: {-forest_reg_scores.mean()} ± {forest_reg_scores.std()}")
print(f"Optimized SVR CV RMSE: {-svr_optimized_scores.mean()} ± {svr_optimized_scores.std()}")

```



```python
# Sensitivity Analysis (Example with Random Forest)
# Here you can vary the number of estimators to see the effect on RMSE
estimator_range = [50, 100, 200]
for n_estimators in estimator_range:
    model = RandomForestRegressor(n_estimators=n_estimators, random_state=42)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f"RMSE for {n_estimators} estimators: {np.sqrt(mean_squared_error(y_test, y_pred))}")

# Stress Testing with Extreme Values (Example with Random Forest)
# Adding noise to test data
X_test_noisy = X_test + np.random.normal(0, 1, X_test.shape)
y_pred_noisy = forest_reg.predict(X_test_noisy)
print(f"Random Forest RMSE on noisy data: {np.sqrt(mean_squared_error(y_test, y_pred_noisy))}")

# Initialize TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=5)

# ---------------------- Linear Regression with TimeSeriesSplit ----------------------
lin_reg_time_scores = cross_val_score(lin_reg, X_imputed, y, cv=tscv, scoring=rmse_scorer)
print(f"Linear Regression Time Series CV RMSE: {-lin_reg_time_scores.mean()} ± {lin_reg_time_scores.std()}")

# ---------------------- Random Forest with TimeSeriesSplit ----------------------
rf_time_scores = cross_val_score(forest_reg, X_imputed, y, cv=tscv, scoring=rmse_scorer)
print(f"Random Forest Time Series CV RMSE: {-rf_time_scores.mean()} ± {rf_time_scores.std()}")

# Deep Learning with TimeSeriesSplit
# Function to create and compile a new Keras model
def create_compile_model(input_dim):
    model = Sequential([
        Input(shape=(input_dim,)),
        Dense(64, activation='relu'),
        Dense(64, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model
model = create_compile_model(X_train.shape[1]) # Using the function to create the model
```



```python
# TimeSeriesSplit for Deep Learning model
tscv = TimeSeriesSplit(n_splits=5)
fold = 1
deep_learning_rmse_scores = []

for train_index, test_index in tscv.split(X_imputed):
    print(f"Processing Fold {fold}...")
    X_train_fold, X_test_fold = X_imputed[train_index], X_imputed[test_index]
    y_train_fold, y_test_fold = y[train_index], y[test_index]
    scaler = StandardScaler()
    X_train_fold_scaled = scaler.fit_transform(X_train_fold)
    X_test_fold_scaled = scaler.transform(X_test_fold)
    model_fold = create_compile_model(X_train_fold_scaled.shape[1]) # Recreating the model for each fold
    model_fold.fit(X_train_fold_scaled, y_train_fold, epochs=100, batch_size=16, verbose=0)
    y_pred_fold = model_fold.predict(X_test_fold_scaled).flatten()
    fold_rmse = np.sqrt(mean_squared_error(y_test_fold, y_pred_fold))
    deep_learning_rmse_scores.append(fold_rmse)
    print(f"Fold {fold} Deep Learning RMSE: {fold_rmse}")
    fold += 1

average_rmse = np.mean(deep_learning_rmse_scores)
std_rmse = np.std(deep_learning_rmse_scores)
print(f"Deep Learning Time Series CV RMSE: {average_rmse} ± {std_rmse}")

# ---------------------- SVR with TimeSeriesSplit ----------------------
svr_pipeline = make_pipeline(StandardScaler(), SVR(C=1.0, epsilon=0.2))
svr_time_scores = cross_val_score(svr_pipeline, X_imputed, y, cv=tscv, scoring=rmse_scorer)
print(f"SVR Time Series CV RMSE: {-svr_time_scores.mean()} ± {svr_time_scores.std()}")
```

References

Bank of England. (2024). Interest rates and Bank Rate. Bank of England. https://www.bankofengland.co.uk/monetary-policy/the-interest-rate-bank-rate

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1-8.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time Series Analysis: Forecasting and Control. John Wiley & Sons.

Braun, M. (2021). Advanced Financial Analysis with Python. Finance Press.

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chollet, F., et al. (2015). Keras. https://keras.io

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Creswell, J. W., & Creswell, J. D. (2017). Research Design: Qualitative, Quantitative, and Mixed Methods Approaches. Sage publications.

Dietterich, T. G. (2000). Ensemble methods in machine learning. International Workshop on Multiple Classifier Systems, 1-15.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The Journal of Finance, 25(2), 383–417.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119–139.

Graham, B., & Dodd, D. L. (1934). Security Analysis. McGraw-Hill.

Gooding, P. (2023, September 20). Consumer price inflation, UK - Office for National Statistics. https://www.ons.gov.uk/economy/inflationandpriceindices/bulletins/consumerpriceinflation/august2023

Hair Jr, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2010). Multivariate Data Analysis (7th ed.). Prentice Hall.

Hawkins, D. M. (2004). The problem of overfitting. Journal of Chemical Information and Computer Sciences, 44(1), 1–12.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer.

Kahneman, D., & Tversky, A. (1979). Prospect Theory: An Analysis of Decision under Risk. Econometrica, 47(2), 263–291.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. International Joint Conference on Artificial Intelligence, 14(2), 1137-1145.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

London Stock Exchange. (2024). Table - FTSE 100 FTSE constituents. https://www.londonstockexchange.com/indices/ftse-100/constituents/table

McKinney, W. (2012). Python for Data Analysis. O’Reilly Media, Inc.

Mitchell, R. (2018). Web Scraping with Python: Collecting More Data from the Modern Web. O’Reilly Media.

ONS. (n.d.). Gross Domestic Product: quarter on quarter growth rate: CP SA % - Office for National Statistics. https://www.ons.gov.uk/economy/grossdomesticproductgdp/timeseries/ihyn/ukea

Partington, R., & Wearden, G. (2019, January 20). FTSE 100 tumbles by 12.5% in 2018 – its biggest fall in a decade. The Guardian. https://www.theguardian.com/business/2018/dec/31/ftse-100-tumbles-by-125-in-2018-its-biggest-fall-in-a-decade

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. The Journal of Finance, 19(3), 425–442.

Stock, J. H., & Watson, M. W. (2015). Introduction to Econometrics (3rd ed.). Pearson.

Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. The Journal of Finance, 62(3), 1139-1168.

Tsay, R. S. (2005). Analysis of Financial Time Series. Wiley-Interscience.

Wearden, G. (2019, December 31). FTSE 100 posts 12% gain for 2019 after strong year for market – as it happened. The Guardian. https://www.theguardian.com/business/live/2019/dec/31/global-markets-rally-shares-ftse-100-pound-oil-markets-business-live

Wearden, G. (2021, December 31). FTSE 100 bounces back despite Covid to finish 14.3% up in 2021. The Guardian. https://www.theguardian.com/business/2021/dec/31/ftse-100-bounces-back-despite-covid-to-finish-143-up-in-2021

Wearden, G. (2022, December 30). FTSE 100 ends 2022 slightly up despite global turmoil. The Guardian. https://www.theguardian.com/business/2022/dec/30/ftse-100-2022-up-share-index-pound-dollar

Zhang, G. P., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 14(1), 35–62.