Predicting S&P 500 Daily Prices: Integrating Macroeconomic Factors with Technical and Sentiment Indicators Using Machine Learning Models

Michail Patsiarikas

SID: 3308220020

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2025

THESSALONIKI – GREECE

Predicting S&P 500 Daily Prices: Integrating Macroeconomic Factors with Technical and Sentiment Indicators Using Machine Learning Models

Michail Patsiarikas

SID: 3308220020

Supervisor:	Prof. Christos Tjortjis
Supervising Committee Members:	Dr. Paraskevas Koukaras
	Dr. Christos Berberidis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

JANUARY 2025

THESSALONIKI – GREECE

Abstract

This study examines the efficiency of Machine Learning (ML) models in stock market prediction, when macroeconomic factors are combined with technical and sentiment indicators. Focusing on the prediction of the S&P 500 index, a careful modelling architecture is proposed. It starts with sentiment scoring in contextual data with TextBlob and pre-trained DistilBERT-base-uncased from Hugging libraries, dataset preprocessing, feature engineering and selection techniques, such as lagging and Recursive Feature Elimination (RFE), respectively. Next, a selection of traditional models, such as Linear Regression (LR), Random Forest (RF) and Gradient Boosting (GB), with more advanced ones, such as XGBoost and Multi-Layer Perceptron (MLP) is examined. LR and MLP provide high R^2 scores at 0.998 and low error MSE and MAE rates to average 350 and 13 points respectively, across both training and test datasets, slightly improved in terms of error prediction when Recursive Feature Elimination (RFE) is applied, resulting in superior accuracy that enhances predictive capabilities and valuable implications for investors to optimize their decision-making and risk strategies. In all, results also highlight potential limitations for researchers to further explore and adapt in financial modelling techniques.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Christos Tjortjis, for his invaluable support, patience, inspiration and guidance throughout this work. Furthermore, I would like to thank PhD candidate, Georgios Papageorgiou, and Postdoctoral Researcher Dr. Vangelis Sarlis for their valuable feedback and suggestions during this period.

Michail Patsiarikas

17/01/2024

Contents

List of Figures

List of Tables

1. Introduction

Financial forecasting of stock markets has always been an objective of great interest in modern economies, where precision and timeliness can yield significant economic and strategic advantages. Accurate predictions of stock market movements have a great effect on investors, policymakers, and financial institutions' decisions, which are important for resource allocation, risk management, and return maximization. In this remarkably demanding and dynamic environment of financial markets, changes in stock prices depend heavily on traditional financial indicators, but also on macroeconomic conditions and market sentiment that constantly changes. Technological improvements have also increased access to data, giving the opportunity to exploit the computing power of advanced ML techniques, which capture complex patterns in financial time series, beyond traditional forecasting models.

Financial forecasting used to mainly rely on either fundamental analysis or technical indicators to forecast stock prices, until great progress in advanced forecasting technologies was made. Fundamental analysis is a technique that focuses on analyzing the financial health of the company and market conditions, while Technical Analysis (TA) examines past price and volume information to identify trends and patterns. While technology improvements have introduced the integration of more sophisticated models, including those based on ML algorithms, which exploit temporal dependencies in stock data to improve performance, most of the previous studies have focused on technical indicators and sentiment analysis to predict stock prices [25],[34],[35],[36],[58], [59],[60],[61]. These approaches often overlook the contribution that macroeconomic indicators might make to a wider view of economic sentiment and the impact of exogenous shocks.

This study addresses the research gap by incorporating macroeconomic indicators with technical and sentiment-based features in the task of stock price predictions. Such a combination serves to enrich the feature space with varied inputs, further enabling the understanding of various market dynamics, which improves accuracy.

This study mainly aims to predict the closing price of the US stock market using an enriched feature dataset and to find the most effective ML model on stock market

prediction. Specifically, it tries to integrate macroeconomic indicators, technical indicators, and sentiment scores into one framework for unified forecasting. Although sentiment analysis and technical indicators have been widely used in financial forecasting, macroeconomic indicators reflecting the broader economic sentiment have been relatively unexplored in this domain [33],[42],[43].

To accomplish the above-mentioned aim, the following objectives are defined for the study:

1. To create a sophisticated dataset leveraging macroeconomic, technical and sentiment scoring indicators that will be utilized in the prediction of the adjusted daily closing price of the S&P 500 index. While the explainability of technical and sentiment indicators has been already studied in the research, macroeconomic indicators signal different aspects of the economic state that can enhance the predictability of the stock index when they are combined with the aforementioned indicators.

2. To evaluate the efficiency of traditional and more advanced ML models, such as Linear Regression (LR), Random Forest (RF), Gradient Boosting (GB), XGBoost Regressor, and Multi-Layer Perceptron (MLP) in predicting stock prices. Each of these models leverage strengths of linear, tree-based, boosting, and neural network methods that are able to effectively capture the different aspects of the features and ensure a comprehensive stock index prediction.

3. To identify the optimal combination of features by applying feature selection techniques.

4. To assess the contribution of each feature in the price movement by examining the performance through feature importance approach.

These objectives aim to enhance the existing literature by investigating the integrated impact of macroeconomic conditions, technical indicators, and market sentiment on stock price predictions and also examines how ML models would perform under different feature combinations, with and without prior feature selection.

In order to accomplish the above objectives, this study utilized various techniques for sentiment scoring, data pre-processing and engineering, as well as feature selection, which generate a comprehensive dataset able to be utilized by the models. ML

models are trained, and their performance is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE) and R-squared (R^2).

Initial results indicate exceptional model performance, with R^2 scores consistently exceeding 0.99 in both training and test datasets across multiple scenarios. However, challenges related to overfitting emerged, particularly with RF, GB and XGBoost, highlighting the necessity for additional tuning. In addition, feature importance analysis reveals the dominance of technical indicators, particularly Exponential Moving Average (EMA), among other features in prediction, with less contribution evident from macroeconomic indicators, Business Confidence Index (BCI) and Consumer Sentiment Index (CEI). Sentiment scores were found to provide negligible contribution to the overall prediction since no valuable supplementary insights emerged.

Overall, the study's novelty is highlighting that a broader set of features enhances the predictability of ML models. However, limitations and weaknesses are also evident based on the architecture developed, in terms of engineering and modelling, indicating the potential for improvement and further contribution.

2. Literature Review

In this section the academic literature is outlined spanning from the early approaches to predicting the stock market up to the most advanced techniques implemented so far. In particular, the core theories and the TA fundamental approach are discussed, followed by the impact of sentiment analysis and the implementation from traditional to more advanced ML models.

2.1. Traditional Views on Stock Market Prediction

Prior to any of the revolutionary studies that set the ground for the prevailing approaches for stock market prediction, the main methods applied were strongly assumed to provide some sort of accuracy regarding the stock market future price movements. These methods mainly relied on TA, fundamental analysis, and market timing strategies and were considered adequate to provide efficient estimations for future price movements. While these investment strategies for stock market prediction were broadly used and accepted by professionals and investors, they were challenged later by the fundamental works on the Efficient Market Hypothesis (EMH) and the random walk theory by Fama, shifting financial approach to another direction.

2.1.1. Core Theories

One of the earliest studies on stock market prediction was carried out by Fama (1965) who built on previous work from Bachelier (1900), introducing the idea of random walk in stock prices. In his work he suggested that stock prices follow a random walk, meaning that any changes that occur in price movements do not depend on past prices and therefore they cannot affect future prices. In essence, stock prices incorporate any new available information, and this is reflected in any given time leaving no space for consistent yields. By demonstrating the findings of his study, he set the ground for EMH.

In Fama (2017), an extensive overview of the EMH was developed. The hypothesis stated that financial markets are efficient in reflecting all the available information in asset prices so that modern passive investment strategies are possible to be applied. In particular, the hypothesis stated that market efficiency is separated into three categories depending on the information that is incorporated into prices. The first

category is a weak form of efficiency as the information depends on historical prices and therefore it is not possible to accurately predict future prices. The second category is the semi-strong form where the efficiency of the financial markets is based on public information. This was found to highly affect the price movements, which were rapidly adjusted to new public information. The last category, which is considered the strongest, suggests that the market incorporates all the available information, leaving investors unable to outperform the market continuously. In overall, the study provided an initial insight into the predictive ability of the stock market based on these categories, summarizing the inability for investors to obtain abnormal returns on active trading.

2.1.2. Technical Analysis

The idea to predict stock prices by applying techniques and methods suggested by TA strategy is dated centuries ago and is based on the foundation that the patterns found on historical prices and trading volumes charts can carry valuable information for future price movements. These patterns provide a guidance on future price movements' trend since they carry all the investors psychology at the time of an event. Their consistent repetition on prices charts also provides a confirmation for investors and professionals on the proactive decisions that they will need to make. As suggested by Dow (1900), the stock market movement is based on primary, secondary and minor trends and each one of these trends provide meaningful signals for the future price movements. His Dow Theory is a milestone that established the ground for TA and is actively used in several different forms also today.

Some of the most common techniques used as part of the TA include the moving average, the support and resistance levels and chart patterns. In moving average, price movements are smoothed out by taking constantly the daily average price and in that way the price movement direction and the trend is identified. The technique of support and resistance levels is mainly focused on detecting the stop point of an ascending or descending behaviour, which act as floor or ceiling, in the sense that from that point there is price movement reverse. Finally, chart patterns, like head and shoulders and double tops/bottoms are used to detect signals of future price reversals or continuations.

In another work by Magee and Edwards (1948), the authors suggested that investors are confident to predict future price movements when patterns provided by TA methods are understood. This confidence is based on the foundation that all relevant

information regarding a stock is fully captured in the price and volume movements. Therefore, analyzing historical patterns, investors shape anticipation for the next movements.

In opposition to these beliefs, Kendall (1953) challenged the capability of price patterns to predict future movements. In his work he found that the behaviour of stock price movement is random and therefore they do not depend on patterns. Following his work, he set the ground for the effect of market efficiency in stock market price changes, challenging at the same time TA as the dominant technique for prediction.

2.2. Sentiment Analysis in Stock Market Forecasting

On the basis of examining information that steam from social media in order to predict stock price changes, L.I. Bing et al. (2014) collected public tweets and historical closing stock prices data and employed Natural Language Processing (NLP) techniques along with data mining techniques. Their working paper explored and confirmed the possibility of internal association in the multilayer hierarchical structures. In particular, they proved relationship patterns between public sentiment and stock prices with accuracy up to 76.12%.

In Cakra (2015) study, five classification algorithms were employed to their model in order to predict the Indonesian stock market and the effect Twitter sentiment had on it. The main objectives of their model included margin percentage, stock price and price volatility prediction. For the prediction of tweets, support vector machine, naïve bayes, decision tree, RF and neural network were employed. With an accuracy of 60.39% and 56.5% respectively RF and naïve bayes classifiers dominated among the other algorithms, while LR achieved a 67,73% accuracy on prices prediction. The main weakness of their study though was the limited time frame of the five last dates used for data retrieval based on which their model was developed.

Hana (2015) explored the statistically significant stock price change that could be obtained from information in news articles with breaking tweets volume. Stock market news with breaking tweets and one hour stock prices charts were collected in order to predict any potential trend in stock prices. The outcome of the study demonstrated that applying logistic regression with 1-gram keyword to the data had a good effect on predicting price movements. Nevertheless, this was not the case when

extracted document level sentiment features were used since they did not have significant effect on hourly price change predictions, but this was more related to the breaking news data used.

The paper by Shweta et al. (2020) describes a study on the prediction of stock prices using regressor algorithm and twitter sentiment analysis. The authors collected data for stock prices from NSE websites of different companies such as Yahoo Finance, while consumer sentiment data were collected from Twitter using Python library and Tweepy. In their study, they mainly focused on the application of RF Algorithm to analyze the data and predict future stock prices as they suggest that ensemble methods provide higher reliability than the accuracy of individual models. The study found that the algorithm applied provided an 85% in predicting stock prices and strong correlation between all classified sentiments (positive, negative, and neutral), unlike other studies that presented correlation only for neutral behavior. In overall, the paper presented an interesting application of data mining techniques to predict stock prices. However, it is important to note that stock market predictions are inherently uncertain and subject to various factors beyond the scope of the study.

Gupta et al. (2019) examined the prediction of stock prices using two different approaches. In their first approach they proposed a model that consists of three main algorithms used to predict stock prices while the second model is based on the sentiment analysis of twitter feeds. The authors initially used a dataset of historical stock prices for six banking stocks and applied the three different prediction algorithms KNN, Genetic Algorithm and SVR separately. Experimental results showed that KNN algorithm achieved an accuracy of 65-70% on the test data when data are not largely skewed, otherwise the accuracy was below 50%. Better accuracy was achieved by Generic Algorithm, but this was enhanced by SVR which reached an accuracy of 89%. Finally, the second model was applied to the dataset considering twitter sentiment analysis. As suggested by the authors, stock market is highly influenced by market news and feeds and as a result attaching a sentiment analysis to the dataset will potentially increase the choice of a less riskier investment decision. The final results presented proved that accuracy of the model increases to 70-75% when sentiment analysis is also applied to the data.

Various studies have analyzed and proved the correlation between the movements in stock market prices and the news being relevant to it. A vast amount of text data available from various sources has been excessively used in research in order to predict stock market volatility. A relevant work to this is from Kim et al. (2014) examined this relationship by using unstructured data from mobile channels and social network services.

In another study on stock price prediction based on news analysis, (Selimi et al., 2017) applied text-mining techniques that were related to text sentiment analysis and formed two models in order to capture the effect of financial news on stock price changes. Initially, the sentiment in the news was analyzed on historical media news data by applying the Naïve Bayesian classifier for document classification. Then a 5-day average rate of change of stocks' market price was also calculated and added in one of the two models in order to compare their accuracy on training results. In their results, they found that the model consisting only of textual content variables provided limited knowledge on future stock price movements. Any addition of variables that reflect the stock price volatility such as the 5-day average rate of stocks' price produced higher accuracy models. In their study when such a feature was added, the accuracy of the model increased from 49,49% to 94,29%.

In another approach to prove the relationship between financial news and stock price movements, Khedr et al. (2017) performed a sentiment analysis and examined the effect of news on historical stock prices in order to predict their behavior in the future. Their methodology was based on Naïve Bayes algorithm which was used in order to categorize news polarities and perform sentiment analysis. Their results proved that their proposed model provided high accuracy up to 86,21% in their experiment in prediction when only text data were employed while accuracy increased to 89,80% after the addition of numerical features.

In an attempt to study the hypothesis that the stock price prediction resulting from the text mining on financial news can be further improved, Hagenau et al. (2013) employed expressive attributes, such as market feedback, in their text mining methods among other variables. According to their results, such an addition had drastically improved the accuracy of their model since the majority of noisy data was removed and hence the classifier avoided overfitting during classification of the text. Further

employing a feedback-based feature selection with a combination of 2-words resulted in an even better accuracy of 76% that also added significance and more information to the sentiment classification.

Joshi et al. (2016) also examined the field of prediction of stocks by taking the financial news feed related to a single company and carried out a sentiment analysis in order to predict stock price movements. Similarly to previously mentioned studies, they also approached stock price volatility with respect to news polarity expanding though the examined period of stock prices and news articles to three years. In their attempt to examine Apple Inc. they built the training set labelling these articles in a dictionary-based approach to positive and negative financial specific words. Then, the pre-processing of the data was carried out so that the financial dictionary and related stop words to be determined. On top of that they used three models for classification and testing. The outcome of their study proved that the highest accuracy for the test cases was achieved when the RF algorithm was employed, ranging from 88% to 92%. A significantly high accuracy rate of 86% was also achieved when the Support Vector Machines (SVM) algorithm was trained. In contrast, Naïve Bayes algorithm had the lowest performance compared to the other algorithms with an accuracy of 83%, which was still relatively high.

Likewise, Kaya et al. (2010) implemented a study where they combined news articles and stock market prices for the same period that referred to previous year data. As per previous studies they labeled articles as positive and negative sentimental categories considering their effects on stock prices. The difference in their approach was that the categorization of articles data was based on the analysis of text data at the level of word pairs of a noun and a verb and not as a single word. Having formed this categorization they employed the SVM method and attained an accuracy of 61% which can be characterized relatively low compared to aforementioned studies.

Patric et al. (2014) performed sentiment analysis in financial markets by analyzing the news in stock market based on the German language. They applied several text mining techniques and integrated word association and lexical resources using sentiWS tool for sentiment analysis. The aim of their study was to find a relationship between future stock prices and the sentiment measures they modeled so that they could support investors with lower risk investment recommendations.

In another attempt to investigate the relationship between future stock prices and news, Shynkevich et al. (2015) went a bit further by analyzing two categories of articles, such as sub-industry and actual stock related. The aim of their study was to prove that the category on news could increase prediction of stock trend accuracy affected though by the news themselves and the historical stock price data. In their estimations they applied several methods to find accuracy on their prediction and used open and close attributes for historical stock prices. Results provided increased accuracy when polynomial kernels were applied in their model estimation on news categories reaching 79.59%. Worse prediction accuracy was achieved with all the other methods used such as support vector machine and k-NN.

Umbarkar and Nandgaonkar (2016) aimed to predict stock market events using association rule mining on financial news. The study collected financial news articles from the Wall Street Journal for the period from 2007 to 2012. The articles were preprocessed using NLP techniques such as tokenization, stemming, and stop word removal to extract relevant information. The study then applied association rule mining techniques by using six important trading technical indicators to generate rules and based on them they analyze the news articles and predicted stock market events such as bullish or bearish trends.

Their study proposed a new association rule mining algorithm called the improved apriori algorithm (IAA) that was designed to handle the large amount of data in financial news articles. The algorithm was compared with other existing algorithms such as the apriori algorithm and the frequent pattern growth (FPG) algorithm. The results of the study showed that the IAA algorithm outperformed the other existing algorithms in terms of accuracy and efficiency. The study found that the IAA algorithm was able to extract useful association rules from financial news articles and predict stock market events with high accuracy.

Overall, authors contributed to the literature on stock market prediction by proposing a new association rule mining algorithm that was shown to be effective in predicting stock market events using financial news articles. The study demonstrated the importance of using association rule mining techniques for analyzing large volumes of textual data and predicting complex stock market events.

Price et al (2018) conducted research to predict stock market prices using text mining and rule-based techniques on financial news articles. The study collected financial news articles from the Wall Street Journal for the period from 2010 to 2015. The articles were preprocessed using NLP techniques such as tokenization, stemming, and stop word removal to extract relevant information. The study then used a rule-based technique called the Mamdani fuzzy inference system (FIS) to analyze the news articles and predict the stock market prices.

The Mamdani FIS is a rule-based technique that uses a set of fuzzy rules to transform inputs into outputs. In the context of the study, the Mamdani FIS was used to predict stock market prices based on the sentiment of the news articles. The study compared the performance of the Mamdani FIS model with other existing models such as the support vector machine (SVM) model and the artificial neural network (ANN) model. The performance of the models was evaluated using various statistical measures such as accuracy, precision, recall, and F1-score. The results of the study showed that the Mamdani FIS model outperformed the other existing models in terms of predicting stock market prices using financial news articles. The study found that the Mamdani FIS model was able to extract useful information from the news articles and capture the complex relationships between the news and the stock market prices.

Overall, authors contributed to the literature on stock market prediction by demonstrating the effectiveness of the Mamdani FIS model on financial news articles. The study showed that the Mamdani FIS model was a promising tool for predicting stock market prices using textual data and could be used in real-world applications such as financial investment and trading.

Desai and Gandhi (2014) conducted a research study that aimed to predict stock market prices using data mining techniques on historical stock market data. The study collected historical stock market data from the National Stock Exchange of India (NSE) for the period from 2000 to 2015. The data included daily closing prices, trading volumes, and other financial indicators for a selected set of stocks. The study used various data mining techniques such as regression analysis, decision trees, and Neural Networks (NN) to analyze the historical data and predict future stock market prices. The performance of the models was evaluated using various statistical measures such as MAE and MSE. The results of the study showed that the neural network model

outperformed the other data mining techniques in terms of predicting stock market prices. The study found that the neural network model was able to capture the nonlinear relationships between the stock market variables and predict future stock market prices with high accuracy.

In Seker et al. (2014) study, authors aimed to investigate the relationship between stock market prices and economic news using time series analysis and text mining techniques. The study collected daily stock market data from the Istanbul Stock Exchange (ISE) for a period of six years, from 2007 to 2013. The data included the daily closing prices of the ISE 100 Index, trading volumes, and other financial indicators. The study also collected news articles related to the Turkish economy from major Turkish news sources. The study used various time series analysis techniques such as autoregressive integrated moving average (ARIMA) models and generalized autoregressive conditional heteroscedasticity (GARCH) models to analyze the historical stock market data and identify patterns and trends. The study also used text mining techniques to extract sentiment and other valuable information from the news articles such as SVM and KNN classifiers. The results of the study showed that there was a significant correlation between stock market prices and economic news. The study found that positive news about the Turkish economy had a positive effect on the stock market, while negative news had a negative effect. The study also showed that the sentiment of the news articles had a significant impact on the stock market prices.

Kim et al. (2016) conducted a study in order to explore the usefulness of text mining techniques for predicting stock market movements based on news articles. The study collected news articles related to the Dow Jones Industrial Average (DJIA) from various sources such as The New York Times and The Washington Post for a period of 10 years, from 2003 to 2013. The study used a text mining technique called sentiment analysis to extract the sentiment of the news articles. Their model was built by using NLP and used various ML algorithms such as decision tree, RF, and Support Vector Machine (SVM) to predict the direction of the DJIA index movement based on the sentiment of the news articles. The study also compared the performance of these algorithms with the traditional buy-and-hold strategy. The results of the study showed that the sentiment of the news articles had a significant impact on the stock market movements. The study found that the SVM algorithm performed the best in predicting the direction of the DJIA index movement based on the sentiment of the news articles.

The study also showed that the ML algorithms outperformed the traditional buy-and-hold strategy.

Abdullah et al. (2018) conducted a study that aimed to explore the use of text mining and NLP techniques in analyzing Bangladesh stock market based on news articles. The study collected news articles related to the stock market from various sources such as Yahoo Finance and Google Finance for a period of 10 years, from 2007 to 2017 and different fundamental factors related to companies that include, EPS, P/E ratio, beta, correlation, and standard deviation along with price trend. The study used text mining techniques such as topic modeling, sentiment analysis, and keyword extraction to analyze the news articles. In particular the information parser algorithm and Apache OpenNLP ML toolkit were used. The study then used statistical analysis techniques such as regression analysis and correlation analysis to examine the relationship between the news articles and the stock market movements. The study also compared the performance of the text mining techniques with the traditional TA approach. The results of the study showed that the sentiment of the news articles had a significant impact on the stock market movements. The study also showed that the text mining techniques outperformed the traditional TA approach in predicting the stock market movements.

In an attempt to explore the influence of media sentiment on Microsoft, Tesla, and Apple stock prices for the period 2022 to 2023, Cristescu et al. (2023) employ various regression models. In their study, they employed TextBlob library in order to analyze news headlines and descriptions which categorize them into positive, neutral, or negative based on sentiment polarity. Several statistical tools were employed such as Pearson correlation, wavelet coherence, and various regression models, which revealed significant correlations between the stocks' title polarity and their corresponding closing prices. The wavelet coherence provided also temporal patterns where strong relationships between data were found for specific periods. Next, they compared the effectiveness of non-linear and linear models and found the ability of non-linear models, such as cubic regressions, to encompass more predictive power for capturing stock price fluctuations. Finally, the authors recognized the limitations of TextBlob and suggested more advanced sentiment allocation techniques like BERT and prediction methods like BEKK modeling for future analysis since they are more capable to overcome covolatility and persistence patterns between sentiment and stock prices.

Another study that explores the relationship between news sentiment and stock prediction using ML techniques is performed by Costola et al. (2023). They aimed to capture the impact of COVID-19-related news sentiment on S&P 500 index returns, volatility, and trading volumes from January to June 2020. Business and science COVID-19 related articles data from several sources like MarketWatch, Reuters, and the New York Times (NYT) were extracted and categorized through BERT modes by sentiment score, variance and volume of COVID-19 related news. Along with those data, other controlling variable like VIX index, OFR Financial Stress Index, global COVID-19 growth rate, and Google search trends for "coronavirus" were also incorporated to enhance model predictive power. Having employed a multivariate LR, the authors found statistical significance of NYT sentiment and MarketWatch sentiment on returns at 1% and negative correlation between NYT sentiment and trading volume at the 5% level.

Further studies from Koukaras et al. (2022) and Nousi and Tjortjis (2021), investigate the impact of public sentiment, derived mainly from Twitter and StockTwits, for the prediction of Microsoft stock price variations. Considering the fact that external economic factors may influence stock prices, the authors combined 90,000 tweets from Twitter and 7,440 posts from StockTwits between 16 July 2020 and 31 October 2020 with stock prices and adjusted closing values and, by using seven ML algorithms (KNN, Support Vector Machine, LR, Naive Bayes, Decision Tree, RF, and Multilayer Perceptron) in order to investigate their impact on forecasting stock prices.

For the classification of consensus sentiment and its intensity, authors employed TextBlob and VADER sentiment analysis tools in the twits. The results of their analysis highlight several remarkable findings for the StockTwits dataset combined with TextBlob. The highest performance among the models was reached by the SVM algorithm with an F-score of 68.7% and an AUC of 53.3%. On the other hand, when VADER sentiments were used on StockTwits data, both LR and SVM performed similarly (68% F-score) and with AUC values 55% for SVM and 54.75% for LR. That suggests that the model could predict increases in stock prices slightly better when using VADER than TextBlob.

In analyzing the Twitter dataset, which contained more data, results were even better. The SVM model with TextBlob sentiment analysis achieved an F-score of 75%, while RF model had an F-score of 69.6%, accurately forecasting the rise in stock prices

for 16 continuous days. KNN and DT models were two other strong performers, each with an AUC as high as 68% and F-scores equal to 72%. That indicated that though SVM successfully predicted stock movements, RF provided a more consistently accurate forecast over a longer period.

Finally, this study showed that the Twitter dataset produced the most reliable predictions overall, especially when combined with the VADER sentiment analysis tool. The SVM model, which had an F-score of 76.3% and an AUC of 67%, was the best among all, predicting 15 continuous days of increases in stock prices. This, therefore, indicates that Twitter data combined with VADER sentiment scores are effective in stock price forecasting and thus of great importance to those who seek to apply social media sentiment in their decision-making processes. In this regard, the study established that sentiment analysis-in particular, the VADER tool applied to Twitter data-offers a formidable method for the forecast of stock price movements.

In another study from Koukaras et al (2021), the authors examined the impact of Twitter data on stock market predictions for the period between December 2018 and July 2019, focusing on the issue of noise inherent in social media datasets. Their innovation was the employment of the PageRank algorithm due to its ability to assess user importance-based user follower and hence to prioritize relevant information by weighting tweets.

For the creation of the dataset, the authors focused on robust technical indicators for short-term trend analysis and Twitter data for which sentiment was quantified using VADER and TextBlob. Having utilized cashtags to filter stock-related tweets they further employed the graph theory analysis, which was conducted with the NetworkX library, and assigned daily PageRank scores across 242 user-follower graphs. HITS algorithm was also employed but it was proved computationally infeasible due to its lack of convergence. Based on the input features, they generated three different datasets called economic, sentiment, and PageRank-weighted sentiment and employed five ML models against the price of a 30-stock comprised portfolio.

XGBoost was found to deliver the lowest errors for 13 stocks with the greatest robustness across datasets, while economic dataset resulted the only profitable among the datasets with a 0.75% cumulative return. In all, authors suggested that their innovation to generate a PageRank dataset returned robust predictive accuracy but failed

to predict profitability, reflecting the complexity of aligning prediction models with actionable trading strategies.

2.3. Stock Market Forecasting through Machine Learning

Alshammari et al. (2022) presented a study on the prediction of stock prices using big data mining techniques. The purpose of their paper was to examine the Gulf Cooperation Council (GCC) and in particular Kuwait stock market by applying big data mining. The authors used a dataset consisting of historical stock prices and other fundamental financial indicators such as oil price, gold price, the exchange rate of Kuwaiti dinar (KWD) to US dollar (USD), money supply, interest rate, earnings per share (EPS), dividends per share (DPS) and Gulf stock market index namely, Kuwait, Oman, Saudi Arabia, Bahrain and Dubai. For prediction purposes they applied big data mining techniques such as regression, support vector machine, decision tree and RF to extract knowledge along with variables that affected the Kuwaiti stock market.

The study found that the use of big data mining techniques improved the accuracy of stock price predictions compared to traditional data mining techniques. The accuracy of the multinomial logic regression test was up to 54.24% while the confusion matrix of the polynomial kernel function of SVM achieved an accuracy of 52,73% which was relatively lower. Decision tree and RF test algorithms estimated accuracy almost at 53%. In all, authors noted that the scalability and processing power of big data technologies allowed for the analysis of large datasets in a reasonable amount of time with relatively similar results among the algorithms employed.

One of the most widely used data mining techniques in stock market analysis is NNs. NNs are a type of ML algorithm that can identify complex patterns and relationships in large datasets. In the stock market, NNs are often used to predict stock prices and identify trends. The most common type of neural network used in stock market analysis is the MLP model, which consists of input, hidden, and output layers. The input layer takes in the data, the hidden layer processes it, and the output layer generates predictions.

In a study by Hua et al. (2017), MLP NNs were used to predict the closing prices of six Chinese stocks. The authors found that the MLP model outperformed traditional

statistical models, such as the autoregressive integrated moving average (ARIMA) model, in terms of prediction accuracy. Yeh et al. (2019) used NNs to predict the future prices of stocks based on historical data. They collected data on market trends, company financials, and stock prices and used NNs to model the relationships between these factors. They found that NNs were an effective tool for predicting the future prices of stocks.

Decision trees are another popular data mining technique used in stock market analysis. Decision trees are a type of supervised learning algorithm that can be used for classification and regression analysis. In the stock market, decision trees are often used to identify patterns in stock prices and predict future trends.

In a study by Kara et al. (2011), decision trees were used to predict the direction of the stock market using technical indicators. The authors found that decision trees could accurately predict the direction of the stock market, with an accuracy of 75%. Basak et al. (2018) used decision trees to predict the future prices of stocks based on historical data. They collected data on market trends, company financials, and stock prices and used decision trees to model the relationships between these factors. They found that decision trees were an effective tool for predicting the future prices of stocks.

SVMs are a type of supervised learning algorithm that can be used for classification and regression analysis. SVMs work by finding the hyperplane that maximally separates the data into different classes. In the stock market, SVMs are often used to predict stock prices and identify trends.

In a study by Huang (2012), SVMs were used to predict the closing prices of 20 Korean stocks. The authors found that SVMs outperformed traditional statistical models, such as the ARIMA model, in terms of prediction accuracy.

Hu et al. (2022) conducted a research study that aimed to predict the future prices of stocks using SVMs with an improved training algorithm. The study collected data on two companies stock prices of the Chinese stock market for a period of two years from 2018 to 2020. The data was preprocessed to remove outliers and missing values. The study then used SVMs to model the relationships between these factors and predict the future prices of stocks. The study proposed an improved training algorithm for SVMs that used a hybrid optimization approach combining the particle swarm optimization (PSO) algorithm and the gradient descent (GD) algorithm. The proposed algorithm was

compared with other existing algorithms such as the standard PSO algorithm, the standard GD algorithm, and the traditional SVM algorithm.

The results of the study showed that the proposed algorithm outperformed the other algorithms in terms of accuracy and efficiency. The study found that SVMs with the proposed training algorithm were an effective tool for predicting the future prices of stocks. The study concluded that the proposed algorithm could be used as a practical tool for stock price prediction in real-world applications. Overall, Hu et al. (2022) contributed to the literature on data mining techniques used in stock market analysis by proposing an improved training algorithm for SVMs that was shown to be effective in predicting the future prices of stocks.

In another study that was conducted on Vietnam stock index closing prices and news information from publications, Hoang (2014) proposed a model in which the accuracy achieved was 75%. For their estimation the applied support vector machine algorithm which was also combined with linear SVM.

RF is a type of ensemble learning algorithm that combines multiple decision trees to improve prediction accuracy. In the stock market, RF is often used to predict stock prices and identify trends.

Klein et al. (2016) conducted research that aimed to forecast oil price volatility using new hybrid models that incorporated intraday data. The study collected intraday data on oil prices and market indicators such as stock market indices and exchange rates for the period from January 2009 to December 2016. The data was preprocessed and cleaned to remove outliers and missing values. The study then used a hybrid approach that combined various models such as GARCH, SVR, and SVM to forecast the volatility of oil prices. The study proposed two new hybrid models that combined the GARCH-SVR model and the GARCH-SVM model. The models were compared with other existing models such as the GARCH model, the SVR model, and the SVM model. The performance of the models was evaluated using various statistical measures such as root mean square error (RMSE) and mean absolute error (MAE).

The results of the study showed that the proposed hybrid models outperformed the other existing models in terms of forecasting accuracy. The study found that the GARCH-SVR model and the GARCH-SVM model were effective in forecasting the volatility of oil prices using intraday data. Overall, Klein et al. (2016) contributed to the

literature on forecasting oil price volatility by proposing new hybrid models that incorporated intraday data and outperformed other existing models in terms of forecasting accuracy. The study demonstrated the importance of using hybrid models and intraday data for accurate forecasting of oil price volatility.

Association rule mining is a data mining technique used to identify frequent patterns and relationships in datasets. In the stock market, association rule mining is often used to identify the relationships between different stocks and their prices.

Ahn and Han (2017) conducted a research study titled "Stock price prediction using deep learning on financial news" published in the Expert Systems with Applications journal. The study aimed to predict stock prices using deep learning techniques on financial news articles. Financial news articles were retrieved from the Reuters news agency for the period from 2005 to 2015. The articles were preprocessed using NLP techniques such as tokenization and stemming to extract relevant information. The study then used a deep learning model called a convolutional neural network (CNN) to analyze the news articles and predict the stock prices. The study compared the performance of the CNN model with other existing models such as the RF model and the support vector machine (SVM) model. The performance of the models was evaluated using various statistical measures such as accuracy, precision, recall, and F1-score.

The results of the study showed that the CNN model outperformed the other existing models in terms of predicting stock prices using financial news articles. The study found that the CNN model was able to extract useful information from the news articles and capture the complex relationships between the news and the stock prices. Overall, Ahn and Han (2017) contributed to the literature on stock price prediction by demonstrating the effectiveness of deep learning techniques on financial news articles. The study showed that the CNN model was a promising tool for predicting stock prices using textual data and could be used in real-world applications such as financial investment and trading.

Bhandari et al. (2022) analyzed the relationship between stock market index S&P500 and fundamental, macroeconomic and technical data by applying deep learning techniques in order to set up a model able that would predict the future closing price. The data consisted of macroeconomic indicators that have significant impact on the stock

market performance such as index-based options index, federal funds rate, unemployment, CEI and US dollar index. In addition, they enhanced their dataset by adding technical indicators that are designed to analyze short term movements and volatility in the market such as Moving Average Convergence Divergence (MACD), Average True Range (ATR), and Relative Strength Index (RSI). Having checked the correlation of their data, they continued to prepare the dataset though Wavelet transformation, which is suitable for denoising stock price data, and normalization, in order to address the issue that the range of one feature varies greater than others. The transformed data were then utilized in the implementation of single layer and multilayer LSTM architectures to predict the closing price. For each architecture, different sets of hyperparameters were tested with the overall conclusion being that the single LSTM model with 150 hidden neurons providing the best fit and the highest prediction accuracy.

In particular, the prediction metrics scores for RMSE (square root of the mean square error), MAPE (size of the error computed as the relative average of the error) and R (linear correlation between actual and predicted values) when the single layer LSTM with 150 neurons where corresponding to 40,45, 0,7989 and 0,9976 outperforming any of the other combinations.

In Agrawal et al. (2019), they proposed an optimal LSTM deep learning architecture and adaptive Stock Technical Indicators (STIs) concept in order to predict the price and trend of three banking stocks listed in National Stock Exchange (NSE) – India. In particular, they calculate the most prevailing STIs, such the RSI, Moving Average (MA) of n days, Stochastic Oscillator (%K), William (%R), EMA and MACD and then they calculate the correlation between pairs of technical indicators using Pearson correlation. Then they introduced the concept of Correlation-Tensor where they transform the correlation vectors of STIs to tensors allowing for richer data representation. These tensors are then fed into the Optimal LSTM model in order for the model to capture relationships that would forecast stock prices.

The results of their work are also compared to Support Vector Machine (SVM), Logistic Regression (LR) and one deep learning model (ELSTM) and show that the highest accuracy and mean accuracy are 65.64% and 59.25% respectively, which are much higher than SVM, LR and the deep learning approach (ELSTM).

In another study from Chang et al. (2024), the focus was on the prediction of economic trends and stock market prices applying advanced ML techniques, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. In particular, the study aimed to forecast the technological sector by examining four major tech companies. To enhance the robustness of the major models applied in their dataset, the authors compared their results to those derived from traditional methods, such as ARIMA, Facebook Prophet and XGBOOST. They split their dataset into 80% for training and 20% for testing, while they also employed cross validation and min-max normalization to achieve model reliability and data consistency. Their neural network architecture starts with an input layer that consists of time series data over a defined previous period, followed by an LSTM layer of 128 units aiming to capture long term dependencies. A GRU layer with 128 units is also added for computational efficiency purposes and two dense layers of 64 and 32 units respectively lead to the final output layer.

Evaluating the results of their models, they indicated that the GRU model generally outperformed for most metrics the LSTM model. For example, for Apple stock the RMSE of 3.43 and an MAE of 6.53 obtained by the GRU model is notably lower compared to the LSTM's RMSE of 9.15 and MAE of 7.81. Further compared to results from literature, the GRU model lacked the prediction capability of S-GAN model while performing better than traditional ARIMA model. Finally compared to other models, the authors found that for all stocks LSTM outperformed XGBoost which had also performed relatively well compared to ARIMA and Facebook Prophet which were lagging significantly.

In another study from Sangeetha and Alfia (2024), the authors applied Evaluated Linear Regression-based ML (ELR-ML) technique on S&P 500 price index related data such as Open, Close, Low, High, and Volume, aiming to forecast stock values. After performing noise removal and feature selection data preprocessing techniques that would enhance model's prediction accuracy, they applied the ELR technique to find relationships between stock prices and price index related dependent features.

Their results indicated a moderate accuracy for their model with a correlation (R2) of 0.358 and relatively accepted volatility prediction considering that the dataset is spanning during the financial crisis period. In overall, the study raised limitations using

LR for the prediction of nonlinear stock price movements but suggested the potentiality for improvement with a more sufficient dataset and training.

In another study from Yao and Yan (2024), short-term trends in China's stock market are examined under the concept of data segmentation to strong, weak trends and fluctuations to enhance prediction accuracy. They evaluated SE large-cap, mid-cap, and small-cap indexes as sample data for the period (2011-2022) to represent effectively and reliably the Shanghai Stock Exchange by applying a DLWR-LSTM model construction. Initially, they used the DLWR methodology to separate the stock data into distinct layers that represent strong trends, weak trends, and noise and derive trend and fluctuation from their dataset and then they normalize it to values 0 to 1. For the LSTM methodology to be applied, they built a model with three LSTM layers, two dense layers, and dropout to prevent overfitting, optimizer set to "adam" with 20 epochs, batch size 16, and validation split to 0.1.

Prediction results indicated that the model effectively captured short-term trends as the number of separations increased and compared to other traditional approaches such as ARIMA, DLWR-ARIMA, LSTM, the DLWR3-LSTM model provided the lowest MAPE and RMSE and R^2 near 1, offering high accuracy and robustness across different market volatilities.

In Tsai et al (2023), the authors analyzed mid-term to long-term Taiwanese stock market trends given the effect of technical indicators, financial ratios and sentiment analysis. The distinctive feature of their study is the introduction of the intrinsic values of companies as financial ratios (liquidity, leverage, asset efficiency, market value, and profitability ratios) in the concept of market price prediction given the fact that they reflect companies' financial health and management quality, and prices may not always reflect intrinsic values due to external factors. Their methodology is built on a set of 97 stocks from the Taiwan Stock Exchange which is used to create top 10 and top 20 portfolios, the TW50 index which serves as a benchmark portfolio and 18 financial ratios among other features. All features are standardized to ensure consistency and then trained by employing various ML models (RF, Feedforward Neural Network, Gated Recurrent Unit and Financial Graph Attention Network) in order to define the training target of relative quarterly returns. In order to assess the results, they use High Portfolio Scores, Excess Returns, Risk-Return Relationship and Top-k Precision metrics. FinGat

model provides the best results among other models regarding the High Portfolio Scores metrics, while for Excess Returns both RF and FinGAT models exceeded returns above 100% for all tested portfolios.

For Top-k Precision metric, all models achieved precision percentages of 16.4% to 21.8% for top 10 portfolios, outperforming the 10.3% baseline of random selection, while for top 20 portfolios, precision ranged from 26.8% to 29.5%, compared to a baseline of 20.6%. In overall, their study confirmed that the models employed given these features are capable enough to provide 10 stock portfolios with high excess returns and acceptable risk or 20 stock portfolios with lower risk and still achieve excess returns compared to TW50 index.

Kim et al. (2024) challenged the current research by introducing the need to pursuit effective combinations of features in order to accurately predict the stock market. Given the fact that there are various elements affecting the stock market, which are not possible to be represented by technical indicators, and considering at the same time the necessity to define the ideal combinations of these elements for accurate stock price predictions, the authors employed 16 feature selection algorithms on six feature classes (price indicators, technical indicators, economic indicators, financial indicators, fundamentals, and market sentiment) across ten sectors. The algorithms were divided into filter, wrapper, embedded and ensemble methods producing 60 configurations per sector by adjusting parameters like “feature selection percentile” (e.g., keeping the top 10%, 20%, 30%, etc., of features). Then, RF, LightGBM and LSTM evaluation models were employed to test the performance of the selected features on the stock price prediction concept. Results of their study showed that the wrapper-based methods and especially SHAP provided the optimal combination of features to be used for the prediction of sector index trends.

Sunantha (2020) studies the efficiency of NNs versus Ordinary Least Squares Regression (OLSR) on a dataset of 151 companies from various sectors for the period between 2002 and 2015 in order to predict the Shanghai Stock Exchange market volatility. In their study, they examined the effect of 21 independent variables related to macroeconomic conditions, market sentiment, and institutional factors and evaluate their results using the Absolute Percent Error (APE). NNs present higher capability in capturing complex patterns in the consumer goods and finance sectors, while traditional

Ordinary Least Squares Regression method demonstrates higher accuracy in conglomerates, healthcare, and industrial goods sectors.

The study from Jabeur et al (2024) articulates in detail the criticality of gold as a financial asset, especially in turbulent economic periods where it acts as a hedge against stock market downturns. In the light of the importance of gold to investors, mining companies, and economies worldwide, the study introduced a new method of gold price forecasting using various ML techniques combined with SHAP technique in analyzing feature importance. While most of the literature combines ML techniques with factor detection methods, this study directly evaluated six models like LR, NNs, RF, LightGBM, CatBoost, and XGBoost using common metrics such as R^2, RMSE, and MAE. All the models employed on diversified monthly data including gold and silver prices, crude oil prices, exchange rates (USD/EUR, USD/CNY), inflation rates, and S&P 500 index. The best performance was found for XGBoost yielding an R^2 of 0.994 and an RMSE of 34.921. XGBoost outperformed other models like CatBoost and RF, hence giving more confidence in its strong predictive ability.

The SHAP approach provided also a clear insight into how each variable varies affects gold prices. In particular, SHAP dependence plots reflected that higher Chinese exchange rates increased gold price volatility, especially in response to variation in crude oil prices. In terms of local interpretability, SHAP technique highlighted the value of the features impacting the predictions at each observation level. Features such as China's exchange rate and inflation tend to drive down gold prices, while those related to silver and crude oil had a positive impact. In overall, this work reinforced financial forecasting by suggesting that XGBoost is an efficient ML method for predicting the price of gold and presented the usefulness of SHAP in interpreting complex ML models. These findings offered key insights into gold price drivers for investors and policymakers.

Papageorgiou et al. (2024), examine the potential of predicting NVIDIA (NVDA) stock price using a reinforcement learning algorithm. In particular, they applied a Deep Double Q-Network (DDQN) algorithm in a three-phase approach to enhance the model’s predictive capability and profitability by progressively incorporating financial (phase 2) and sentiment indicators (phase 3) to the initial training phase of using closing prices only. The financial indicators they employed in their study were in essence technical indicators that capture trend and volume, while for sentiment indexing, they

applied Twitter-roBERTa-base method on user posts tagged with $NVDA that were gathered from StockTwits platform. Data utilized in the study span from 2020 to 2023, reflecting a significantly volatile period for tech market. Their model architecture was based on the careful design of an agent that predicts the stock price changes, signaling BUY and SELL, and a reward system that encourages the agent to take the relevant signal action. They also optimized their model in order to refine the decision-making process by applying advanced optimization techniques such as experience replay memory, step-decaying learning rate and decaying epsilon-greedy approach.

After implementing the model, the authors showed that for both training and evaluation phases DDQN model's trading performance improved as far as the data complexity was increasing but at the same time variability in outcomes was also introduced adding uncertainty in their predictions. All over, their study suggested that optimized DDQN model has high potential to predict the market when applied to progressively incorporated dimensions of different financial related data features.

3. Data

In this section, a detailed presentation of the features employed in this study is analyzed, indicating the sources and the reasons for being considered important to be studied. In the last part of this section, features are also statistically analyzed in order to provide an insight into their basic statistical analytics and their dependencies across the dataset.

3.1. Stock Market Data

The S&P 500 Index, or Standard & Poor's 500, is an American stock market benchmark index composed of the 500 largest publicly traded companies listed on U.S. stock exchanges. Therefore, this index serves as a broad indicator of trends in the market since it represents companies from all sectors, serving in that way as a gauge to the performance of the overall U.S. economy. In addition, given the fact that S&P 500 Market Cap Weighted Index reflects the level of influence of each company on the market capitalization of the Index, then the larger the company, the bigger the impact it will have on the price movement of the index (Investopedia, 2024).

S&P 500 Index is being closely monitored by investors, policymakers, and analysts as an important indicator of the overall health of the US financial market. Depending on the magnitude of the price change, every movement in the index reflects the macroeconomic conditions of the wider economy and their corresponding microeconomic effect in different market sectors. Hence, its performance is indicative of investors' confidence and changes during economic cycles, among other factors. For instance, the S&P 500 also serves as a benchmark for many mutual funds and ETFs since its performance is a proxy of the direction of the U.S. market and overall economic stability (Investopedia, 2024).

With regards to the examination period that spans from 2008 to 2016, it is important to make a historical overview of the critical events that took place and heavily influenced the S&P 500. During the financial crisis of 2008, the index experienced extreme turmoil that was mainly caused by the fall of the housing market and extensive problems within the financial sector. which in turn led to recession, seriously affecting the United States and world economies. During that time, and in March 2009, the index

plummeted to its lowest point (Figure 1). Thereafter, it started recovering when the Federal Reserve and the U.S. government began introducing economic stimulus to stabilize the financial markets (Investopedia, 2023).

Figure 1. S&P 500 Index Price Evolution 08/2008 - 05/2016

source: macrotrends.net

During the post-recession period, the U.S. economy had slowly recovered with the help of low interest rates and other accommodative policies. These conditions generated the incentive for investors to turn into bull their investment decisions and slowly strengthen the market, a process that extended until 2015 and was characterized by significant gains for the S&P 500, reviving any confidence being lost previously. Then, in 2015, volatility of the prices resumed, mainly due to global concerns for the slowing economy of China and the uncertainty over oil prices. This followed period was again characterized by instability that perpetuated fluctuations in the index but not to such an extent as the recession of 2008 that plummeted the Index trend. (Investopedia, 2023).

Considering the above, S&P 500 is an important feature for study due to its all-inclusive nature. The period 2008 to 2016 serves satisfactory to reflect the larger reality of economic trends as it incorporates a recovery phase following a financial crisis. Understanding the index volatility during that period which was caused by many factors, including investor sentiments and macroeconomic effects, and employing them against the price movement for prediction purposes, is an important input into economic and financial modeling that will support investors and policymakers to take the correct decisions in the future.

3.2. Sentiment Data

Sentiment analysis has become increasingly important in recent years for the prediction of stock markets since it reveals insight into the mood of the general public or investor sentiment that often precedes market trends. Analysis of the tone and content in news and social media can accurately represent any change in mass opinion and psychological reaction to various global events that often precede the movement of financial markets. As most of these sentiments are bound to have an impact on short-term market dynamics, incorporating them within modelling tools can offer richer insight for predictions on stock indices like the S&P 500.

A typical example of such sentiment-based data is the dataset employed in this study comprising top-ranked historical headlines from the WorldNews subreddit, /r/worldnews, starting from June 8, 2008, and ending on July 1, 2015. This dataset incorporated the top 25 most upvoted daily headlines that show public interest and sentiment toward major events happening globally on a daily basis. On Reddit, users have options to upvote or downvote news posts, so only those that get the most attention and attraction can climb the ranking lists. Therefore, given that headlines capture leading issues and people's feelings, this dataset acts as an insightful indicator of market sentiment variation over time.

This strength of this dataset is its authenticity that is derived from the user-driven nature of Reddit to rank top headlines based on community engagement. This key feature not only underlines critical world events but also reflects what these events are perceived to be of importance to the public. Sentiment analysis of the headlines, when scored and quantified, would therefore give a strong basis for measuring collective sentiment on dates and correlating it with stock market performance.

Incorporating the sentiment data from the Reddit WorldNews dataset into the macroeconomic and technical indicators brings a whole new dimension to having insight into the prediction of stock markets. In contrast to the structural data from the macroeconomic indicators or historical patterns that are being portrayed by technical indicators, the sentiment data picks up on the public's psychological reaction to major events happening around the world, offering real-time context that often actually lines up with the markets' reaction. As an example, shifts in sentiment regarding geopolitical tensions or announcements related to economic policy create sudden movements in the stock indices. This integration of sentiment scores with conventional and macroeconomic data sources will enhance model accuracy for S&P 500 index predictions instead of utilizing only conventional indicators.

3.3. Macroeconomic Indicators

The S&P 500 represents one of the most critical benchmark indicators of U.S. stock performance. It reflects the performance of 500 leading companies in the country and has been generally used as an indicator of broader economic health. Its importance therefore requires a quite accurate forecast of this indicator's trend with immense value to investors, economists, and policymakers. In order to capture the broader economic health in the US, a set of macroeconomic features that incorporate indirectly the view of individuals and policymakers in the US is employed along with a global macroeconomic index. The macroeconomic indicators of this study include the US 10-Year Treasury Bond Note Yield, ISM Manufacturing PMI (PMI), US Equity Market-related Economic Uncertainty Index, Economic Policy Uncertainty Index (EPUI), CEI, and Business Confidence Index (OECD), which all signal different aspects of the economic state that may impact the S&P 500. Each one of these variables is discussed in this paper with respect to its potential relevance as an input for S&P 500 predictions, discussing a causal chain which may be used to link those variables to the market.

The US 10-Year Bond Yield is a proxy for long-term investor expectations about economic conditions and inflationary pressures. It is a "risk-free" rate of return, and any changes to this yield have a strong effect on the overall financial markets, such as the price of the S&P 500. Increases in bond yields reflect upward revisions in expectations about future inflation and possible higher borrowing costs for companies. Higher costs weigh on corporate earnings and therefore on stock prices, such as those in the S&P 500.

Moreover, when bond yields rise, bonds start to look more attractive compared to stocks; that might trigger a flow of capital from equities into bonds, dampening stock prices (Damodaran, 2012). Conversely, if the 10-year yield falls, then borrowing becomes less expensive and stocks become more appealing relative to bonds. That may be a very favorable environment for the growth-oriented sectors capitalized under the S&P 500, enticing investors to undertake more equity exposure. The yield curve-the continuum of interest rates from the short term to the long term, within which the 10-year yield is included-also acts as a guide on recession risk. The inversion-a condition wherein the yields of the short-term go above the long-term rate-has historically been a prelude to economic recessions and could foreshadow some weakness in the S&P 500 (Fama, 1986).

The PMI is one of the most used indicators related to the economic health for the manufacturing sector in the United States. This indicator contains data about changes in production levels, new orders, employment, and inventories of the manufacturing industry and gives insight into supply chains and business activity. Readings above 50 on the PMI signal expansion, while readings below 50 points to contraction. The S&P 500 contains several companies that are directly or indirectly involved in manufacturing, so changes in PMI usually result in marked movements in the index. Generally speaking, a high PMI is indicative of economic growth, whereby companies benefit through increases in production and a growth in revenue, promoting positive investor sentiment toward the S&P 500. On the other hand, a declining PMI could signal weaker demand and reduced revenues going forward, especially for economically cycle-sensitive sectors like industrials and consumer discretionary stocks. For this reason, the PMI can also be applied as a leading indicator to project subsequent quarters of the S&P 500 performance based on current levels of manufacturing activity (Investopedia, 2024).

The Equity Market Uncertainty Index (EMUI) measures the anxiety of the investors and the volatility of the market. This index captures the extent of uncertainty concerning economic conditions relevant to the equity markets and focuses on factors such as fiscal policies, trade tensions, and geopolitical risks. Higher uncertainty times exhibit more volatility in the S&P 500, as investors react to the possibility of risks in order to reassess portfolio weightings (Baker et al., 2016). During periods of high uncertainty in the economy, investors prefer to switch to safer assets. This can be one of the reasons for taking investing positions in other assets from equities-a thing that

should, in principle, affect negatively the S&P 500. Lower uncertainty enhances riskier decisions and may encourage investment in equities. For this very reason, the EMUI is an important input to the S&P 500 prediction models: high uncertainty usually goes along with greater market swings, which often involves downward pressure on the index (Gulen & Ion, 2016).

The EPUI epitomizes the level of uncertainty related to economic policy, which in turn would have a considerable impact on market behavior. Some of the causes of EPUI are fiscal policy, trade policy, regulation, and international relationships. High levels of economic policy uncertainty may make corporations invest less and consumers more cautious. Thus, a scenario like that would be expected to affect corporate profitability and, consequently the returns on, S&P 500 performance (Baker et al., 2016). High readings of EPUI are usually synonymous with high market volatility and a conservative outlook on corporate earnings. For instance, companies, in the face of a highly policy-uncertain environment, may postpone investments or expansion based on uncertainty over possible changes in regulation or trade. This would depress stock valuations and retard growth in the markets. On the other hand, a low EPUI environment supports business expansion and investment, therefore it may ensure positive performance within the S&P 500. Hence, the EPUI can be considered one of the most important determinants of market trends, because changes in policy certainty immediately affect investor sentiment and stock prices (Pastor & Veronesi, 2012).

The CEI is a measure of the attitude of consumers concerning their personal finance, business conditions, and economy in general. Since consumer spending encompasses over a third of the total economic activity, consumer sentiment directly and immediately affects many S&P 500 sectors, especially those in retail, consumer discretionary, and financials. A strong CEI suggests solid consumer spending and reflects a healthy economy and perhaps stronger revenues for S&P 500 companies. Increased consumer confidence translates into increased spending on goods and services, which directly benefits industrials in the retail, automotive, and entertainment sectors-all well-representative in the index. Conversely, low consumer sentiment indicates that consumers may pull back on spending, which negatively impacts those sectors and weighs down the S&P 500. By incorporating consumer sentiment into predictive models, this study gains a better understanding of how these consumer-driven sectors might perform within the index (Ludvigson, 2004).

The OECD BCI captures business managers' opinions concerning output, new orders, and prospects of the global economic conditions. A high confidence level among businesses suggests that companies may perceive favorable conditions ahead and will probably increase hiring, capital investment, and production. Such a sentiment is positive for stock prices as well, because increased investment and hiring by companies are most likely to result in better revenue and profitability, hence positive for the S&P 500 (OECD, 2021). Conversely, when business confidence is lower, it might indicate a belief that the economy is going to take a turn for the worse and that people will be very cautious about spending and investment. This limitation in spending weighs on the stock market, particularly in capital-intensive industries. Including the BCI in S&P 500 forecast models enables the encapsulation of trends in corporate sentiment that may influence market performance. Good business confidence may improve market expectations for technology, finance, manufacturing sectors, among others, and all of these are significant constituents of the S&P 500 (Christiano et al., 2014).

These macroeconomic indicators were selected because they represent a wide range of the economic drivers of the S&P 500, like interest rates, the strength of manufacturing, stability of policies, consumer spending behavior, and business sentiment. Other indicators do exist, but this particular combination reflects immediate and long-term economic influences and therefore is a robust input set for predictive modeling of future index prices. This set of factors provides a comprehensive overview of the economic and market conditions that drive performance in the S&P 500, offering a view that is both anchored in the cyclical business indicators themselves and responsive to sudden changes in economic uncertainty.

The choice of indicators provides insight into the study in several dimensions of the economy, such as the trend of production, confidence in investment, inflation expectations, and also consumer behavior. All these together allow a more encompassing approach toward the forecast for S&P 500 movements and, therefore, enhancements in accuracy and robustness for the forecasting modelling.

3.4. Technical Analysis Indicators

Technical indicators are key components in predicting stock market movements that have been heavily used in academic literature. While fundamental analysis makes estimates on the intrinsic value of a company, TA encompasses the study of the

movement in prices and their patterns over time to predict future movements in the market. This paper examines the following five technical indicators: RSI, Stochastic Oscillator, Williams %R, EMA, and MACD. Each of these indicators has its own strengths when it comes to suggesting the future performance of the S&P 500 through various ways of inspection: indications of momentum, overbought or oversold conditions, and trend direction.

RSI is an indication of the speed and magnitude of the changes in the price series-usually over a 14-day period-which finds overbought or oversold conditions. RSI fluctuates between 0 and 100, with values above 70 considered overbought and those below 30 considered oversold. In principle, the RSI will tend to be used more effectively in highly liquid markets, such as that of the S&P 500, where large institutional investors may easily push the price toward a wider volatility. The RSI is useful under the idea that it will help the trader not just locate the points of probable reversal but also provide reinforcement for other trend-following indicators, such as the MACD, toward correct predictions in the trends concerning direction and strength. The RSI focuses on price momentum and mean reversion, making it apt for predictions related to the S&P 500, as overbought and oversold signals quite often appear before the corrective movements. Additionally, RSI is viewed as reliable by a wide number of market players in major indices because it catches high-frequency shifts in investor sentiment, thus acting as an early warning system for reversals (Murphy, 1999).

Stochastic Oscillator is one of the momentum indicators, which compares a certain closing price with its range for some period. Computed with the help of %K and %D values, this indicator identifies potential points of reversal and the state of overbuying or overselling. The stochastic oscillator tends to work well with the highly dynamic price action that characterizes the S&P 500, since such an oscillator catches the relative position of the closing price of a certain stock within a high-low historical range. When used in conjunction with RSI, it fortifies trend identification, as a high stochastic value with an overbought RSI depicts the likelihood of a reversal. It is most effective in determining the S&P 500 trends, which tend to appear in a seesaw-pattern in an enclosing trend. The S&P 500 is very sensitive to economic data and to investors' sentiment, matching well with the short-term trend information coming from the Stochastic Oscillator, thus enabling traders to take advantage of intraday or at best short-term market fluctuations (Murphy, 1999).

Another oscillator used is the Williams %R, which also gives an indication of overbought and oversold conditions but on its own scale-from -100 to 0. Readings above -20 are indicative of overbought conditions, while readings below -80 suggest conditions of being oversold. The indicator is good for the S&P 500 because it gives rapid insights into the relative strength or weakness of price action within a recent period. Above all, Williams %R is useful because it will often give indications of reversals well in advance of most other indicators, especially in markets that show volatility-a common factor in index trading. This sensitivity makes Williams %R suited to markets such as the S&P 500, where strong institutional shifts can affect short-term price fluctuations. The quick response of Williams %R could make this indicator effective in showing points of entry and exit, hence enhancing decision-making for the trend following strategy (Murphy, 1999).

EMA is an indicator following trends that gives more prominence to recent prices, thus being sensitive to recent price action. Compared with a simple moving average, an EMA reacts more quickly to changes in price direction and can thereby enable traders to identify and follow trends in the S&P 500. The EMA is mostly helpful for identifying spot trends and filtering out market noise, common in large indices influenced by macroeconomic factors. This will provide a view of both the current momentum and the longer-term trend direction in S&P 500 forecasts, using both short-term-e.g., 10-day or 20-day-and longer-term EMAs, such as 50-day or 200-day. Crossovers between short- and long-term EMAs are, as a matter of fact, usually very solid signals for reversals or continuations in trends, thus EMA is very helpful in any swing trading strategy to take advantage of catching larger price movements (Murphy, 1999).

The Moving average convergence/divergence (MACD) embeds both trend-following and momentum because the difference between the 26-day EMA and the 12-day EMA yields a MACD line, which, together with the signal line derived from the nine-day EMA of the MACD line, forms a crossover system. The crossing of the MACD line above the signal line defines the bullish trend, and, conversely, the bearish trend is defined when the line crosses below the signal line. Often, subtle shifts in momentum before an actual crossover can be detected through an MACD histogram charting the difference between the MACD line and the signal line. This is particularly relevant for the application of MACD in the S&P 500, as the confirmation of longer-term trend shifts

using the MACD serves to prevent false signals that are characteristically found in shorter-term fluctuations. Since the S&P 500 is susceptible both to macroeconomic factors and to investors' emotions, the MACD becomes an appropriate tool that captures prevailing trends in the market and confirms other indicators, including the RSI and EMA (Murphy, 1999).

These indicators will suit very well in momentum, trend following, and overbought/oversold detection for the S&P 500. Among dozens of various indicators, RSI, Stochastic Oscillator, Williams %R, EMA, and MACD are especially fit for index forecasting because they balance sensitivity and confirmation of trends well. With this set of indicators, dependency on any single indicator is reduced by providing a broad view of the market's direction. This makes the approach even more stable and predictable to trade the S&P 500 with (Damodaran, 2012).

The technical indicators to be discussed will include RSI, Stochastic Oscillator, Williams %R, EMA, and MACD-all, which are very useful in giving a robust framework for predicting S&P 500 movements. Their effectiveness lies in the fact that they are complementary to one another: RSI and Stochastic Oscillator provide momentum and reversal points, Williams %R adds sensitivity to overbought/oversold states, while EMA and MACD provide some trend-following insight. Indicators are combined to improve the accuracy of S&P 500 forecasts by identifying different market conditions. Whereas other indicators are more notable for their supportive contributions, such as Bollinger Bands or Parabolic SAR, the chosen indicators present a balanced mix that fits well with the market dynamics of the S&P 500 and thus are particularly appropriate for index prediction.

3.5. Descriptive Statistics

Descriptive statistics are major checkpoints in the analysis of the dataset, allowing an overview of its general characteristics. The statistical computations made such as mean, median, variance, and standard deviation show the distribution of data, outliers, and possible errors in data entry. This initial overview allows us to detect patterns, skewness, or anomalies that may impact the analysis. Based on the results, the preprocessing steps and model choices are defined with guarantees on accuracy and reliability.

3.5.1. Descriptive Statistics Table

The descriptive statistic table (Table 3-1) shows all the features that are considered in the prediction of the closing price of S&P 500 index. Skewness, kurtosis, mean, median, standard deviation, and variance are major statistics that give insight into the distribution and dispersion of each feature, which may be important to the model of prediction.

Starting with Adj Close statistics we get a mean that equals 1,485.89 while the median is quite near to mean value of 1,385.22, both demonstrating an approximately symmetrical adjusted close price, with good support from the quite small value of the skewness that is equal to 0.138. The large value of standard deviation at 403.23 and large variance of 162,594 show that these close rates have significantly varied and may probably be indicative of market volatility in the period under observation.

The technical indicators of the dataset show diverse behaviors, especially RSI, Stochastic Oscillator and William %R. RSI has an average of 52.98, that is close to a neutral value of 50, with low skewness. This suggests a balance between overbought and oversold conditions. The values of the Stochastic Oscillator (%K and %D) have averages of about 67, indicating a soft bullish trend showing a slight negative skewness. William %R has an average of -37.86, which indicates an oversold behavior, but the substantial high variance and standard deviation of 989.87 suggests that it fluctuates wildly, maybe even in cycles.

For Moving Average and MACD we get an EMA's average that stands at 1483.26, relatively close to the average of the Adj Close and should therefore be a good signal of the trend direction. Skewedness for both the MACD_Line and MACD_Signal statistics is very low, averaging about 2.8 and 2.77, respectively, with limited variability. The MACD_Diff has an approximate zero mean, meaning that in the long run, price and moving averages have diverged very slightly. The very slight skewness across these variables suggests that trends are fairly stable without strong directional biases.

With regards to the sentiment scores, both sentiment indicators from Textblob and HF have means close to zero, reflecting that overall, neutrality may be maintained in the data set. However, the Sentiment Score_HF is positively skewed at 0.1, indicating intermittent spikes in positive sentiment, while the Sentiment Score_Textblob is -0.43-skewed. The very low standard deviations and variances of these scores indicate that the

sentiment remains basically stable but would always have the potential to impact near-term price movements whenever peaks or troughs occur.

The macroeconomic indicators such as EPUI, EMUI, TB_Yield_10Y, BCI, CEI, and ISM_PMI are very heterogeneous concerning their statistical properties. For example, the average of the variable EPUI is high with large variance, reflecting great uncertainty in economic policy. In contrast, the variable EMUI has a mean value of 49.49 and, therefore, is much less volatile. TB_Yield_10Y has an average value of 2.59 reflecting low volatility and hence stability in the yields of bonds and hence suggesting a stable interest rate environment for the period under consideration. ISM_PMI is one manufacturing indicator whose distribution is relatively balanced, with a mean of 52.56 and a variance moderate. This suggests the existence of a moderate economic activity without extreme changes.

In overall the dataset reflects a mix of steady macroeconomic factors and more dynamic technical indicators. The stable sentiment indicators and macroeconomic indicators can outline the baseline for the long-term trend prediction, while technical indicators and EMA may capture short-run fluctuation. Besides, minimal skewness across most indicators suggests limited outliers, which could make for a good prediction model with less extensive data preprocessing or outlier treatment.

Table 3-1. Descriptive Statistics

Descriptive Statistics Table

	mean	median	std	variance	skewness	kurtosis	IQR	min	25%	50%	75%	max	count
Adj Close	1485.894567	1385.220032	403.229724	162594.210582	0.138256	-1.275786	729.917511	676.530029	1160.827515	1385.220032	1890.745026	2130.820068	1966.000000
RSI	52.988254	53.548412	5.557736	30.888429	-0.509530	-0.224899	7.892896	33.960915	49.523602	53.548412	57.416498	65.726521	1966.000000
Stochastic Oscillator (%K)	67.917544	76.258843	28.705745	824.019798	-0.737644	-0.621538	46.204411	0.000000	47.045416	76.258843	93.249828	100.000000	1966.000000
Stochastic Oscillator (%D)	67.892492	76.013365	27.909082	778.916840	-0.715545	-0.706948	46.012744	0.777704	46.867125	76.013365	92.879870	99.790222	1966.000000
William (%R)	-37.865488	-30.221116	31.462151	989.866919	-0.486257	-1.115595	56.393561	-100.000000	-65.105443	-30.221116	-8.711882	-0.000000	1966.000000
Exponential Moving Average (EMA)	1483.226097	1385.388161	401.036726	160830.455515	0.151201	-1.284325	740.833329	727.137153	1164.180809	1385.388161	1905.014138	2116.920616	1966.000000
MACD_Line	2.801488	6.337494	15.986559	255.570067	-1.598440	3.798191	18.070276	-77.201147	-3.891083	6.337494	14.179193	32.131558	1966.000000
MACD_Signal	2.775295	5.997103	15.036404	226.093433	-1.597104	3.662913	16.980360	-71.052216	-3.488211	5.997103	13.492149	28.639367	1966.000000
MACD_Diff	0.026193	-0.035807	4.885209	23.865269	-0.304511	1.975953	5.723394	-26.649529	-2.804142	-0.035807	2.919252	18.291069	1966.000000
Sentiment Score_Textblob	0.011907	0.012190	0.041069	0.001687	-0.043422	0.495208	0.053398	-0.166995	-0.014890	0.012190	0.038509	0.197128	1966.000000
Sentiment Score_HF	-0.574532	-0.576970	0.164135	0.026940	0.301099	0.101479	0.224733	-0.983633	-0.696725	-0.576970	-0.471991	0.074818	1966.000000
EPUI	117.025402	100.290000	71.376623	5094.622310	1.676673	4.967145	84.640000	3.320000	65.955000	100.290000	150.595000	626.030000	1966.000000
EMUI	49.491968	27.435000	73.448102	5394.623652	5.555109	50.946801	45.145000	4.800000	11.990000	27.435000	57.135000	1117.230000	1966.000000
TB_Yield_10Y	2.590450	2.520000	0.663753	0.440569	0.381743	-0.998890	1.090000	1.430000	2.020000	2.520000	3.110000	4.080000	1966.000000
BCI	99.646433	99.999110	1.320529	1.743796	-1.936101	3.245388	0.882810	95.274550	99.545990	99.999110	100.428800	101.238000	1966.000000
CEI	76.856968	75.100000	10.592832	112.208091	0.033049	-0.690610	14.200000	55.300000	69.900000	75.100000	84.100000	98.100000	1966.000000
ISM_PMI	52.255799	52.900000	5.787210	33.491801	-1.360541	2.224562	5.700000	32.400000	50.200000	52.900000	55.900000	61.400000	1966.000000

3.5.2. Distribution Visualization of Features

In addition to the descriptive statistics the distribution of the features as seen in the histograms (Figure 2) provide an insightful outlook of their symmetrical nature and the level of bias. This in turn influences the predictive performance of a model.

The distribution of the adjusted closing price is right-skewed, with high concentration toward the low range and a noticeable tail toward high prices. Thus, there is a wider dispersion on the lower prices, but the existence of a few relatively higher values reflects some volatility and probable outliers.

Considering the technical indicators histograms, RSI seems normally distributed around the middle, centering at 50-55, which is indicative of a relatively balanced market sentiment between overbought and oversold conditions. Stochastic Oscillator %K and %D are highly right skewed to the upside from 80 to 100, indicating that the index could have been in overbought territory for a large portion of the time. Williams %R also depicts right-skewed distribution, since many values lie near -20, which further reinforces the overbought signal.

This in general shows that momentum-based TA indicators signal bullish sentiments for the period. EMA follows a relatively similar shape to the distribution of the adjusted close price in its right skew. This makes intuitive sense since the EMA is a kind of trend indicator that smooths out the price over time.

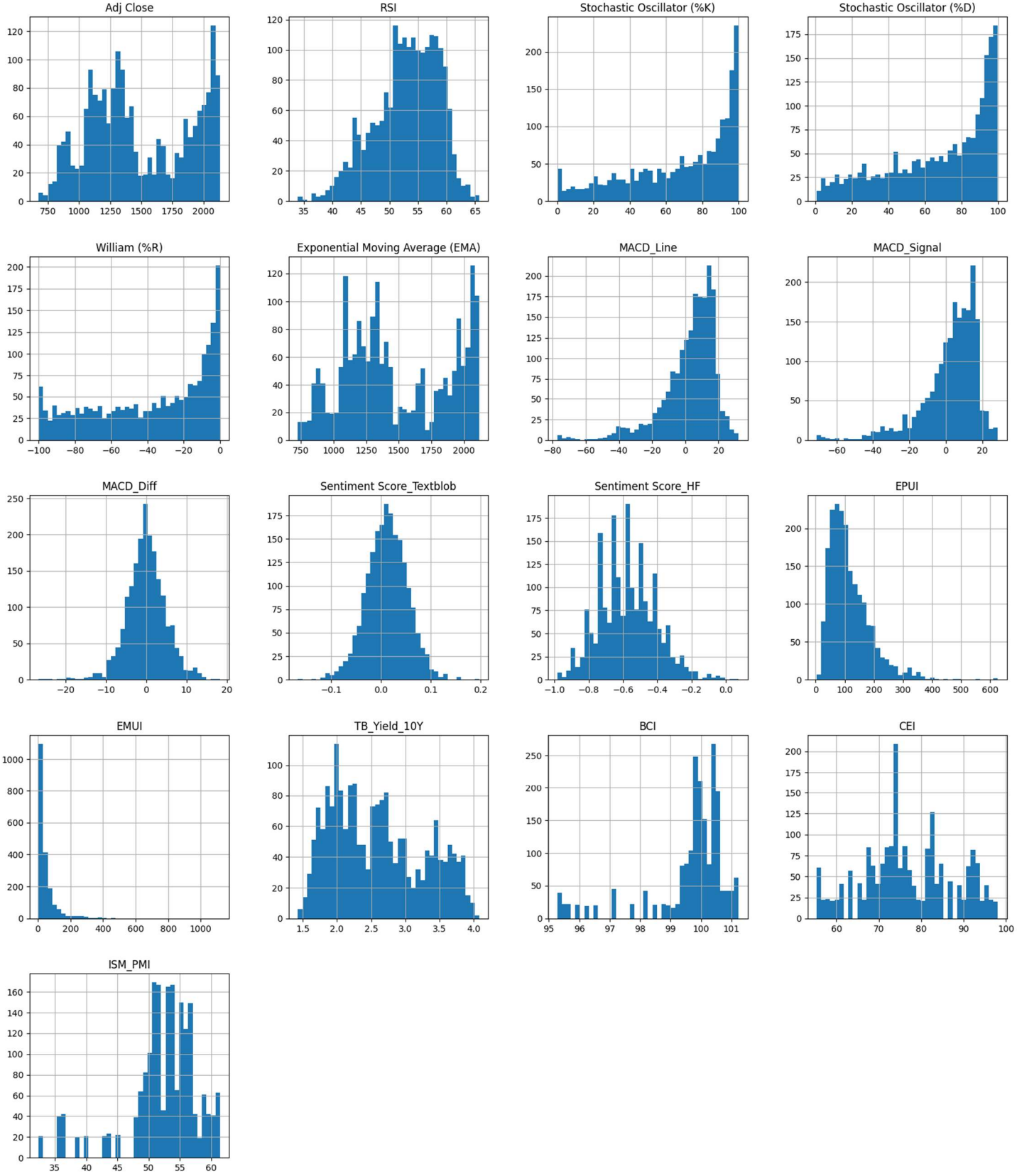
MACD Line and Signal Line most values are around zero, which suggests that on average, there is limited divergence from the trend, with large values occurring sometimes. MACD Diff is centered around zero, with a fairly symmetrical dispersion. This would mean that positive and negative divergences would appear with roughly equal frequency. This symmetry assists in picking out trend reversals, while a lack of bias toward positive or negative values can show sustained periods of momentum shifts.

With regards to the two sentiment scores, distribution is found to be near-normal and centered around zero, reflecting an overall neutral sentiment. This balance perhaps helps in flattening extreme values in sentiment data and hence being stable for predictions based on sentiments. The clustering is compact around zero, so very few large changes in sentiment occur, with most signals from sentiment being modest.

Lastly, the macroeconomic Indicators such as EPUI, exhibit positively skewed behavior, which represents low values with spikes upward. This could imply that uncertainty is usually low and occasionally peaks, perhaps during times when economies are under stress. EMUI also displays a similar but more extreme right skew, with many values clustered in the lower end. That could indicate that uncertainty is generally low, with perhaps some high peaks during newsworthy economic events. TB_Yield_10Y has a relatively uniform distribution with slight concentration in the middle ranges, suggesting a stable interest rate environment over the period. In addition, both BCI and CEI are somewhat right skewed with high peaks at certain values, which reflects more stability in business and consumer confidence, respectively, and also fewer extreme fluctuations. Finally, PMI concentrates around 50-55, which suggests that the economic activity is moderate and hovering around the expansion-contraction threshold of 50.

In general, the histogram visualization of the input features reveal that technical indicators skewed towards bullish signals, sentiment scores around neutrality, and macroeconomic indicators that are mostly stable but spiking. With this in mind, it would seem that the model would be better off focusing on the EMA and MACD for obtaining the trend information, while the other two sets of sentiment and macroeconomic indicators provide stability and, respectively, control the external factors at play which influence the price. Skewed distribution in some of the indicators may need transformation to achieve better accuracy in the model.

Figure 2. Distributions



3.5.2.1. Scatter Plot Analysis

In Figure 3, the scatter plots of S&P 500 vs. each input feature, describe the relationship of the S&P 500 adjusted close price with each feature. EMA exhibits the strongest positive correlation that exists between the S&P 500 and any input feature, with a near-linear relationship. This relationship confirms the fact that EMA moves in line with the price trend.

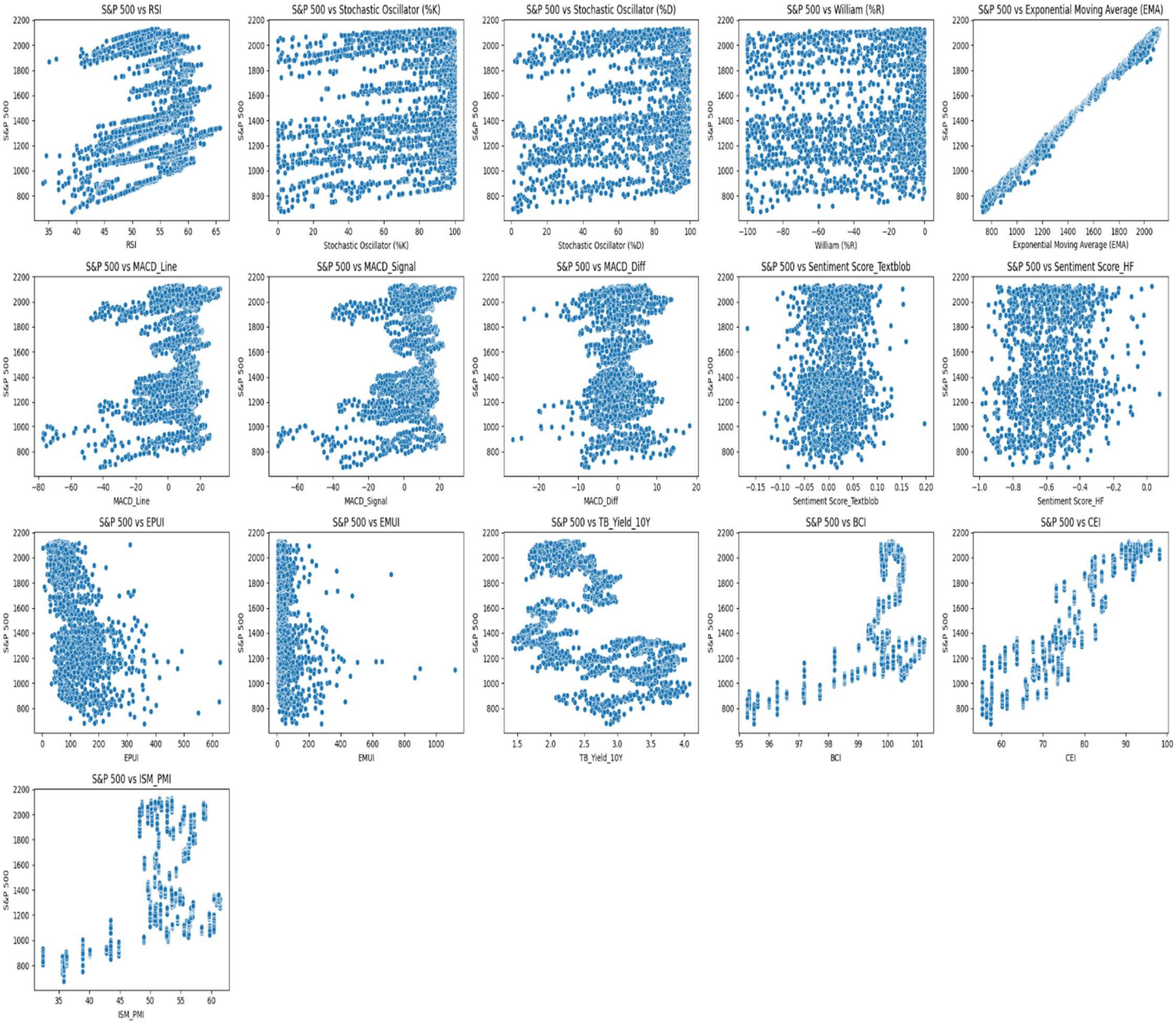
For technical indicators RSI, Stochastic Oscillator, MACD, the plots from these indicators are slightly grouped together, but no definite linear relationship is depicted with the S&P 500. These are normal, since the type of these features capture momentum and trend reversals, rather than definite linear correlations.

Given the neutral nature of sentiment scores from Textblob and HF distributions, both have a rather weak and dispersed relation with the S&P 500. That does not necessarily mean that they will not capture fluctuations in market sentiment, which would be useful in some nonlinear model.

Lastly, the plots of EPUI, EMUI, and TB_Yield_10Y are scattered with no signs of a linear relationship. Conversely, the plots regarding CEI and BCI exhibit a more established relationship with the S&P 500, with the former showing a positive inclining trend with the index. These may be good indicators to capture the long-run influence that the economy has on this market.

Overall, it appears that EMA and CEI have more relation with the adjusted close price of the S&P 500, whereas the technical indicators and the sentiment scores are not as linearly related. Also, given the skewed distributions and given some features with weak linear relationships, GB or NNs will probably help in capturing interactions with complex, non-linear dependencies.

Figure 3. Scatter Plots



3.5.2.2. Feature Correlation Matrix

Correlation matrix (Figure 4) provides an important insight into the relationship between the adjusted close price and all the other features in the dataset. The main key correlations with the Adjusted Close Price are found mainly with EMA, MACD Line, Signal, and Diff and various lag variables. EMA's correlation is very strong at 0.99 with the Adjusted Close price, indicating a strong alignment with the trend of S&P 500 prices. Therefore, EMA would serve as a major indicator for price movement prediction.

MACD Line, Signal, and Diff indicators show a moderate to high positive correlation with the adjusted close price, ranging from approximately 0.75 to 0.79.

Therefore, they strongly relate to timeframes when price trends alter their momentum. Lagged values of the technical indicators, such as RSI_lag10 and EMA_lag10, remain somewhat correlated to the adjusted close price but, in general, tend to be weaker than the current values. This would show that recent trends are more predictive than older ones.

With regards to other technical indicators, RSI is correlated at a moderate positive level of about 0.22, indicating that this might reflect overbought and oversold conditions, but that it is less directly related to the price than the other measures while, Stochastic Oscillators (%K and %D) and William (%R) have low correlations with the adjusted close price. It would, therefore, appear that they reflect very short-term overbought and oversold signals rather than longer-term trends.

Considering macroeconomic indicators, BCI and CEI are moderately correlated to the adjusted close price, at about 0.43 for BCI and 0.29 for CEI respectively. These indices indicate economic confidence and might be useful for making predictions on broad market conditions. EPUI and Economic Media Uncertainty Index (EMUI) seem that only small correlations are given with the adjusted close price, which may indicate that they can give signals only for marginal predictability in the movement of prices.

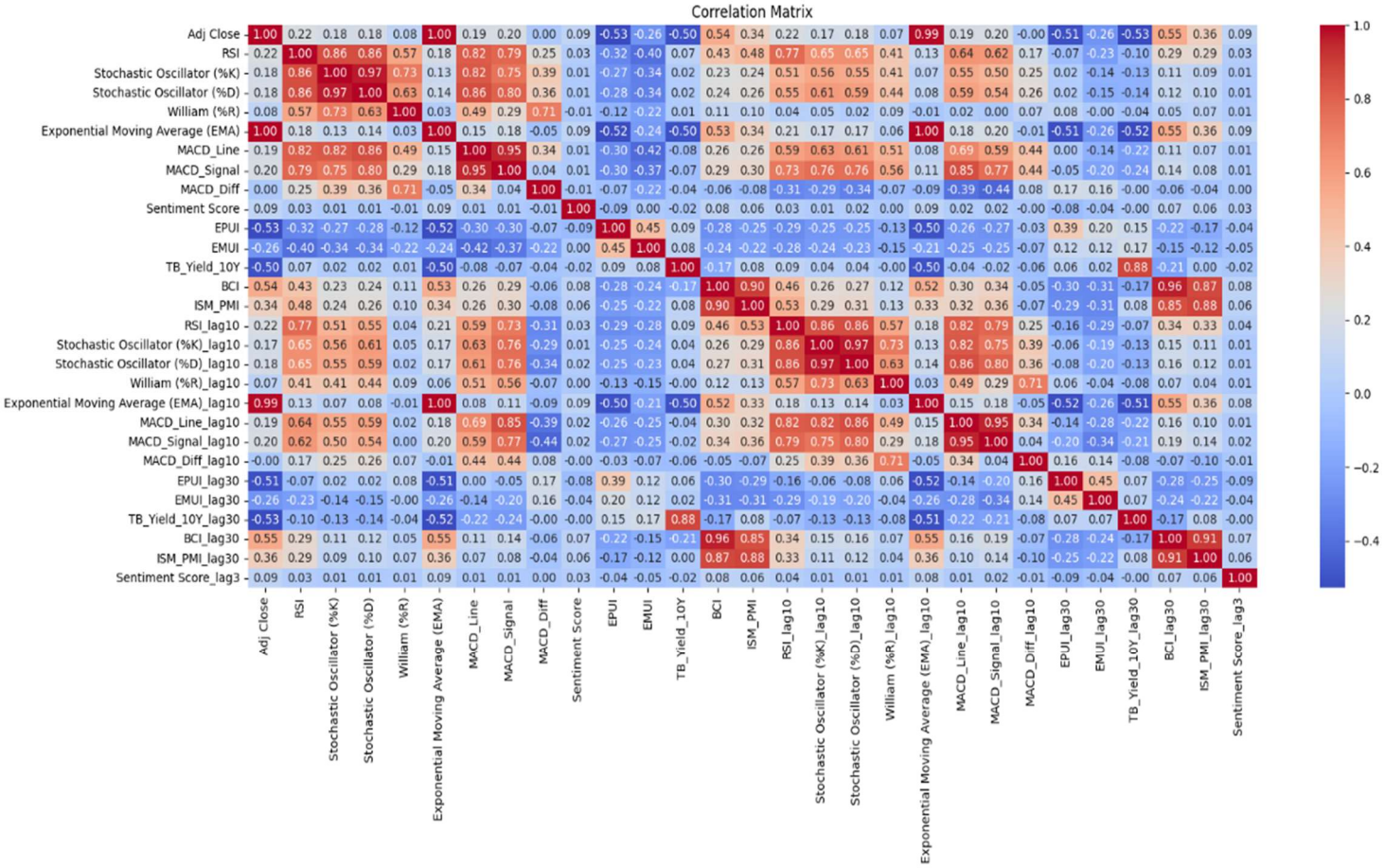
Similarly, both sentiment scores are also very close to zero in correlation with the adjusted close price, suggesting that daily sentiment does not have a significant direct effect on prices within this dataset. This also supports the scattered nature of the points within the scattered plots.

Lastly, any remaining lagged macroeconomic and sentiment indicators are even less correlated with the adjusted close price when using 30-day lags. This would suggest that the immediate economic sentiment is not as relevant as the most recent technical data when it comes to the forecasting of the daily movements in prices.

Overall, EMA and MACD are the most predictive features of S&P 500 adjusted close price, while macroeconomic indicators provide additional information, mostly on long-term tendencies. At the same time, it can be seen that sentiment scores and lagged macroeconomic indicators show very poor correlations and, thus, very low predictability in such context.

Taking these correlations into account, technical indicators provide more reasonable inputs to be employed in the models to follow leaving the use of macroeconomic variables as secondary input features to feed the model. Given also the evidence of multicollinearity between several technical indicators, especially among EMA and MACD components, and their lags, it is confirmed that some of these features might be redundant and would affect the stability in regression models. In this regard, Recursive Feature Elimination (RFE) is applied in order to test the effect of the most important features.

Figure 4. Correlation Matrix



4. Methodology

Data creation and preprocessing are important steps for financial forecasting. An efficient database requires that all relevant features, including technical indicators, macroeconomic data, and sentiment scores, are collected and placed in some form of analytical format. During preprocessing, the quality of the data is further enhanced by standardizing the formats, dealing with missing values, and matching data frequencies to ensure a consistent and accurate dataset on which models can efficiently be trained.

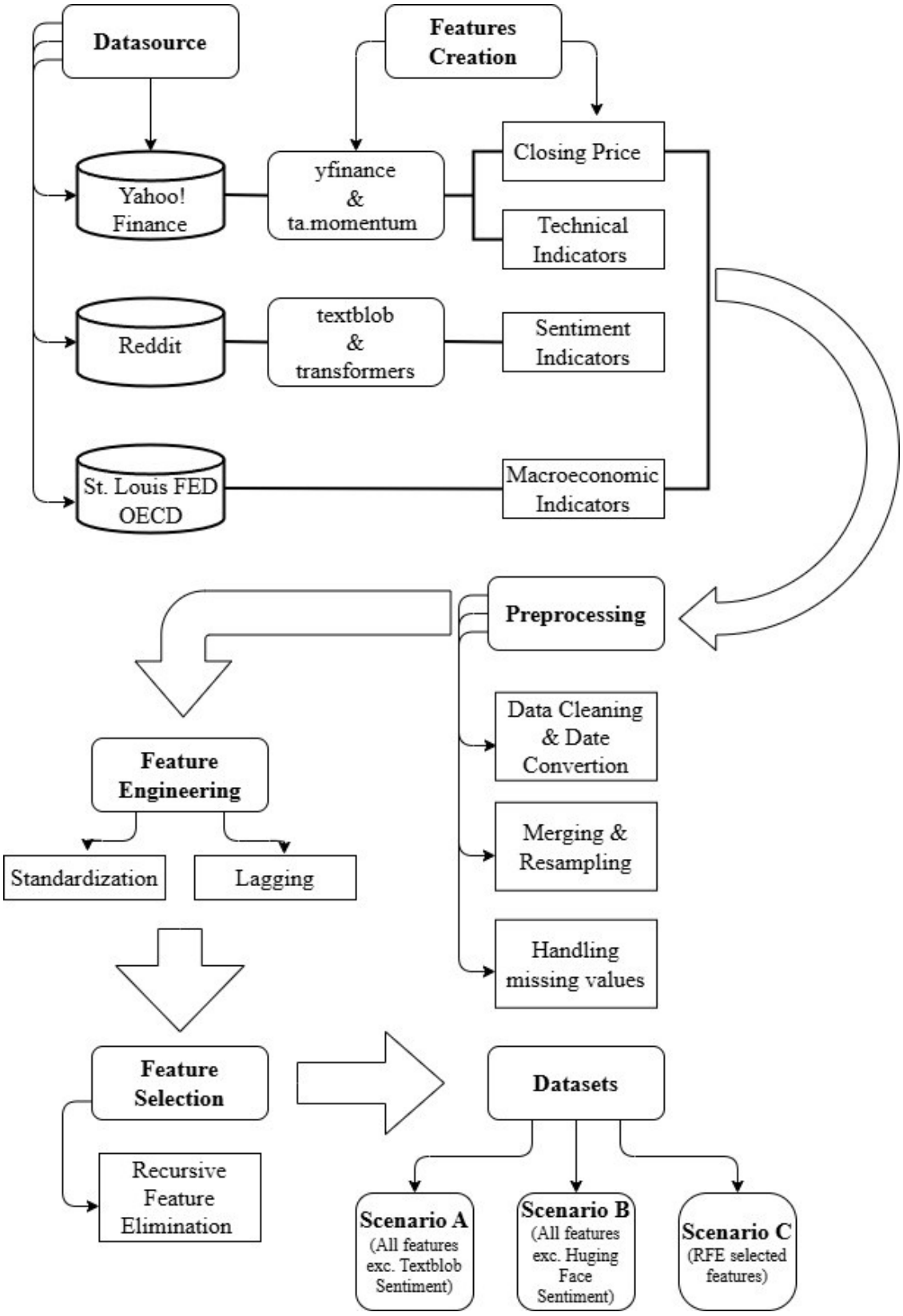
The financial dataset of this research was built on S&P 500 data, calculating technical indicators, while sentiment score was derived from news headlines. Most of the preprocessing steps included resampling, filling monthly indicators with a forward fill approach, and applying lagged values with respect to capturing temporal dependencies. Feature engineering and selection have also been applied in order for the model to be more accurate, but with lower computational complexity. The visualization of this architecture is found in Figure 5.

Accordingly, several ML models, namely LR, RF, GB, XGB Regressor, and a Multi-Layer Perceptron, were utilized to make predictions on S&P 500 index prices. Each of these models provides individual strengths in analyzing these factors for appropriate understanding of stock market movements.

4.1. Data Creation and Preparation

Historical data from the S&P 500 index were downloaded by employing the yfinance library. The initial dataset included fundamental columns, such as Date, Open, High, Low, Close, Adj Close, and Volume over the period of analysis. The yfinance library is regarded as a respected source in the aggregation of stock market information as its main source of information is Yahoo Finance. Hence, the use of yfinance library indicates a commitment to data reliability and efficiency, as the extraction would be through automated means, with no need for manual intervention. For this reason, the study would be guaranteed to operate on updated and complete historical data, an essential ingredient for accurate forecasting models.

Figure 5. Workflow Chart (Dataset Creation)



Technical indicators provide major tools in financial analysis that one employs to quantify the stock price trends, momentum, and volatility. For the calculation of the indicators used in this study the ta.momentum library was employed (Appendix B - Table 1). This library is ideal for the creation of such indicators, as it provides useful methods for capturing market dynamics. The library ta.momentum automates indicator calculations in a consistent manner that is error-free from complex indicators. This also enables the addition of technical indicators in the data, which contributes to identifying short-term patterns that may have a significant effect on the model's performance.

Investor sentiment has nowadays become a key determinant of market fluctuations. Sentiment analysis techniques help to quantify the positive, neutral, or negative tone of news headlines that may influence stock prices. In this work, two methods have been implemented for calculating sentiment scores: TextBlob [67] and pre-trained DistilBERT-base-uncased model [66] from Hugging Face's (HF) transformer library (Appendix B – Table 2). Advanced lexicon approaches and transformer-based pipelines like TextBlob and HF's transformer library surpass other lexicon-based approaches like VADER due to their deep linguistic and contextual analysis capabilities, when they are applied to lengthy or complex world news headlines. VADER and similar approaches perform better when they are applied to short texts found in tweets or short reviews as they are capable to emphasize in capturing nuances like slang, emoticons, and intensifiers. As suggested by Hutto and Gilbert (2014), VADER is suitable for short texts sentiment analysis as its ability to analyze longer, nuanced textual data is constrained.

Sentiment scores have been computed daily for 25 headlines related to world news. TextBlob is a lightweight library that uses the Natural Language Toolkit (NLTK) for sentiment analysis and is suitable at handling formal and structured language. Its methodology lays upon the idea of assigning for each text a polarity score, representing sentiment on a scale from -1 (negative) to 1 (positive) [67]. The average of the sentiment score of the headlines on each day has been computed from 25 daily headlines and gives a consolidated measure of the sentiment on a given day. Unlike prior studies, Cristescu et al (2023) suggest that despite its simplicity, Textzblob library offers advantages

against other libraries in processing and comparing the sentiments of news titles and descriptions.

Additional to Textblob library the pre-trained DistilBERT-base-uncased model from HF transformers library was also employed to secure the effectiveness provided by Textblob and even more to enhance the results. This library gave access to a great number of pre-trained language models, enabling the comprehension of the meaning of news headlines. In contrast to TextBlob, that rely on a naïve polarity score, transformers-based models leverage deep learning models like BERT, RoBERTa, or DistilBERT to capture context, semantics, and nuances in text, making them ideal for complex texts, such as headlines. The daily average of the sentiment score was calculated in the same manner as in the case of TextBlob. Headlines with no string values were assigned with a 0-value indicating a rather neutral sentiment.

4.2. Data Preprocessing

In this subsection, the basic steps of preprocessing are presented for the dataset to be structured enough for engineering and feature selection techniques.

4.2.1. Cleaning and Converting the Data

Preprocessing of the data was highly necessary in order to be able to have consistent and easily usable datasets. This first step was the conversion of the 'Date' column to 'YYYY-MM-DD' format. Next the renaming of column names was followed for better readability and to remove ambiguity. The columns 'Open', 'High', 'Low', 'Close', and 'Volume' were dropped as only adjusted closing prices with technical indicators and sentiments scores were used in the analysis.

4.2.2. Data Merging and Resampling

Merging data of different frequencies creates peculiar problems, especially when one merges daily and monthly data sets. Since technical indicators, sentiment scores and most of the macroeconomic indicators were available at a daily frequency, they were combined directly into the daily-adjusted closing price data of the S&P 500.

For some macroeconomic indicators, like the BCI, Consumer Confidence Index (CEI), and Purchasing Managers' Index (PMI), that were available monthly, in order to incorporate them into a daily frequency dataset, forward filling was used where the last known monthly value was propagated forward to fill in missing values for the month.

This assumption is based on the idea that month-over-month indicators have fairly static effects during the month. It thus becomes suitable to use a forward-filling approach in capturing the daily stock price effect.

4.2.3. Handling Missing Values

Missing values can distort model accuracy and result in biased predictions when not properly treated. Columns that contained data errors, such as dots (.), were replaced with NaN to normalize these missing entries. Then, for these entries and for any remaining gaps, a rolling mean approach was implemented. This technique takes the average of the values, one before the others within a specified window, for each missing value as its replacement. Such functionality removes spikes in this data and provides a much smoother time-series on which to train a model.

Finally, for consistency purposes, the dataset was trimmed down to the period starting from August 8, 2008, up to May 31, 2016, considering availability for both market and sentiment data.

4.3. Feature Engineering

Feature engineering is the process of creating new input features to improve the learning and performance of a model. In this study, feature engineering was performed by lagging and standardizing key indicators to capture the temporal dependencies and avoid data leakage.

Lagging features is the process of bringing the values of the past forward to make new predictive inputs. This step helps the model identify temporal relationships and ensures that future data is not mistakenly used in predictions, preventing data leakage. In this study technical indicators were lagged for 1 day to show how their past values affect the immediate future market movements. Similarly, daily macroeconomic indicators were lagged by one day to ensure that only prior day data provides impact to the predictions preserving causal integrity. For monthly indicators, start-of-month and end-of-month values were resampled to a daily frequency. Each such value was then shifted by one month to account for its lagged effect. Finally, since news stories move markets very quickly, the sentiment scores were also lagged by 1 day-that would give the models the context of sentiment on any previous day.

StandardScaler method was used to standardize the features before model training as an important step toward best performance. This method transforms each feature by removing its mean and scaling it to unit variance, according to the following formula:

$$z = \frac{x - \mu}{\sigma}$$

where x is the original feature value, μ is the mean, and σ σ is the standard deviation computed from the training data. This scaler is then fitted directly on the training dataset to avoid data leakage and next is applied on both training and test datasets. Standard scaling of data is important in most ML models that are sensitive to the magnitude of features, especially those using gradient-based optimization methods. It allows most ML algorithms to operate optimally by making sure all features are given equal importance in the process of training, as it stabilizes numeric scale or range, increases the speed of convergence when training models, and ensures that each of the features is weighted equally in the learning process, particularly in models like LR, where unscaled features can distort coefficient estimates. Standardizing the data in this study ensures that input features are better utilized by all the models, therefore enhancing overall performance reliability.

4.4. Feature Selection

After generating and lagging features, the number of input features significantly increased. For this reason, RFE was selected to identify the most predictive features that would potentially make the models more efficient with increased performance. RFE is the technique that, with every iteration, takes away the least important features with regards to model performance, until it finds the best subset of features that can be utilized for stock price prediction.

Feature ranking using the base model in this study was done by applying this technique (Appendix B – Table 3). As seen in Table 4 - 1, the features up to rank 2 were tested by the models and compared to the results produced by the models without considering this technique. The aim was to reduce computational complexity and consequently avoid overfitting by removing uninformative or redundant features.

Table 4-1. Feature Selection (Ranking)

	Feature	Ranking
0	RSI_lag1	1
1	Stochastic Oscillator (%K)_lag1	1
2	Stochastic Oscillator (%D)_lag1	1
4	Exponential Moving Average (EMA)_lag1	1
5	MACD_Line_lag1	1
6	MACD_Signal_lag1	1
7	MACD_Diff_lag1	1
10	TB_Yield_10Y_lag1	1
14	Sentiment Score_Textblob_lag1	1
15	Sentiment Score_HF_lag1	1
11	BCI_lag1	2
13	ISM_PMI_lag1	3
12	CEI_lag1	4
3	William (%R)_lag1	5
8	EPUI_lag1	6
9	EMUI_lag1	7

4.5. Model Description

In the study, five ML models were employed for the prediction of the S&P 500 index price: LR, RF Regressor, GB Regressor, XGBoost Regressor, and MLP Regressor. To evaluate the impact of different feature combinations, the models were tested under three distinct scenarios:

(1) all lagged features incorporated including only the sentiment score variable derived from pre-trained DistilBERT (Sentiment Score_ pre-trained DistilBERT _lag1),

(2) all lagged features incorporated including only the sentiment score variable derived from TextBlob (Sentiment Score_Textblob_lag1),

(3) only the features selected through RFE were incorporated up to level 2.

The dataset was then split into 80% for training and 20% for testing to facilitate unbiased model evaluation and ensure the generalizability of the results (Figure 6).

The hyperparameters of these models were initially optimized using RandomizedSearchCV, a method that randomly samples candidates from the hyperparameter space [68]. Computationally cheap, this approach generally yields comparable results to GridSearchCV, which performs an exhaustive search over a user-specified hyperparameter space [69]. Whereas GridSearchCV will ensure the best configuration is found, doing so is at a far higher computational cost. These techniques are important, as they allow models to find the optimal performance of the model by systematically testing and choosing the best hyperparameters for the data and the given task. The performance of all models during the search was evaluated using the same performance metrics: MSE, MAE, and R^2. The purpose was to ensure that the tuning process was aligned with the goal of minimizing prediction error while maximizing explanatory power. In addition, checks for overfitting were also employed and additional tuning to the hyperparameters of the models was made for those instances where overfitting was found.

Feature importance analysis was also conducted for all models to interpret the contribution of individual features to the predictions. This step is particularly useful in understanding what really drives the S&P 500 index price and thus provides insight into which technical, macroeconomic, or sentiment-based variables had greater predictive power. For ensemble models, such as RF and GB, feature importance was derived from the impurity or loss reduction attributed to each feature. Other models utilized coefficients or SHAP values to interpret feature contributions. Rigorous hyperparameter tuning and careful feature importance analysis boded well for models optimized for predictive accuracy, but also interpretable-a methodological alignment with the dual goals of robust forecasting and actionable insights.

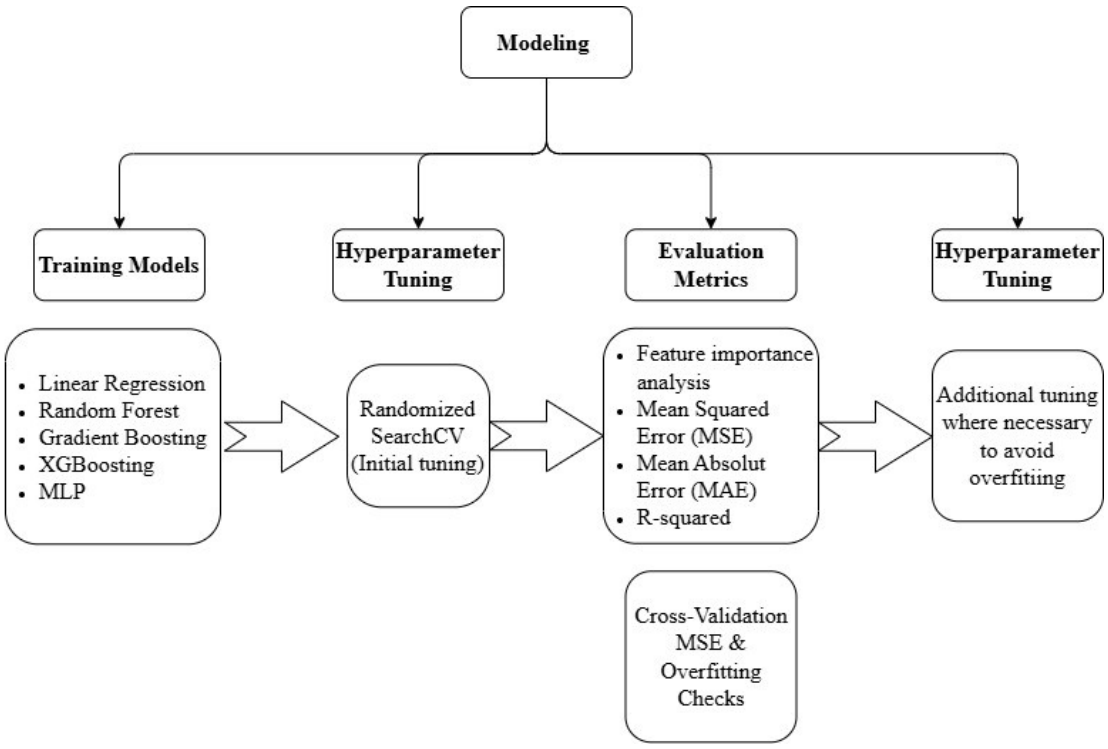
4.5.1. Linear Regression

LR is considered a traditional model in ML, which tries to capture the linear relationship between a set of input features and the target variable with an added bias term [70]. The simplicity and interpretability of the model makes it a good starting point for any predictive analysis [70]. Mathematically, this model is represented as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon$$

where y is the forecasted value, β_0 is the intercept, β_i is the coefficient for the i^{th} feature, x_i is the i^{th} feature, and ϵ is the error term [70].

Figure 6. Workflow (Modelling Architecture)



Its simplicity makes it possible for quick implementation and forms the basis for understanding the relationship between features and the target variable. In this study, its interpretability was especially helpful in assessing the predictive power of technical, macroeconomic, and sentiment-based features. The coefficients allowed for quantification of the strength and direction of each separate feature's influence on the price of the S&P 500 index. It also provided a baseline performance measure against the other models employed, since their outperformance signaled an indication that the relationship of the data is highly linear, and that features engineering and data preprocessing is in order.

In this work, the implementation of LR was to fit the model on the scaled training dataset using the LinearRegression() class from Scikit-learn. The StandardScaler was applied to make all features be on a comparable scale, because unscaled data results in biased coefficient estimates in models sensitive to the magnitude of the predictors. This is summarized in the following code (Appendix B).

4.5.2. Random Forest Regressor

RF Regressor is a robust ensemble learning model that combines multiple trees to improve the prediction performance and at the same time reduce overfitting [70]. Every tree in the forest is built on a random subset of data and features, using a method called bootstrap aggregation (bagging). This approach reduces the variance by averaging the prediction of each tree, thus resulting in a model that balances accuracy and generalization.

The choice of this model in the study is critical because of its ability to capture nonlinear relationships and interactions between features, often prevalent in financial datasets[70]. Compared to other models, this makes no strong parametric assumptions, enabling the methodology to be more adaptive to high-dimensional and complex data. This flexibility is especially valuable given the diverse nature of the input features since they all vary significantly.

In addition, the model's resistance against overfitting allows for robust performance against the presence of noisy or less relevant predictors. By aggregating predictions from multiple trees, it mitigates potential overfitting to particular patterns in the training data which is common in financial modeling [70].

Hyperparameter tuning for the RF Regressor was carried out using RandomizedSearchCV (Table 4-2). The choice of the parameter grid was made with considerations for model complexity, generalization, and computational efficiency. Key parameters included:

Table 4-2. Random Forest Hyperparameters Tuning (Random Search)

Hyperparameter	Tested Values	Best Value	Description
`n_estimators`	[100, 200, 300]	300	Number of trees in the forest. Higher values improve robustness and reduce variance but increase training time.
`max_depth`	[None, 10, 20, 30]	30	Maximum depth of the trees. Prevents overfitting by restricting tree growth.
`min_samples_split`	[2, 5, 10]	2	Minimum number of samples required to split an internal node. Controls tree growth and balance between bias and variance.
`min_samples_leaf`	[1, 2, 4]	1	Minimum number of samples required to be at a leaf node. Ensures small nodes are not excessively penalized.
`max_features`	['auto', 'sqrt', 'log2']	'sqrt'	Number of features considered when splitting a node. 'sqrt' adds randomness and reduces overfitting.

Having tested the model under the hyperparameters provided by the RandomizedSearchCV resulted in overfitting results, so additional tuning was made and the hyperparameters of Table 4-3 were eventually selected for this study.

Table 4-3. Random Forest Hyperparameters Tuning

Hyperparameter	Value	Description
`max_depth`	15	The maximum depth of each tree, limiting its growth to prevent overfitting.
`max_features`	'sqrt'	The number of features considered for the best split is the square root of the total number of features.
`min_samples_leaf`	5	The minimum number of samples required to be at a leaf node, controlling overfitting.
`min_samples_split`	10	The minimum number of samples required to split an internal node.
`n_estimators`	200	The total number of trees in the forest.
`random_state`	42	Ensures reproducibility by fixing the random seed for model building.

These hyperparameters allow the RF Regressor to effectively learn the non-linear relationships in the data while avoiding overfitting. Hence, this combination was balanced enough to be both reasonably predictive but also not overfitting for a good S&P 500 index price forecast.

4.5.3. Gradient Boosting Regressor

GB Regressor is an ensemble learning model that builds sequentially upon the weak learners (decision trees) in order to come up with a strong predictor. Each consecutive tree is trained to minimize the residual error of the previous tree using the gradient descent approach in order to optimize its loss function. This iterative approach ensures that the model focuses on instances that are difficult to predict, effectively shrinking bias and increasing accuracy [70].

Since nonlinear relationships and feature interaction are usually present in financial data, employing such a model creates additional value in this study. Unlike simpler models, it uses additive corrections and as a result it is highly adaptive to the complexities of the dataset. This adaptability is very useful considering the diversity of the feature set that entails technical indicators, macroeconomic variables, and sentiment scores [70].

Another strength of GB is that, with proper tuning of the parameters, it can potentially be less biased and have a lower variance, hence generally being more resistant to overfitting. While this model has a greater propensity toward overfitting than the RF, it can be regularized with different modifications in learning rate, limiting tree depth, and subsampling in order to minimize such risks [70].

The hyperparameter optimization for GB Regressor is done using RandomizedSearchCV with 5-fold cross-validation to make the evaluation robust (Table 4-4). The choice for the parameter grid is a trade-off between model complexity and computational efficiency:

Table 4-4. Gradient Boosting Hyperparameters Tuning (Random Search)

Hyperparameter	Tested Values	Best Value	Description
`n_estimators`	[100, 200, 300]	200	Number of boosting stages (trees). Balances performance and training time.
`learning_rate`	[0.01, 0.05, 0.1]	0.05	Shrinks the contribution of each tree to prevent overfitting.
`max_depth`	[3, 5, 7]	7	Limits the depth of trees to control model complexity.
`min_samples_split`	[2, 5, 10]	5	Minimum number of samples required to split a node.
`min_samples_leaf`	[1, 2, 4]	1	Minimum number of samples required at a leaf node.
`subsample`	[0.8, 1.0]	1.0	Fraction of samples used for training each tree.
`max_features`	['auto', 'sqrt', 'log2']	'sqrt'	Number of features considered for the best split.

Overfitting also existed after employing the model given the hyperparameters tuning suggested by RandomizedSearchCV and therefore further tuning was made to secure minimum overfitting after training the model. The most appropriate tuning was found as per Table 4-5.

Table 4-5. Gradient Boosting Hyperparameters Tuning

Hyperparameter	Value	Description
`learning_rate`	0.005	Shrinks the contribution of each tree, controlling the model's step size during optimization.
`max_depth`	3	The maximum depth of each tree, limiting its complexity to prevent overfitting.
`max_features`	0.8	The fraction of features considered for the best split, promoting generalization.
`min_samples_leaf`	4	The minimum number of samples required to be at a leaf node, helping control overfitting.
`min_samples_split`	10	The minimum number of samples required to split an internal node.
`n_estimators`	1000	The total number of boosting stages to perform.
`subsample`	0.8	The fraction of samples used for fitting each tree, adding randomness and reducing overfitting.
`random_state`	42	Ensures reproducibility by fixing the random seed for model training.
`n_iter_no_change`	10	Stops training early if no improvement in the validation score is observed for a given number of iterations.

Given this hyperparameters tuning, complex patterns in the financial dataset are captured effectively by the GB Regressor. The sequential learning structure ensures a

gradual reduction of the bias, while the use of regularization parameters keeps the risk of overfitting low. Such characteristics make the GB Regressor a very apt candidate for the task of forecasting the S&P 500 index price, which needs to be highly generalized.

4.5.4. XGB Regressor

Extreme Gradient Boosting (XGB Regressor) can be considered an advanced ensemble model, which works on major principles of GB to enhance efficiency and performance of the results. It trains the decision trees sequentially to minimize the residual errors of the previous iterations using methods from gradient descent optimization and boosting. Thus, combining these methods, the model is more efficient in reducing bias and controlling variance [71].

The key advantage of this model is that it is bult to include regularization, parallelized learning, and optimized tree pruning enhancing its performance. The regularization technique is vital for the prevention of overfitting even on high-dimensional data. By controlling the learning rate, limiting the depth of trees, and setting minimum thresholds for the child weight of a tree node, the model enforces generalization without losing any valuable accuracy. Besides, it introduces randomness in subsampling rows and columns while training, which makes XGBoost more generalized to unseen data [71].

The implementation of the model begins with some baseline prediction which is usually the mean of the dependent variable. Iteratively, it computes residual errors, trains decision trees to minimize those residuals, and updates predictions by summing up weighted outputs of newly trained trees. Then, regularization is applied to adjust these weights during training and maintain generalization in good levels avoiding overfitting at the same time. The accumulation of predictions from all trees yields the final output [71].

In this study, XGB Regressor has been selected due to its ability to capture complex, nonlinear relationships within financial data. Hyperparameter tuning was initially applied using RandomizedSearchCV with 5-fold cross-validation to ensure a robust evaluation as found in the Table 4-6, but after model implementation, overfitting was found in the results. To overcome this issue, further tuning was applied as per the Table 4-7 below:

Table 4-6. XGBoosting Regressor Hyperparameters Tuning (Random Search)

Hyperparameter	Tested Values	Best Value	Description
`n_estimators`	[100, 200, 300]	200	Number of boosting rounds. Higher values may improve performance but increase training time.
`max_depth`	[3, 5, 7, 10]	7	Maximum depth of a tree. Controls model complexity and helps prevent overfitting.
`learning_rate`	[0.01, 0.05, 0.1, 0.2]	0.05	Step size shrinkage to prevent overfitting. Lower values ensure smoother learning but may require more rounds.
`subsample`	[0.6, 0.8, 1.0]	0.6	Fraction of samples used for each boosting round. Helps reduce overfitting.
`colsample_bytree`	[0.6, 0.8, 1.0]	1.0	Fraction of features used per tree. Higher values increase feature usage, lower values promote regularization.
`min_child_weight`	[1, 3, 5]	1	Minimum sum of instance weights needed for a child. Larger values prevent overfitting by creating more conservative splits.

Table 4-7. XGBoosting Regressor Hyperparameters Tuning

Hyperparameter	Value	Description
`subsample`	0.8	The fraction of samples used for fitting each tree, adding randomness to improve generalization.
`n_estimators`	300	The total number of boosting stages (trees) to perform.
`min_child_weight`	5	The minimum sum of instance weights needed in a child node, controlling overfitting.
`max_depth`	5	The maximum depth of each tree, limiting model complexity.
`learning_rate`	0.01	Shrinks the contribution of each tree, controlling the optimization step size.
`colsample_bytree`	0.8	The fraction of features considered for each split, introducing randomness.
`gamma`	0.1	Minimum loss reduction required to make a split, adding regularization for tree structure.
`reg_lambda`	1	L2 regularization term on weights, controlling overfitting.
`reg_alpha`	0.5	L1 regularization term on weights, promoting sparsity in the model.
`random_state`	42	Ensures reproducibility by fixing the random seed for model training.

4.5.5. MLP Regressor

MLP Regressor is a NN based regression model that dynamically captures complex and non-linear relationships within data. Compared to other tree-based methods that partition data by splitting the features, this model processes input data through interconnected layers of neurons. Each neuron performs weighted transformations

followed by nonlinear activation functions that allow the model to approximate complex patterns and dependencies [70].

In addition, this model is also characterized by its flexibility to model various data distributions especially for datasets with non-linear relationships between features, just like the case of this study since technical indicators, sentiment scores, and macroeconomic variables interact in unpredictable ways. It also incorporates regularization techniques, such as using L2 penalties, controlled through an alpha parameter, that can be helpful against overfitting by constraining model complexity. Lastly, the inclusion of activation functions like ReLU or tanh enables the model to adapt to varying data structures, enhancing its generalization [70].

Compared to the other models in this study, MLP Regressor provides a NN architecture that utilizes a layer of neurons directly to learn mappings between input and output instead of depending on decision trees to find patterns. This unique approach makes it especially powerful for scenarios in which the relationships between features are too complex or too subtle to be captured by traditional methods [70].

During the implementation of the model, the input space consists of scaled features. These inputs are fed through one or more hidden layers, where each neuron computes a weighted sum of inputs followed by a nonlinear activation. The output layer produces a prediction for the regression problem in study, and the model repeatedly updates its weights via a process termed backpropagation that minimizes some predefined loss function. Regularization is used during training to achieve better generalization, and techniques like adjusting the learning rate also optimize the performance of the model [70].

As in previous models, hyperparameter tuning was initially applied using RandomizedSearchCV with 5-fold cross-validation to ensure a robust evaluation as found in Table 4-8. Results were also checked for overfitting with not significant evidence of important existence.

Table 4-8. MLP Regressor Hyperparameters Tuning (Random Search)

Hyperparameter	Tested Values	Best Value	Description
`hidden_layer_sizes`	[(50, 50), (100,), (50, 50, 50)]	(50, 50, 50)	Defines the architecture of the neural network. More layers and nodes allow capturing complex patterns.
`activation`	['relu', 'tanh']	'relu'	Activation function for the hidden layers. 'relu' helps with non-linear patterns and prevents vanishing gradients.
`solver`	['adam']	'adam'	Optimization algorithm. 'adam' is efficient for large datasets and noisy gradients.
`alpha`	[0.0001, 0.001]	0.001	Regularization parameter to prevent overfitting by penalizing large weights.
`learning_rate_init`	[0.001, 0.01]	0.01	Initial learning rate for weight updates. Affects the convergence speed of the model.

The MLP Regressor is particularly suitable for financial modeling, since the technique operates admirably with high-dimensional data and allows for complex relationships between various predictors. The flexibility here complements the structured approach of tree-based models, providing a useful alternative to capture intricate patterns in financial datasets. This makes the method an integral component in building robust predictive models within the domain [42].

5. Results

In this section the results per model are presented after evaluating their performance using MSE, MAE and R^2. In addition, feature importance results per model are also presented to determine the most valuable features contributing to the predictive power of the models. Lastly, a discussion follows regarding model overfitting evaluation, given additional metrics and learning curves.

5.1. Linear Regression

Following the implementation of the LR model, MSE, MAE and R^2 were calculated to evaluate its performance for each one of the selected scenarios.

As per Table 5-1, for the scenarios where feature selection was not applied, the model gave almost identical results regardless of the sentiment technique. Both MSE and MAE were almost identical, and with the high value of $R^{2,}$ the model indicated that data are very well-fitted, providing strong predictive power, regardless of the sentiment score employed. With feature selection applied, MSE slightly increased to 375.19 and MAE to 13.88, whereas R^2 remained high at 0.9977, demonstrating that while the model's complexity was decreased when the input features were reduced to only the most important ones, there was not any significant loss in predictive capability.

Table 5-1. Linear Regression (Results)

Scenarios			MSE	MAE	R^2
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	370,05	13,79	0,9977
b	No-Feature Selection	Sentiment Score_Textblob	371,03	13,84	0,9977
c	Feature Selection	Up to rank 2	375,19	13,88	0,9977

After employing feature importance analysis (Table 5-2) to interpret the contribution of each feature to model predictions, it was found that EMA_lag1 was the most influential feature for models without feature selection, with a value at 398. Stochastic Oscillator (%K)_lag1 and MACD_Diff_lag1, were also dominant in driving predictions, while both sentiment features had negligible effect, indicating that quantitative price-based patterns were more critical to index movement prediction.

Using feature selection, EMA_lag1 remained dominant, with its importance increasing slightly to 399.59, while Stochastic Oscillator (%K) _lag1 and MACD_Diff_lag1 were still significant contributors, and sentiment-based features retained their marginal influence. This reduction in feature set size served well to emphasize the strong predictors, thus reassuring that this model concentrates on the important drivers without sacrificing accuracy.

Table 5-2. Linear Regression (Feature Importance)

	Scenarios		
Feature	**A**	**B**	**C**
RSI_lag1	4,15	4,10	5,54
Stochastic Oscillator (%K)_lag1	28,71	28,72	30,68
Stochastic Oscillator (%D)_lag1	-25,40	-25,38	-27,19
William (%R)_lag1	1,65	1,68	#N/A
Exponential Moving Average (EMA)_lag1	397,99	398,02	399,59
MACD_Line_lag1	4,48	4,47	4,64
MACD_Signal_lag1	0,48	0,48	0,40
MACD_Diff_lag1	12,96	12,93	13,73
EPUI_lag1	-1,67	-1,63	#N/A
EMUI_lag1	-0,09	-0,14	#N/A
TB_Yield_10Y_lag1	-2,96	-2,96	-2,10
BCI_lag1	-2,51	-2,42	-0,22
CEI_lag1	0,87	0,86	#N/A
ISM_PMI_lag1	2,74	2,72	#N/A
Sentiment Score_ pre-trained DistilBERT _lag1	0,65	#N/A	#N/A
Sentiment Score_Textblob_lag1	#N/A	0,45	0,52

Residual analysis takes a closer look at the difference between actual and estimated values, and therefore provides more detailed insights into model performance, as found in the following Figures (7, 8, 9). Models without feature selection exhibited similar residuals between both sentiment sources. Most values fell between ± 85. While those with values at -4.93 and -4.96, presented good short-term predictions, residuals with greater values, such as -85.40 and -84.72, presented challenges during extreme market changes. When feature selection was applied, residual patterns remained similar, with a few outliers larger in some cases, like -87.76. The residual behavior, however,

remained quite similar to models without feature selection, showing feature selection does not worsen much the predictive accuracy of the models. All these point to the model's robustness and ability to maintain performance under varying configurations.

Figure 7. Linear Regression Results (Scenario A)

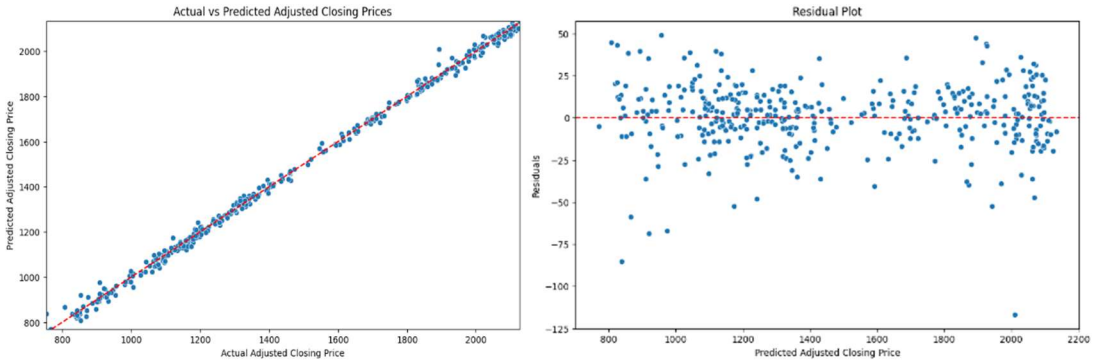


Figure 8. Linear Regression Results (Scenario B)

Figure 9. Linear Regression Results (Scenario C)

The various evaluation metrics and analyses of feature importance bring out the resilience and effectiveness of the model. The minor rise in both MSE and MAE after feature selection demonstrates that the model can focus on the most relevant predictors without risking its performance. The dominance of technical indicators indicates their strong predictive power in capturing price trends and momentum, making them reliable

inputs for the prediction of the index price. On the other hand, sentiment scores suggest that they are not capable enough to support the prediction of market movements. Overall, high R^2 indicates that the model explains almost all the variance in the index and it is reliable and precise in conditions of stable markets.

5.2. Random Forrest Regressor

Following the LR results, the RF Regressor model was also trained in order enhance predictive power. Initial results indicated slight differences across scenarios with high R^2 and relatively higher errors compared to other models. As seen in Table 5-3, without feature selection, scenario A returned an MSE of 533,58, an MAE of 16,98, and an R^2 value of 0.9969, while scenario B yielded a slightly higher MSE of 540,3 and MAE of 17,27, though R^2 value remained high at 0.9968. With the refinement of features, scenario C resulted in an MSE marginally higher at 556,94, and decreased MAE of 17,15, indicating improved prediction accuracy. R^2 value declined slightly at 0.9967 but without affecting the robustness of the model. The obtained results in this instance suggest minimum usefulness of feature selection, since limiting features only to those which provide maximal insights, had no significant impact to the overall model performance.

Table 5-3. Random Forrest (Results)

Scenarios			**MSE**	**MAE**	**R^2**
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	533,58	16,98	0,9969
b	No-Feature Selection	Sentiment Score_Textblob	540,30	17,27	0,9968
c	Feature Selection	Up to level 2	556,94	17,15	0,9967

In addition to the initial evaluation, feature importance analysis was employed at this stage, providing further insights into the key drivers of the model, as per Table 5-4. In all scenarios, (EMA)_lag1 was found to be the most influential feature as it accounted for almost 37% of the importance in scenario A & B, while in scenario C this value increased to 58%. Less important contribution was found in CEI_lag1 and BCI_lag1, but still relevant for the prediction task, while sentiment-based features proved to be the least important. Similarly, BCI_lag1 and TB_Yield_10Y_lag1 were found to have

moderate contribution in scenario C, with the least impactful features to be the sentiment-based features.

Table 5-4. Random Forrest (Feature Importance)

	Scenarios		
Feature	**A**	**B**	**C**
Exponential Moving Average (EMA)_lag1	0,373	0,375	0,575
CEI_lag1	0,242	0,235	#N/A
BCI_lag1	0,109	0,105	0,191
TB_Yield_10Y_lag1	0,091	0,095	0,149
EPUI_lag1	0,069	0,072	#N/A
ISM_PMI_lag1	0,056	0,055	#N/A
RSI_lag1	0,020	0,021	0,034
EMUI_lag1	0,015	0,016	#N/A
MACD_Signal_lag1	0,010	0,009	0,022
MACD_Line_lag1	0,006	0,006	0,013
Stochastic Oscillator (%D)_lag1	0,004	0,004	0,004
Stochastic Oscillator (%K)_lag1	0,003	0,003	0,004
MACD_Diff_lag1	0,002	0,002	0,006
William (%R)_lag1	0,001	0,002	#N/A
Sentiment Score_ pre-trained DistilBERT _lag1	0,001	#N/A	#N/A
Sentiment Score_Textblob_lag1	#N/A	0,001	0,002

Finally, residual analysis shown in Figures 10, 11 and 12 indicated the model's strong prediction performance for all scenarios. In scenario A, residuals remained within reasonable bounds, except for some predictions where larger deviations were recorded, such as -31.35 and 27.96. The application of Sentiment Score_Textblob selection had a similar trend as in scenario A, where residuals were found at -27.72 and 21.23. In scenario C, feature selection significantly improved the residual values, varying from -14.35 to 20.65. These results indicated reduced variability and closer alignment with the actual values. They also confirmed that key predictors have a significant impact on the model's performance when it refers to its capability of handling stable and dynamic market conditions with a reduced error margin.

Figure 10. Random Forrest Regressor Results (Scenario A)

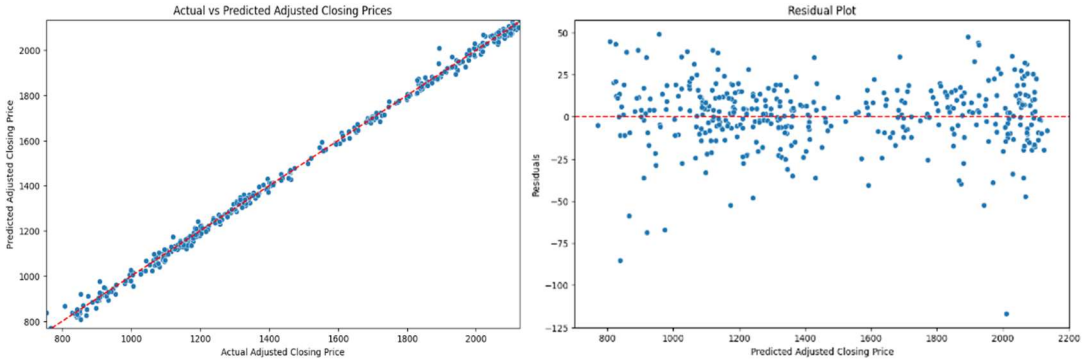


Figure 11. Random Forrest Regressor Results (Scenario B)

Figure 12. Random Forrest Regressor Results (Scenario C)

5.3. Gradient Boosting Regressor

After employing the GB algorithm, results provided negligible differences between the scenarios tested, as seen in Table 5-5. For the no-feature selection scenarios sentiment score_ pre-trained DistilBERT scenario had an MSE of 407,98 and an MAE of 14,79 while sentiment score_Textblob scenario yielded a higher MSE of 408,05 and MAE of 14,80. Despite these differences, both models showed high R^2 values, underlining the model's robust predictive power and capability to capture the underlying dynamics of the data. Employing the model after feature selection resulted in both MAE

and MSE increasing to 15,16 and 424,83, respectively, while R^2 remained very high at 0.9974. Limiting features to the most relevant for building a model with high explanatory power proved that the overall accuracy of the predictions slightly deteriorated.

Table 5-5. Gradient Boosting (Results)

Scenarios			MSE	MAE	R^2
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	407,98	14,79	0,9975
b	No-Feature Selection	Sentiment Score_Textblob	408,05	14,80	0,9975
c	Feature Selection	Up to rank 2	424,83	15,16	0,9974

Next, feature importance analysis (Table 5-6) indicated that EMA_lag1 was the most relevant feature for all scenarios, due to its significance in capturing price movement trends, with a value of 0,801 for both Sentiment Score_ pre-trained DistilBERT and Sentiment Score_Textblob scenarios, respectively. Similarly, in the feature-selected model, EMA_lag1 dominated among the other features with a value of 0,883. With respect to the rest of the features, CEI_lag1 and TB_Yield_10Y_lag1, demonstrated moderate importance, while sentiment-based features exhibited even lower importance.

Table 5-6. Gradient Boosting (Feature Importance)

	Scenarios		
Feature	**A**	**B**	**C**
Exponential Moving Average (EMA)_lag1	0,801	0,801	0,883
CEI_lag1	0,146	0,146	#N/A
BCI_lag1	0,032	0,032	0,090
ISM_PMI_lag1	0,010	0,010	#N/A
TB_Yield_10Y_lag1	0,006	0,006	0,022
RSI_lag1	0,002	0,002	0,003
William (%R)_lag1	0,001	0,001	#N/A
MACD_Diff_lag1	0,001	0,001	0,001
MACD_Line_lag1	0,000	0,000	0,000
Stochastic Oscillator (%K)_lag1	0,000	0,000	0,001
Stochastic Oscillator (%D)_lag1	0,000	0,000	0,000
MACD_Signal_lag1	0,000	0,000	0,000

EPUI_lag1	0,000	0,000	#N/A
EMUI_lag1	0,000	0,000	#N/A
Sentiment Score_ pre-trained DistilBERT _lag1	0,000	#N/A	#N/A
Sentiment Score_Textblob_lag1	#N/A	0,000	0,000

With regards to the residual analysis, it was found that residuals were mainly between acceptable bounds with some variation in instances depending on either the sentiment score or the feature selection (Figures 13, 14 and 15). Without feature selection, residuals using Sentiment Score_ pre-trained DistilBERT were at -21.50 in one instance, while slightly higher predictive error was noticed when using Sentiment Score_Textblob, as the value was -33.32. Feature selection generally resulted in improved residual behavior. Deviations between the actual and predicted values became smaller. For instance, the residual of -6.79 in the feature-selected model is smaller compared to larger deviations in models without feature selection. Yet, even with such improvements, all models show excellent performance in stable market conditions but reveal their limitations during volatile or extreme periods.

Figure 13. Gradient Boosting Results (Scenario A)

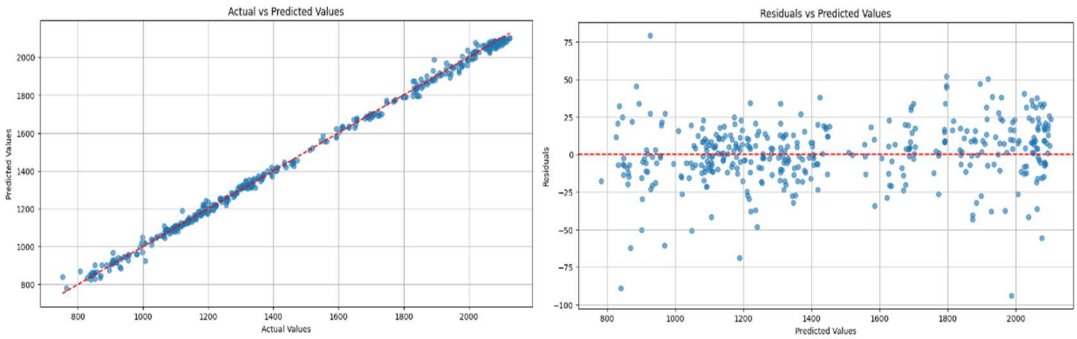


Figure 14. Gradient Boosting Results (Scenario B)

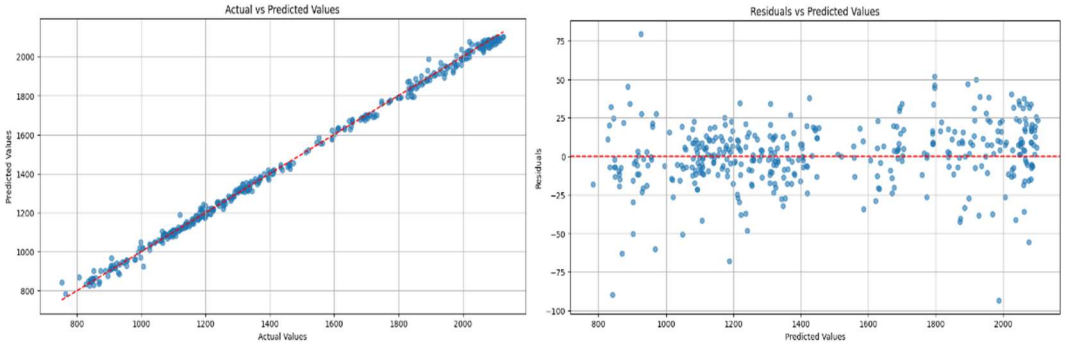
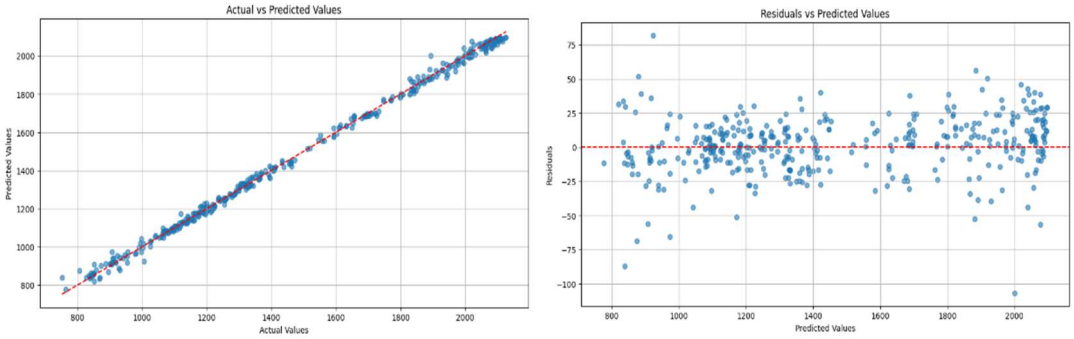


Figure 15. Gradient Boosting Results (Scenario C)



5.4. XGBoost Regressor

XGBoost Regressor also provided high accuracy results for all scenarios with minor variations stemming mainly from the feature engineering strategy (Table 5-7). For no feature selection scenarios, employing the Sentiment Score_Textblob feature resulted in an MSE of 908,73, an MAE of 23,97 and R^2 of 0.9947. Instead, adopting Sentiment Score_ pre-trained DistilBERT feature in the model, resulted in an MSE of 905,29, an MAE of 23,95, and an R^2 of 0.9947. Employing only the features as suggested by RFE resulted in a minor effect on the performance of the model, as MSE increased to 1159,31 and MAE to 26,65, while R^2 slightly decreased to 0.9932.

Table 5-7. XGBoost Regressor (Results)

Scenarios			MSE	MAE	R^2
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	905,29	23,95	0,9947
b	No-Feature Selection	Sentiment Score_Textblob	908,73	23,97	0,9947
c	Feature Selection	Up to level 2	1159,31	26,65	0,9932

In the next step of feature importance analysis shown in Table 5-8, CEI_lag1 was found to be the most important predictor in the models trained so far, in the first two scenarios. In particular the feature contributed more than 48% to the model's importance for scenarios A and B, while other variables, such as (EMA)_lag1, ISM_PMI_lag1 and BCI_lag1, had a relatively minor contribution. In the feature selection scenario, EMA_lag1 importance increased up to 65%, reflecting the model's reliance on this technical indicator. Features like TB_Yield_10Y_lag1and RSI_lag1 increased their

importance weight after feature selection, supporting their impact in prediction accuracy improvement.

Table 5-8. XGBoost (Feature Importance)

	Scenarios		
Feature	**A**	**B**	**C**
CEI_lag1	0,482	0,481	#N/A
Exponential Moving Average (EMA)_lag1	0,273	0,273	0,653
BCI_lag1	0,071	0,071	0,164
ISM_PMI_lag1	0,051	0,052	#N/A
TB_Yield_10Y_lag1	0,043	0,043	0,097
EPUI_lag1	0,037	0,035	#N/A
MACD_Signal_lag1	0,012	0,013	0,028
EMUI_lag1	0,011	0,012	#N/A
RSI_lag1	0,010	0,010	0,030
MACD_Line_lag1	0,005	0,005	0,018
Stochastic Oscillator (%D)_lag1	0,003	0,003	0,005
Stochastic Oscillator (%K)_lag1	0,001	0,001	0,002
William (%R)_lag1	0,001	0,001	#N/A
MACD_Diff_lag1	0,001	0,001	0,002
Sentiment Score_ pre-trained DistilBERT _lag1	0,000	#N/A	#N/A
Sentiment Score_Textblob_lag1	#N/A	0,000	0,001

In residual analysis, the model's accuracy is further confirmed as shown in Figures 16 - 18. Without feature selection, scenarios are characterized by small variations in residuals ranging between -10.59 and 19.31, thus capturing reasonable accuracy market trends. In feature selection scenario, residuals kept this trend, as minimal deviations from actual values were reported raging between -0.19 and 14.55.

Figure 16. XGBoost Regressor Results (Scenario A)

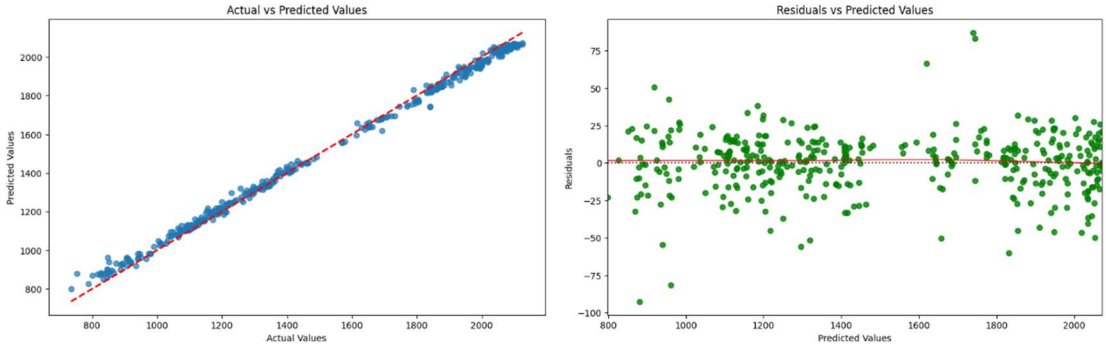


Figure 17. XGBoost Regressor Results (Scenario B)

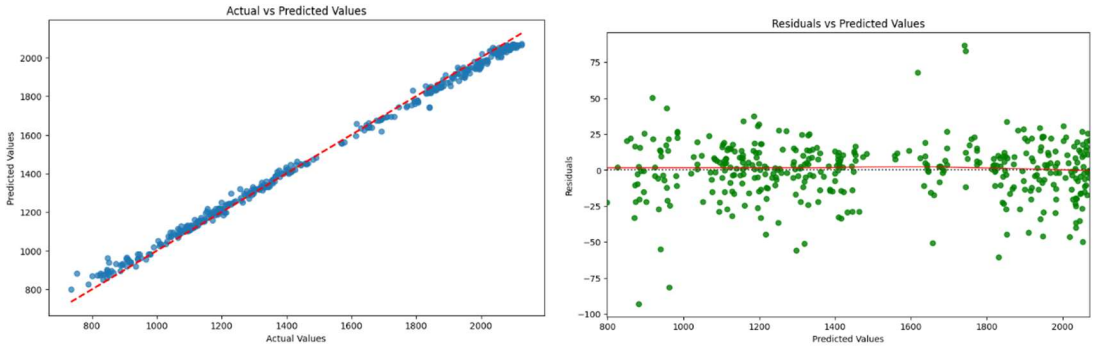
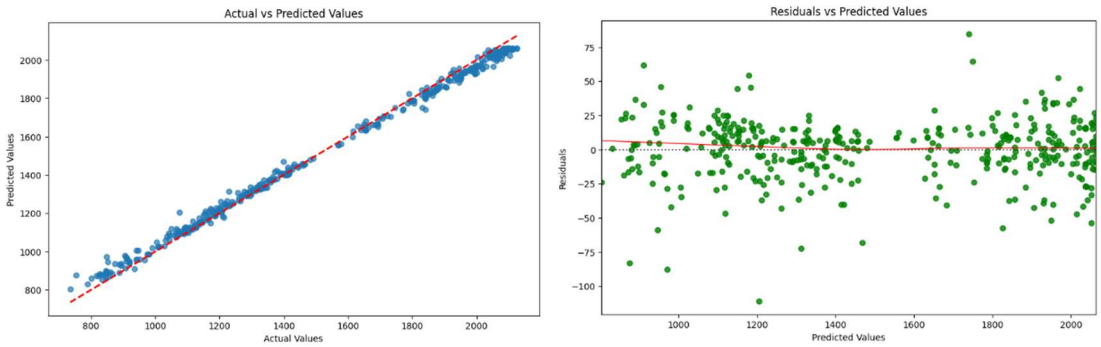


Figure 18. XGBoost Regressor Results (Scenario C)



5.5. MLP Regressor

The last model employed in this study was MLP Regressor. Similar to the results of the other models, minor variations in accuracy were estimated, depending mainly on the use of feature selection and sentiment score sources, as seen in Table 5-9. No feature selection scenario A achieved an MSE of 410.35, an MAE of 14,39, and an R^2 of 0.9976. In scenario B, MSE and MAE were found at 413,13 and 15,50, respectively, with an improved R^2 of 0.9976, reflecting even better fit and stronger ability to capture the underlying data patterns. The model continued to perform well with feature selection

criteria, as MSE was found at 336,53, MAE at 13.90, and R^2 at 0.9980. This approach, while slightly more accurate, streamlined the model and improved computational efficiency.

Table 5-9. MLP Regressor (Results)

Scenarios			MSE	MAE	R^2
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	410,35	14,39	0,9976
b	No-Feature Selection	Sentiment Score_Textblob	413,13	15,50	0,9976
c	Feature Selection	Up to level 2	336,53	13,90	0,9980

Feature importance analysis shown in Table 5-10 indicated that EMA_lag1 dominated among all scenarios, with an importance score between 1.778 and 1.847, making it again the most critical feature in the prediction. Secondary features with an adequate importance to the model were Stochastic Oscillators, BCI_lag1, and MACD_Signal_lag1. Sentiment scores had minor positive effects that added minor context for improving the overall accuracy. During feature selection, lower-relevance variables, including ISM_PMI_lag1 and William (%R) _lag1, were excluded so that the model focuses only on impactful predictors.

Table 5-10. MLP Regressor (Feature Importance)

	Scenarios		
Feature	**A**	**B**	**C**
Exponential Moving Average (EMA)_lag1	1,831	1,778	1,847
MACD_Signal_lag1	0,008	0,006	0,027
BCI_lag1	0,008	0,005	0,004
Stochastic Oscillator (%K)_lag1	0,007	0,006	0,011
Stochastic Oscillator (%D)_lag1	0,007	0,008	0,009
ISM_PMI_lag1	0,005	0,011	#N/A
TB_Yield_10Y_lag1	0,004	0,005	0,003
MACD_Diff_lag1	0,003	0,001	0,001
CEI_lag1	0,003	0,004	#N/A
EMUI_lag1	0,002	0,000	#N/A
RSI_lag1	0,002	0,003	0,006
William (%R)_lag1	0,002	0,001	#N/A

MACD_Line_lag1	0,002	0,003	0,035
EPUI_lag1	0,001	0,000	#N/A
Sentiment Score_ pre-trained DistilBERT _lag1	0,000	#N/A	#N/A
Sentiment Score_Textblob_lag1	#N/A	0,000	0,000

Model accuracy was also evident in residual analysis as seen in Figures 19 - 21. Most of residuals exhibited minimal deviation suggesting high predictive power between the actual values and their corresponding predictions. In scenario A, without feature selection, residuals ranged from -22.48 to 21.91 while in scenario B from -19.04 to 13.46, reflecting the superior alignment of the predicted values. The feature selection scenario exhibited similar residual variation, but emphasized the model's reliance on fewer yet impactful predictors.

Figure 19. MLP Regressor Results (Scenario A)

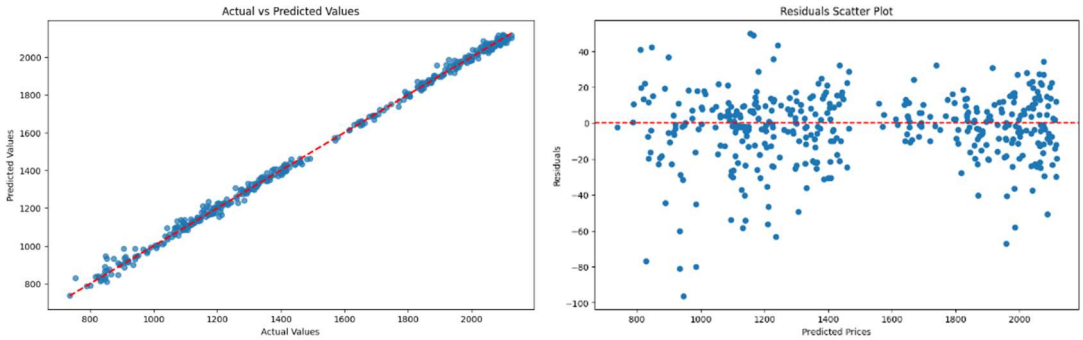


Figure 20. MLP Regressor Results (Scenario B)

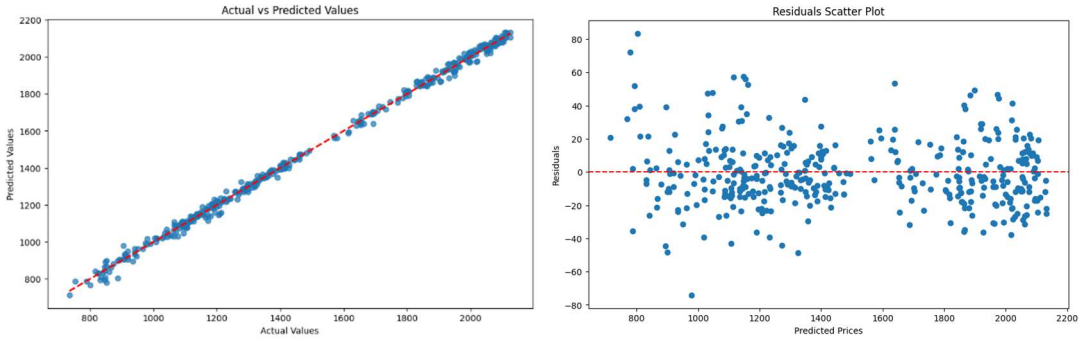
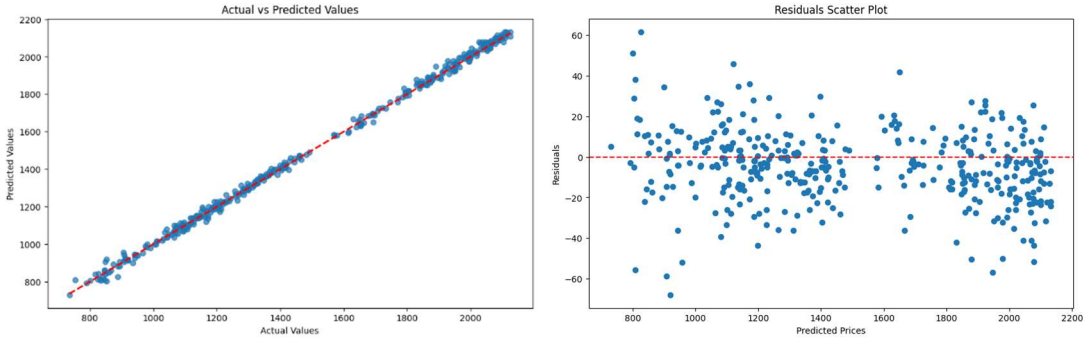


Figure 21. MLP Regressor Results (Scenario C)



5.6. Overfitting Evaluation

Overfitting is considered a drawback on the results in predictive modeling, as it suggests that a model adapts to patterns specific to the training data rather than generalizable trends. After overfitting was spotted during the initial modeling phase, given the tuning provided by the RandomizedSearchCV technique, efforts were made to enhance generalization by refining hyperparameters and making use of regularization techniques in order to reach robust results. In this subsection, the evaluation of overfitting is discussed along with the effect of the additional tuning on the hyperparameters for the models, where overfitting was evident, comparing the performance metrics in each scenario, including the cross-validation MSE and the learning curve graphical representation.

5.6.1. Linear Regression

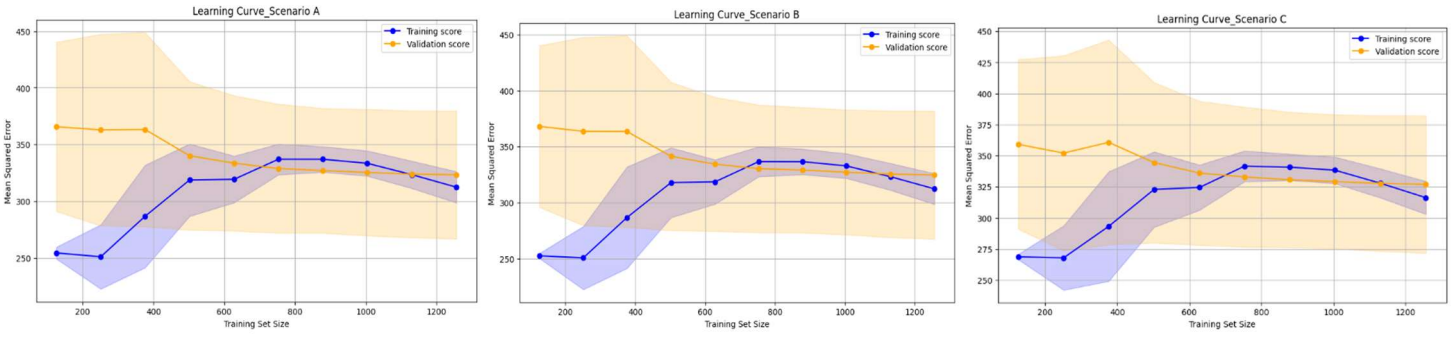
In LR, no signs of overfitting were found in the results. As suggested by the metrics in Table 5-11, R^2 were identical for both training and test datasets, indicating the model generalizes well, and this was further supported by marginal difference in errors. Cross-validation MSE was also close to the MSEs of the datasets presenting good generalization as the model performed consistently across different data splits. This indication was evident for all scenarios.

Table 5-11. Linear Regression (Overfitting Results)

Scenarios			Dataset Type	MSE	MAE	R^2	Cross-Validation MSE
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	313,20	12,74	0,9981	323,21
			Test	370,05	13,79	0,9977	
b	No-Feature Selection	Sentiment Score_Textblob	Training	313,42	12,74	0,9981	324,71
			Test	371,03	13,84	0,9977	
c	Feature Selection	Up to level 2	Training	317,20	12,80	0,9980	326,98
			Test	375,19	13,88	0,9977	

This is further confirmed by the Learning Curves for each scenario. As seen in Figure 22, training MSE and validation MSE begin to converge as long as the training size increases. This holds true as the validation MSE decreases gradually up to the point where it is stabilized close to the training MSE. There is also evident that there is a balance between training and validation MSEs as the former is not excessively low compared to the latter.

Figure 22. Linear Regression Learning Curve



5.6.2. Random Forest

Initial hyperparameter tuning, as suggested by the RandomizedSearchCV technique, resulted in a large discrepancy between the training and testing datasets, when

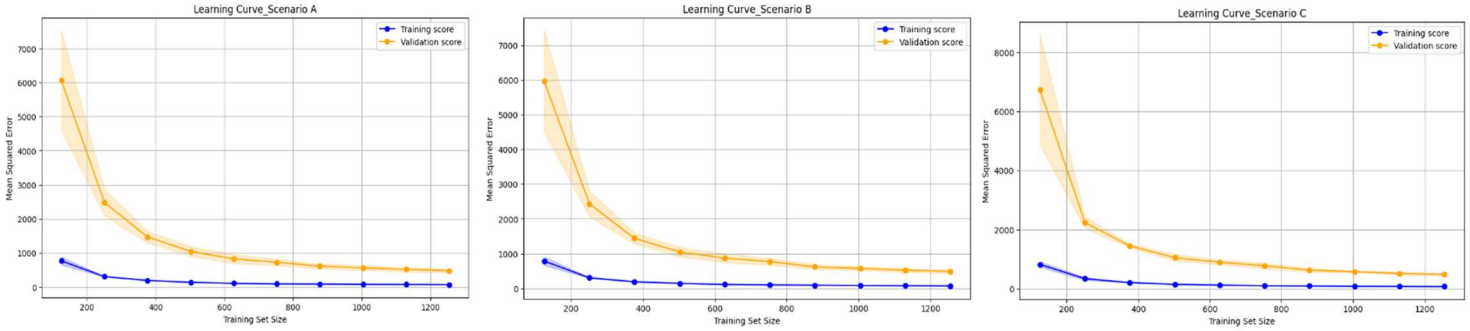
RF was implemented. As seen in Table 5-12, the model fitted the training data exceptionally well, but at the same time it was unable to adapt to new unseen data.

Table 5-12. Random Forrest (Overfitting Results)

Scenarios			Dataset Type	MSE	MAE	R^2	Cross-Validation MSE
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	60,75	5,76	0,9996	475,99
			Test	361,42	13,99	0,9979	
b	No-Feature Selection	Sentiment Score_Textblob	Training	60,31	5,69	0,9996	488,40
			Test	378,09	14,26	0,9978	
c	Feature Selection	Up to level 2	Training	62,32	5,68	0,9996	480,65
			Test	371,78	13,73	0,9978	

This is further supported by the learning curves for each scenario in Figure 23. Training MSE was reportedly low, irrespective of the training set size, while the validation score decreased, but their gap remained substantial as the training set size increased.

Figure 23. Random Forrest Learning Curve (Random Search Tuning)



In view of the overfitting in the results, further tuning to the hyperparameters of the model was done to reduce it, but it still remained evident to some extent. As seen in Table 5-13, tuning assisted the model to reduce the discrepancy in the errors between the

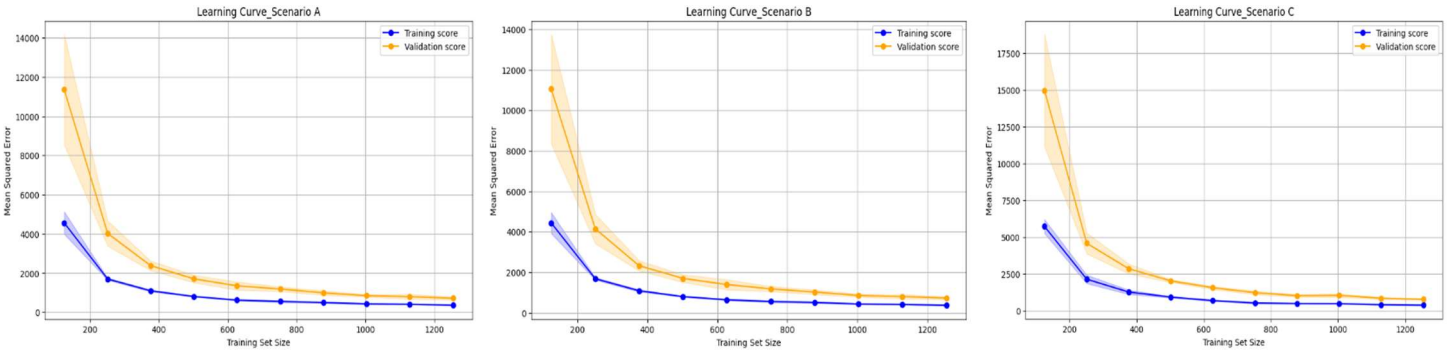
training and testing datasets, but cross-validation MSE was found to be much higher than the training MSE in all scenarios.

Table 5-13. Random Forrest (Overfitting Results After Additional Tuning)

Scenarios			Dataset Type	MSE	MAE	R^2	Cross-Validation MSE
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	318,41	13,07	0,9980	711,69
			Test	533,58	16,98	0,9969	
b	No-Feature Selection	Sentiment Score_Textblob	Training	323,42	13,14	0,9980	722,92
			Test	540,30	17,27	0,9968	
c	Feature Selection	Up to level 2	Training	302,90	12,52	0,9981	771,25
			Test	556,94	17,15	0,9967	

This case is further confirmed by the learning curves in Figure 24, where MSE for the training set is still low, irrespective of the training set size. Gaps between the sets are also apparent, but not in the same magnitude as in the initial tuning.

Figure 24. Random Forrest Learning Curve (Additional Tuning)



5.6.3. Gradient Boosting

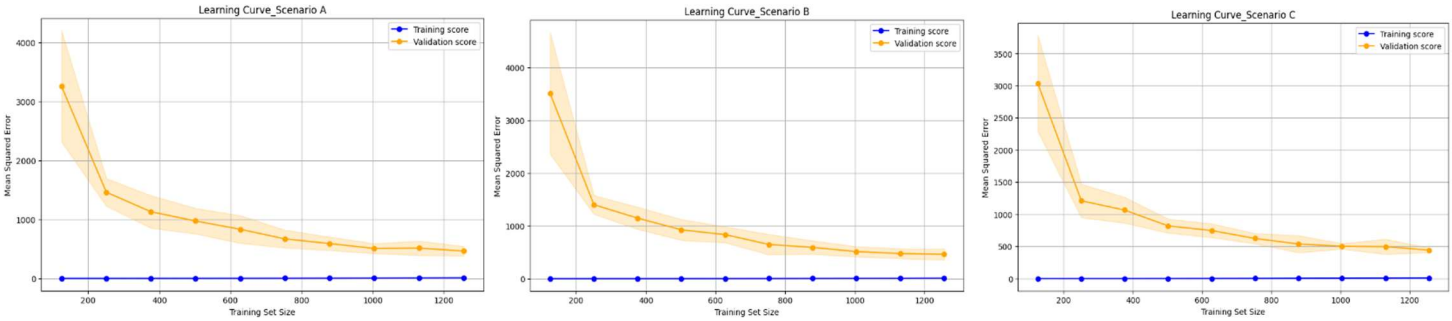
Similarly to the results of the initial tuning of RF, overfitting was also evident after the implementation of GB on the hyperparameters provided by the RandomizedSearchCV technique. Again, the large disparity of the errors between the training and testing datasets is evident in Table 5-14, while the gap between the training

and validation curves in Figure 25 and their inability to converge further indicates that the model is overfitting.

Table 5-14. Gradient Boosting (Overfitting Results)

Scenarios			Dataset Type	MSE	MAE	R^2	Cross-Validation MSE
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	15,52	3,09	0,9999	458,13
			Test	524,65	16,22	0,9968	
b	No-Feature Selection	Sentiment Score_Textblob	Training	14,81	3,01	0,9999	490,32
			Test	476,39	15,84	0,9971	
c	Feature Selection	Up to level 2	Training	15,28	3,07	0,9999	437,37
			Test	496,40	15,57	0,9970	

Figure 25. Gradient Boosting Learning Curve (Random Search Tuning)

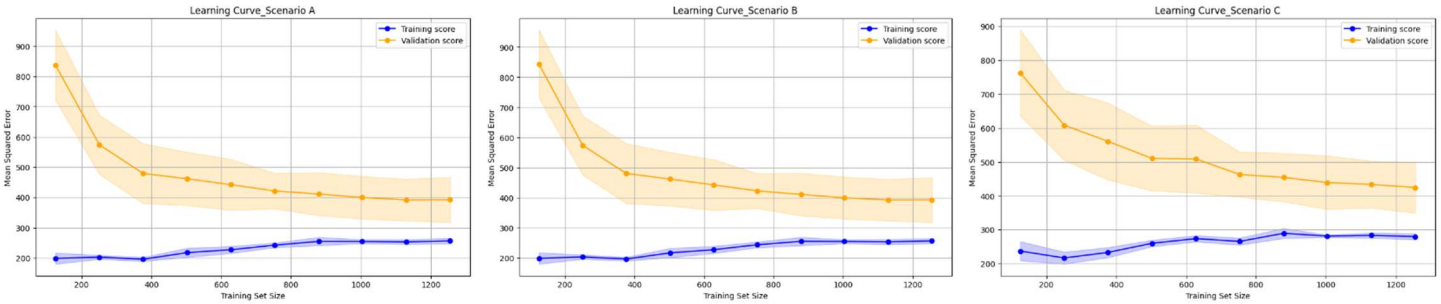


Additional tuning to the hyperparameters did not affect significantly the overall overfitting of the model, as the gap between the errors remained high indicating again that the model still performed significantly better on the training data than on unseen test data. As seen in Table 5-15, cross-validation error remained also higher than that of the training data for every scenario implying the existence of overfitting. Learning curves in Figure 26 remained mainly the same with a slight improvement in training curve effort to converge with that of validation curve, but the gap between the curves was still large.

Table 5-15. Gradient Boosting (Overfitting Results After Additional Tuning)

Scenarios			Dataset Type	MSE	MAE	R^2	Cross-Validation MSE
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	261,98	12,17	0,9984	385,91
			Test	407,98	14,79	0,9975	
b	No-Feature Selection	Sentiment Score_Textblob	Training	261,80	12,17	0,9984	386,21
			Test	408,05	14,80	0,9975	
c	Feature Selection	Up to level 2	Training	284,18	12,83	0,9982	419,06
			Test	424,83	15,16	0,9974	

Figure 26. Gradient Boosting Learning Curve (Additional Tuning)



5.6.4. XGB Regressor

XGB regressor followed the same trend during the initial tuning, providing overfitted results. As per Table 5-16, training error was substantially lower than that of the test error for all scenarios accompanied by an even larger cross-validation error.

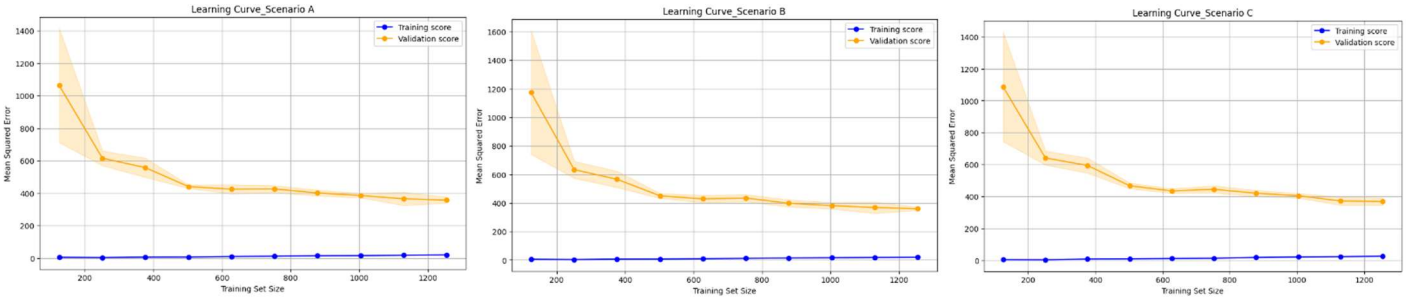
Table 5-16. XGB Regressor (Overfitting Results)

Scenarios			Dataset Type	MSE	MAE	R^2	Cross-Validation MSE
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	25,29	3,94	0,9998	358,24
			Test	285,44	12,22	0,9983	
b			Training	25,31	3,93	0,9998	359,21

	No-Feature Selection	Sentiment Score_Textblob	Test	287,20	12,26	0,9983	
c	Feature Selection	Up to level 2	Training	36,30	4,69	0,9997	366,90
			Test	300,40	12,52	0,9982	

The magnitude of gap of the learning curves in Figure 27 was also large indicating further the inability of the model to perform better on unseen data compared to training data.

Figure 27. XGBoost Regressor Learning Curve (Random Search Tuning)



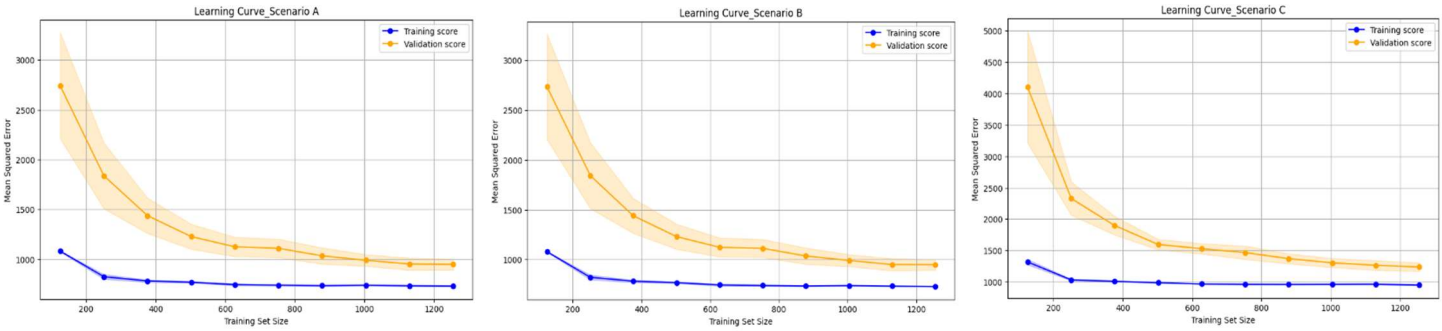
After tuning the hyperparameters, overfitting was still evident, but not to the same extent as in the initial tuning. The discrepancy in error between the datasets was somehow reduced, but the cross-validation MSE remained relatively higher than that of the training dataset in all scenarios (Table 5-17). This was also confirmed by the learning curves (Figure 28), where the two curves are unable to converge, since the training score remains stable, while the training set size increases.

Table 5-17. XGB Regressor (Overfitting Results After Additional Tuning)

Scenarios			**Dataset Type**	**MSE**	**MAE**	**R^2**	**Cross-Validation MSE**
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	728,51	21,59	0,9954	947,62
			Test	905,29	23,95	0,9947	
b			Training	728,15	21,57	0,9955	948,27

	No-Feature Selection	Sentiment Score_Textblob	Test	908,73	23,97	0,9947	
c	Feature Selection	Up to level 2	Training	947,84	23,97	0,9941	1239,48
			Test	1159,31	26,65	0,9932	

Figure 28. XGBoost Regressor Learning Curve (Additional Tuning)



5.6.5. MLP Regressor

Tuning hyperparameters under the input of RandomizedSearchCV technique provided mixed overfitting results, even though there was an improvement in generalization performance. As seen in Table 5-18, in no feature selection scenarios is the model slightly overfitting, considering the gap between the test and the training MSE. This is also supported by the higher Cross-Validation MSE compared to the training MSE. While overfitting in scenario B is lower than scenario A, in scenario C, where feature selection is implemented, no signs of overfitting are observed. In this scenario test MSE is lower than training MSE, suggesting improved generalization performance and this is further supported with MAE values. Cross validation MSE is also the lowest among the scenarios, further supporting the lack of overfitting in scenario C.

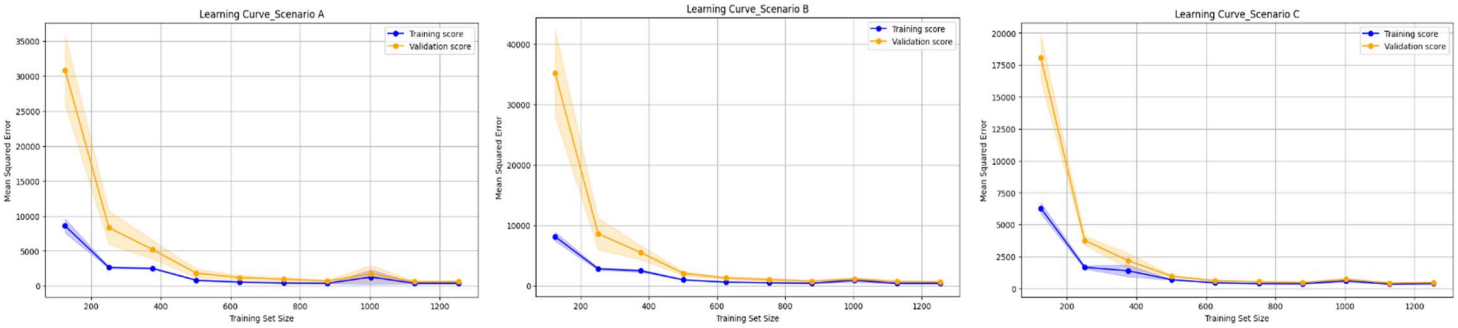
Table 5-18. MLP Regressor (Overfitting Results)

Scenarios			**Dataset Type**	**MSE**	**MAE**	**R^2**	**Cross-Validation MSE**
a	No-Feature Selection	Sentiment Score_ pre-trained DistilBERT	Training	359,66	14,52	0,9978	565,94
			Test	410,35	14,39	0,9976	
b			Training	381,71	14,91	0,9976	593,18

	No-Feature Selection	Sentiment Score_Textblob	Test	413,13	15,50	0,9976	
c	Feature Selection	Up to level 2	Training	355,74	14,30	0,9978	451,50
			Test	336,53	13,90	0,9980	

Learning curves in Figure 29 further confirm the initial evaluation of overfitting. While the curves converge in all scenarios, in scenario C this is observed in smaller training set sizes.

Figure 29. MLP Regressor Learning Curve (Random Search Tuning)



6. Discussion

In this study, an attempt has been made to predict the S&P 500 closing price, based on a diverged set of input features by employing a spectrum ranging from simple to advanced ML techniques. The combination of technical, macroeconomic and sentiment indicators was introduced in three different scenarios in order to reflect the insight of price trends, broader economic conditions and behavioral dimension into the market's price movements.

Results indicated distinct patterns of these features in explaining their ability to predict the market. Initially, sentiment features were found to have negligible contribution suggesting a significant low impact to drive short market fluctuations. Macroeconomic features, while being a mirror of the economic conditions, produced subdued effects in short term predictions, partially explained by their lagging nature and the daily prediction frequency. On the other hand, technical features related to momentum and volatility were found to contribute significantly, aligning with the TA theory that supports their ability to capture periods of overreaction and correction in financial markets.

In terms of the models' predictive power, all models were found to provide a perfect fit, perfectly explaining the variability of the input features in the S&P 500 price for all case scenarios. LR and MLP were the drivers among all models, providing high R^2 and low errors, supporting the concept that traditional and more advanced models can provide comparable results. MLP was found to better capture the dependencies found in scenario C, where features were defined by RFE. Similar results were also found in other models, but overfitting was evident, even after additional tuning of the initial architecture of the hyperparameters provided by the RandomizedSearchCV technique.

While these findings indicate that the combination of these features can perfectly predict index price movements, further balance between model complexity and generalizability is also evident. Despite the efforts towards mitigating overfitting, such as hyperparameter tuning, cross-validation, and feature selection using RFE, to draw more robust conclusions about the interplay between features and their contribution to market predictions, RF, GB and XGB Regressor where not found reliable.

In all, this analysis leads to the conclusion that stock market behavior is so complex that no single variable acts dominantly. The integration of technical, macroeconomic, and sentiment data provides a comprehensive perspective, but also brings out the requirement for rigorous preprocessing and thoughtful choice of modeling strategies. While predictive accuracy can be realized, the nuanced interactions between variables continue to be a limiting factor for financial time series forecasting.

6.1. Benchmarking Results against the Literature

The aim of this study was to challenge and further enhance the predictive power of the models employed in the literature in the scope of finding the optimal feature combination to predict the stock market. While each study addresses distinct aspects of forecasting stock prices, the focus of this subsection is to highlight the limitations that this work aimed to overcome.

Sangeetha and Alfia (2024) and the current work share the objective of predicting the S&P 500 index, but differ in their methodology. In their study they use basic stock market features (Open, Close, Low, High, Volume) and implement the Evaluated Linear Regression-based ML (ELR-ML) technique, achieving an R^2 of 0.428 and an Adjusted R^2 of 0.352. Error metrics in their study show an SSE of 6E+13 and an MSE of 9E+12, suggesting higher prediction deviations. In contrast, the models in this study achieved R^2 values nearing 0.998 and much lower errors, with an MSE as low as 261.98 for GB, underscoring the advantages of diverse features and sophisticated models. While in their study they employ a simple feature set that limits its ability to capture complex market dynamics, our study's integration of richer data sources improves accuracy significantly. This limitation highlights the weakness of LR to explain non-linear financial data, while the advanced algorithms of this study can excel with proper feature engineering. Combining our study's diverse features with Sangeetha and Alfia (2024) residual analysis could refine both methodologies, emphasizing the importance of robust features and advanced techniques for accurate stock market forecasting.

Sunantha (2020) has also implemented LR and additionally NN in order to predict the stock price movements in Shanghai Stock Exchange using 21 indicators, categorized into macroeconomic, microeconomic, sentiment, and institutional investor data. In their study they found that NN generally outperformed Regression models supported by p-values ranging from 0.08 to 1.00 for paired t-tests, with NN showing

lower APE in most sectors. While their work suggested the superiority of NN models against basic regression models, their improved ability is not that evident in sectors with lower volatility. The sophisticated models in our study offer the ability to handle non-linear relationships more effectively that may explain its higher predictive accuracy compared to both NN and OLSR in Sunantha (2020).

The study from Jabeur et al (2024) utilizes various ML models differentiating significantly from our work in their feature sets and target variables. Both studies highlight the effectiveness of XGBoost in forecasting financial data when various input variables are utilized. In Jabeur et al (2024), XGBoost achieved the highest R^2 of 0.994 with RMSE of 34.921 and MAE of 21.968, showing strong accuracy in forecasting gold prices similar to the R^2 of 0.998 of our study. While errors were significantly lower than in our study for XGBoost and the other models (NN RMSE 195.961, LR RMSE 71.325), the advantage of ensemble models is evident in both studies, highlighting the importance of advanced algorithms for complex market data.

Compared to more advanced NN architectures studies in the literature, our study demonstrates the advantage of combining various features for broader market forecasting and at the same time the disadvantage of higher errors obtained when simpler models are implemented. In particular, Agrawal et al. (2019) employ deep NN (Optimal LSTM and ELSTM) in order to predict banking stock prices using only technical indicators. Their results show superior performance with high accuracies in prediction of 63.59% for HDFC, 56.25% for YES Bank, and 57.95% for SBI, surpassing classical models like SVM and Logistic Regression that were also tested and ranged between 49% to 56%. In addition, the obtained MSEs for deep NN model were found at 0.015 and 0.017 being significantly lower than in our study's results. While deep NN models performed significantly better in terms of error measurement, the importance of leveraging diverse features for a more comprehensive market analysis, as shown in our study could be considered complementary to enhance the models.

Additional work that supports this novelty is further found in Chang et al. (2024) where they predict individual stock prices after employing deep learning models, such as GRU and LSTM, on their temporal dependencies and achieve an RMSE as low as 3.43 for Apple and 8.08 for Microsoft, significantly outperforming traditional methods. In addition, Bhandari et al. (2022) employ LSTM models with different neuron

configurations in order to predict the S&P 500 price using a combination of fundamental, macroeconomic, and technical indicators and report RMSE values ranging from 46.5 to 167.5 and high R^2 values between 0.9935 and 0.9967.

6.2. Threats to validity and limitations

Despite the valuable insights of this study, there are yet a few limitations. The first threat to validity of this study is linked to the indicators selected. Even though the technical and macroeconomic variables that were selected vary among their inferences, it is possible that other influential variables, such as sectoral metrics or global economic indicators, which could have greater contribution in forecasting, might have been left out. Not incorporating in the dataset such feature could negatively affect the strength of the models in capturing stock market dynamics.

Secondly, the decision to incorporate macroeconomic indicators that calculate monthly values and lag them by one month raises questions about their temporal relevance. Economic conditions are generally reflected in the market over longer or variable time frames, and a uniform one-month lag may not adequately capture their effect. Similarly, lagging all other technical and macroeconomic indicators by one day may have constrained the analysis further, since they might require longer periods of lagging in order to reflect their impact accurately.

Forward-filling of monthly macroeconomic data is another threat in this research. While this approach is simple and effective to handle data, it can reduce model accuracy. In this study forward-filling assumes that monthly values reflect the preceding days of the month, and this could potentially misrepresent trends or variations that could influence the market.

Moreover, the scope of the sentiment analysis in this study is based on the effect of general world news on market price conditions. The open economic market that we reside in has set strong dependencies among countries, industries and companies in order to achieve economic growth and development. US economy is dominant worldwide and therefore companies that are incorporated in the S&P 500 can be heavily affected by world developments. While the focus in this study was to deviate from the usual implementation of examining sentiment from financial news and utilize the potential of highlighting the impact of world news on stock market movements, no

significant relationship was found. A possible reason for this limitation could be the source and magnitude of the data retrieved from Reddit. As discussed in the data section, earlier data for sentiment retrieved from /r/worldnews, which can be considered a narrow perspective on market sentiment not focusing particularly on world events, since they might carry information irrelevant to impact business outcome. In order to provide a more detailed base of the behavior in the market, this could be further enriched by a more targeted source, such as business news or relevant social media.

Furthermore, another limitation could be related to feature collinearity. For instance, RSI, EMA, and MACD are computed using overlapping calculations, which might induce redundancy and decrease the marginal contribution of each variable. This might seriously affect the interpretability and the predictive performance of the model, thus requiring a more thoughtful selection and evaluation process.

These limitations highlight the complexity of financial forecasting and the trade-offs inherent in data preparation and modeling choices. These challenges call for the thoughtful refinement of methods and assumptions in future studies.

6.3. Future directions

Implications for future work to enhance the robustness of the models for stock price predictions are evident. Additional metrics that reflect particular sectors of the market and macroeconomic conditions can enhance the prediction power of the models, while alternative sources to broaden the scope of sentiment analysis in order to capture behaviors from financial news social media discussions could strengthen financial modelling.

In addition, applying optimized methods like rolling window regression, transfer entropy etc. in order to assess the optimal lag periods for the macroeconomic indicators could be vital for explaining market prediction. Similarly, the investigation of alternative imputation techniques for monthly data may reduce risks associated with forward-filling and improve data fidelity.

Finally, principal component analysis or feature selection through clustering could be also employed to avoid feature collinearity and improve model interpretability and performance [72]. By addressing this issues, further work can build on the ground led by this study and advance financial modelling to produce more actionable insights.

7. Conclusion

The ability to predict stock market movements has always been of great importance in financial research and practice, due to the potential that it provides for investors, financial analysts, and policy makers to improve their decisions and manage risks. The aim of this study was to develop a comprehensive framework of financial modelling that integrates various input features, such as technical indicators, macroeconomic variables, and sentiment scores as predictors that can address the challenges that come with forecasting stock prices.

The financial modelling developed in this study utilizes ML regression models, including LR, RF, GB, XGBoost, and MLP, that aims to investigate how all these factors combined can eventually provide better accuracy in predicting the adjusted closing price of the S&P 500 index. Results suggest useful insights into the predictive power of different feature combinations and the comparative performances of different ML models in this context.

One of the contributions of this study is that it introduces macroeconomic indicators into the forecasting architecture, in addition to more commonly used technical and sentiment-based features. Given the preprocessing steps and the feature engineering procedure as parts of the overall architecture, variables were lagged appropriately indicating EMA and MACD as the features with the highest contribution. Macroeconomic factors, such as the BCI and CEI were able to modestly explain market trends, confirming technical indicators' established relevance for financial forecasting and modelling in the literature. For additional sentiment information, sentiment scores were computed using TextBlob and HF tools, but their contribution in this study was proved negligible.

Based on the feature dataset chosen for this study, modelling architecture exploits the effectiveness of various traditional and more advanced ML models in order to capture complex and nonlinear relationships hidden in stock market data. Among the models tested, LR and MLP Regressor exhibited superior performance, achieving high R^2 scores of 0.99 and low MSE and MAE rates averaging 350 and 13 points respectively, across both training and test datasets. Additional techniques for robust feature selection, such as RFE, were also utilized, improving slightly model efficiency in terms of error

prediction, as predictive accuracy is already high. However, challenges such as overfitting found in other models underscore the importance of careful hyperparameter tuning and cross-validation techniques to enhance generalizability.

While the results of this study indicate important implications for investors to identify market trends and researchers to further explore hybrid models combining various sets of features with modern ML techniques, limitations are also evident, possibly threating the validity of the results. Shortcomings related to the choice of macroeconomic indicators, the lagging timeframe of features and resampling, the nature and quality of contextual data and the possible collinearity of technical indicators can significantly challenge the credibility of the results.

In conclusion, this study aims to add value to the existent literature to understand stock market predictions dynamics by incorporating the macroeconomic factor in ML models. Despite its possible limitations, actionable findings that serve as a strong base can be identified that could contribute to the growing field of financial forecasting.

References

[1] Fama, E.F. (2017), "Efficient capital markets: A review of theory and empirical work", in Cochrane, J.H. and Moskowitz, T.J. (Eds.), Selected Papers of Eugene F. Fama, University of Chicago Press, Chicago, IL, USA, pp. 76-121. ISBN 9780226426983.

[2] Fama, E.F. (1965), "The behavior of stock-market prices", Journal of Business, Vol. 38, pp. 34-105.

[3] Kim, Y., Jeong, S.R. and Ghani, I. (2014), "Text opinion mining to analyze news for stock market prediction", International Journal of Advances in Soft Computing and its Applications, Vol. 6, No. 1.

[4] Khedr, A.E. and Yaseen, N. (2017), "Predicting stock market behavior using data mining technique and news sentiment analysis", International Journal of Intelligent Systems and Applications, Vol. 9, No. 7, pp. 22-30.

[5] Hagenau, M., Liebmann, M. and Neumann, D. (2013), "Automated news reading: Stock prices prediction based on financial news using context-capturing features", Decision Support Systems, Vol. 55, No. 3, pp. 685-697.

[6] Joshi, K., Rao, J. and Bharathi, H.N. (2016), "Stock trend prediction using news sentiment analysis", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 8, No. 3, pp. 67-76.

[7] Kaya, M.Y. and Karsligil, M.E. (2010), "Stock price prediction using financial news articles", in Proceedings of the 2nd International Conference on Information and Financial Engineering, Chongqing, China, IEEE, pp. 478-482.

[8] Selimi, M. and Besimi, A. (2019), "A proposed model for stock price prediction based on financial news", in Proceedings of the ENTRENOVA - ENTerprise REsearch InNOVAtion Conference, Rovinj, Croatia, pp. 100-107.

[9] Zubair, S.K.J.C. (2015), "Extracting news sentiment and establishing its relationship with the S&P 500 index", 48th Hawaii International Conference on System Sciences.

[10] Walter, W., Ho, K., Liu, W.R. and Tracy, K. (2013), "The relation between news events and stock price jump: An analysis based on neural network", 20th International Congress on Modelling and Simulation, Adelaide, Australia, 1-6 December 2013, pp. 1-6.

[11] Bing, L.I. and Ou, C. (2014), "Public sentiment analysis in Twitter data for prediction of a company's stock price movements", IEEE 11th International Conference on E-business Engineering.

[12] Eru Cakra, B.D.T. (2015), "Stock price prediction using linear regression based on sentiment analysis", International Conference on Advanced Computer Science and Information Systems, pp. 147-154.

[13] Y. Shynkevich, T. M. McGinnity, S. Coleman and A. Belatreche, "Stock price prediction based on stock-specific and sub-industry-specific news articles," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-8,.

[14] Umbarkar, S.S. and Nandgaonkar, S.S. (2015), "Using association rule mining: Stock market events prediction from financial news", International Journal of Science and Research (IJSR), Vol. 4, No. 6, pp. 2319-7064.

[15] Hoang Thanh, P.M. (2014), "Stock market trend prediction based on text mining of corporate web and time series data", Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 18, No. 1.

[16] Price, S.M., Shriwas, J. and Farzana, S. (2018), "Using text mining and rule-based technique for prediction of Stock Market Price ", International Journal of Emerging Technologies and Advanced Engineering, Vol. 4, No. 1.

[17] Desai, R. and Gandhi, S. (2014), "Stock market prediction using data mining", International Journal of Engineering Development and Research, Vol. 2, No. 2, pp. 2780-2784.

[18] Seker, S.E., Mert, C., Al-Naami, K., Ozalp, N. and Ayan, U. (2014), "Time series analysis on stock market for text mining correlation of economy news", International Journal of Social Sciences and Humanity Studies, Vol. 6, No. 1.

[19] Kim, Y., Jeong, S.R. and Ghani, I. (2014), "Text opinion mining to analyze news for stock market prediction", International Journal of Advanced Soft Computing and Applications, Vol. 6, No. 1.

[20] Abdullah, S.S., Rahaman, M.S. and Rahman, M.S. (2018), "Analysis of stock market using text mining and natural language processing", 2018 International Conference on Informatics, Electronics and Vision (ICIEV), pp. 1-6.

[21] Yeh, I.C., Tsai, W.L. and Li, C.W. (2019), "Forecasting stock prices using a hybrid machine learning approach", Applied Soft Computing, Vol. 78, pp. 277-285.

[22] Basak, Suryoday & Kar, Saibal & Saha, Snehanshu & Khaidem, Luckyson & Dey, Sudeepa Roy, 2019. "Predicting the direction of stock market prices using tree-based classifiers," The North American Journal of Economics and Finance, Elsevier, vol. 47(C), pages 552-567.

[23] Kara, Y., Boyacioglu, M. and Baykan, O. (2011), "Predicting direction of stock price index movement using artificial NNs and support vector machines: The sample of the Istanbul Stock Exchange", Expert Systems with Applications, Vol. 38, pp. 5311-5319.

[24] Huang, C.F. (2012), "A hybrid stock selection model using genetic algorithms and support vector regression", Applied Soft Computing, Vol. 12, pp. 807-818.

[25] Hu, M., Tang, Z., Xie, X. and Jiang, M. (2022), "Stock prediction and analysis based on support vector machine", Frontiers in Business, Economics and Management, Vol. 5, No. 2.

[26] Klein, Tony and Walther, Thomas, Oil Price Volatility Forecast with Mixture Memory GARCH (April 1, 2016). Energy Economics, Vol. 58, 2016.

[27] Ahn, H. and Han, I. (2017), "Credit scoring model based on support vector machines with hybrid feature selection using genetic algorithm and decision tree", Applied Soft Computing, Vol. 54, pp. 327-336.

[28] Gupta, A., Bhatia, P., Dave, K. and Jain, P. (2019), "Stock market prediction using data mining techniques", 2nd International Conference on Advances in Science & Technology (ICAST) 2019, K J Somaiya Institute of Engineering & Information Technology, Mumbai, India, 8-9 April 2019.

[29] Alshammari, B.M., Aldhmour, F., AlQenaei, Z.M. and Almohri, H. (2022), "Stock market prediction by applying big data mining", Arab Gulf Journal of Scientific Research, Vol. 40 No. 2, pp. 139-152.

[30] Dow, C.H. (1900), "Dow theory explained", The Wall Street Journal, Dow Jones & Company, New York, NY, USA.

[31] Edwards, R.D. and Magee, J. (1948), Technical Analysis of Stock Trends, Harper & Brothers, New York, NY, USA.

[32] Kendall, M.G. (1953), "The analysis of economic time-series, Part I: Prices", Journal of the Royal Statistical Society, Vol. 116, pp. 11-34.

[33] Bhandari, H.N., Rimal, B., Pokhrel, N.R., Rimal, R., Dahal, K.R. and Khatri, R.K.C. (2022), "Predicting stock market index using LSTM", Machine Learning with Applications, Vol. 9, 100320.

[34] Agrawal, M., Khan, A.U. and Shukla, P.K. (2019), "Stock price prediction using technical indicators: A predictive model using optimal deep learning", International Journal of Recent Technology and Engineering (IJRTE), Vol. 8 No. 2, pp. 2277-3878.

[35] Chang, V., Xu, Q.A., Chidozie, A., & Wang, H. (2024). Predicting economic trends and stock market prices with deep learning and advanced machine learning techniques. Electronics, Volume 13, Issue 17, 3396.

[36] Sangeetha, J. M., & Alfia, K. J. (2024), "Financial stock market forecast using evaluated linear regression-based machine learning technique", Measurement: Sensors, Vol. 31, Article 100950, ISSN 2665-9174.

[37] Yao, D., & Yan, K. (2024). "Time series forecasting of stock market indices based on DLWR-LSTM model," Finance Research Letters, Vol. 68, Article 105821.

[38] Cristescu, M.P., Mara, D.A., Nerisanu, R.A., Culda, L.C., & Maniu, I. (2023). "Analyzing the impact of financial news sentiments on stock prices—A wavelet correlation," Mathematics, 11, Article 4830.

[39] Costola, M., Hinz, O., Nofer, M., & Pelizzon, L. (2023). "Machine learning sentiment analysis, COVID-19 news and stock market reactions," Research in International Business and Finance, Vol. 64, Article 101717.

[40] Tsai, P.-F., Gao, C.-H., & Yuan, S.-M. (2023). "Stock Selection Using Machine Learning Based on Financial Ratios," Mathematics, Vol. 11, Article 4758.

[41] Kim, W., Jeon, J., Jang, M., Kim, S., Lee, H., Yoo, S., & Ahn, J. (2024). "Developing a Dynamic Feature Selection System (DFSS) for Stock Market Prediction: Application to the Korean Industry Sectors," Applied Sciences, Vol. 14, Article 7314.

[42] Sunantha, P. (2020). "Forecasting the Changes in Daily Stock Prices in Shanghai Stock Exchange Using Neural Network and Ordinary Least Squares Regression," Investment Management and Financial Innovations, 17(3), 292-307.

[43] Jabeur, S.B., Mefteh-Wali, S., & Viviani, J.L. (2024). "Forecasting gold price with the XGBoost algorithm and SHAP interaction values," Annals of Operations Research, 334, 679–699.

[44] Baker, S. R., Bloom, N., & Davis, S. J. (2016). "Measuring Economic Policy Uncertainty," The Quarterly Journal of Economics, 131(4), 1593–1636.

[45] Christiano, L. J., Motto, R., & Rostagno, M. (2014). "Risk Shocks," American Economic Review, 104(1), 27–65.

[46] Damodaran, A. (2012). Investment Valuation: Tools and Techniques for Determining the Value of Any Asset. John Wiley & Sons.

[47] Dees, S., & Güntner, J. (2014). "Analysing and Forecasting Price Dynamics across Euro Area Countries and Sectors," Journal of Forecasting, 33(1), 42–60.

[48] Fama, E. F. (1986). "Term Premiums and Default Premiums in Debt Markets," Journal of Financial Economics, 17(2), 175–196.

[49] Gulen, H., & Ion, M. (2016). "Policy Uncertainty and Corporate Investment," Review of Financial Studies, 29(3), 523–564.

[50] Ludvigson, S. C. (2004). "Consumer Confidence and Consumer Spending," Journal of Economic Perspectives, 18(2), 29–50.

[51] Boyd, J. H., Hu, J., & Jagannathan, R. (2005). "The Stock Market's Reaction to Unemployment News: Why Bad News Is Usually Good for Stocks," Journal of Finance, 60(2), 649–672.

[52] OECD. (2021). OECD Business Confidence Index, OECD Economic Outlook.

[53] Murphy, J.J. (1999). "Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications," New York Institute of Finance.

[54] Damodaran, A. (2012). "Investment Valuation: Tools and Techniques for Determining the Value of Any Asset," John Wiley & Sons.

[55] Investopedia Team. (2024). "What Does the S&P 500 Index Measure and How Is It Calculated?" *Investopedia*, 21 July 2024, www.investopedia.com.

[56] Investopedia Team. (2024). " What Is the History of the S&P 500 Stock Index?" *Investopedia*, 13 April 2023, www.investopedia.com.

[57] Investopedia Team. (2024). " Purchasing Managers' Index (PMI) Definition and How It Works," *Investopedia*, 08 May 2023, www.investopedia.com.

[58] G. Papageorgiou, D. Gkaimanis, C. Tjortjis, "Enhancing Stock Market Forecasts with Double Deep Q-Network in Volatile Stock Market Environments", Electronics, 13(9), 1629; MDPI, 2024.

[59] P. Koukaras, C. Nousi and C. Tjortjis, "Stock Market Prediction Using Microblogging Sentiment Analysis and Machine Learning", Telecom, MDPI, 3(2), 358-378, 2022.

[60] P. Koukaras, V. Tsichli, and C. Tjortjis, “Predicting Stock Market Movements with Social Media and Machine Learning”, Proc. 17th Int’l Conf. on Web Information Systems and Technologies (WEBIST 21), 2021.

[61] C. Nousi and C. Tjortjis, “A Methodology for Stock Movement Prediction Using Sentiment Analysis on Twitter and StockTwits Data”, Proc. 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM 2021), 2021.

[62] Yahoo Finance, “S&P 500 Historical Data,” Retrieved from https://finance.yahoo.com.

[63] Hutto, C.J., & Gilbert, E. (2014). "VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text," Proceedings of the Eighth International Conference on Weblogs and Social Media, 216–225.

[64] Federal Reserve Bank of St. Louis. Economic data categories. Federal Reserve Economic Data (FRED). Retrieved from https://fred.stlouisfed.org/categories.

[65] Organisation for Economic Co-operation and Development (OECD). Business confidence index (BCI). Retrieved from https://www.oecd.org/en/data/indicators/business-confidence-index-bci.html

[66] Hugging Face, “distilbert-base-uncased-finetuned-sst-2-english,” Retrieved from https://huggingface.co/distilbert/distilbert-base-uncased-finetuned-sst-2-english.

[67] Loria, S., “TextBlob: Simplified Text Processing,” Retrieved from https://textblob.readthedocs.io.

[68] Scikit-learn, “RandomizedSearchCV: Randomized Parameter Optimization,” Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html.

[69] Scikit-learn, “GridSearchCV: Exhaustive Parameter Search,” Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

[70] Géron, A., “Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow,” 2nd ed., O’Reilly Media, 2019.

[71] Chen, T., & Guestrin, C. (2016). "XGBoost: A Scalable Tree Boosting System," Proceedings of the 22nd ACM SIGKDD International Conference on

Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785

[72] Jolliffe, I. T. (2002). "Principal Component Analysis" (2nd ed.). Springer-Verlag.

Appendix A

Term	Acronym
Machine Learning	ML
Recursive Feature Elimination	RFE
Linear Regression	LR
Random Forest	RF
Gradient Boosting	GB
Multi-Layer Perceptron	MLP
Mean Absolut Error	MAE
Mean Squared Error	MSE
R-squared	R^2
Exponential Moving Average	EMA
Business Confidence Index	BCI
Consumer Sentiment Index	CEI
Efficient Market Hypothesis	EMH
Technical Analysis	TA
Natural Language Processing	NLP
Neural Networks	NN
ISM Manufacturing PMI	PMI
Equity Market Uncertainty Index	EMUI
Economic Policy Uncertainty Index	EPUI
Relative Strength Index	RSI
Moving Average Convergence Divergence	MACD
Hugging Face	HF

Appendix B

Table 1

```
### RSI ###

1. rsi_indicator = ta.momentum.RSIIndicator(close=stock_index['Adj Close'], window=50,
fillna=True)
2. stock_index['RSI'] = rsi_indicator.rsi()
```

```
### Stochastic Oscillator ###

1. stoch_osc = ta.momentum.StochasticOscillator(
2.     high=stock_index['High'],
3.     low=stock_index['Low'],
4.     close=stock_index['Adj Close'],
5.     window=50,
6.     smooth_window=3,
7.     fillna=True)
```

```
8. stock_index['stoch_k'] = stoch_osc.stoch()
9. stock_index['stoch_d'] = stoch_osc.stoch_signal
```

```
                              ### Williams %R Indicator ###

1. williams_r = ta.momentum.WilliamsRIndicator(
2.     high=stock_index['High'],
3.     low=stock_index['Low'],
4.     close=stock_index['Adj Close'],
5.     lbp=14,
6.     fillna=True)
7. stock_index['williams_r'] = williams_r.williams_r()
```

```
                           ### Exponential Moving Average ###

1. ema = ta.trend.EMAIndicator(
2.     close=stock_index['Adj Close'],
3.     window=14,
4.     fillna=True)
5. stock_index['ema_14'] = ema.ema_indicator()
```

```
                                ### MACD Indicator ###

1. macd = ta.trend.MACD(
2.     close=stock_index['Adj Close'],
3.     window_slow=26,
4.     window_fast=12,
5.     window_sign=9,
6.     fillna=True)
7. stock_index['macd_line'] = macd.macd()
8. stock_index['macd_signal'] = macd.macd_signal()
9. stock_index['macd_diff'] = macd.macd_diff()
```

Table 2

```
                                   ### TextBlob ###

1. def textblob_sentiment(text):
2.     if isinstance(text, str):
3.         analysis = TextBlob(text)
4.         return analysis.sentiment.polarity
5.     else:
6.         return 0.0
7. # Apply sentiment analysis to each headline (Top1 to Top25 columns)
8. headline_columns = [f'Top{i}' for i in range(1, 26)]
9.
10. # For each headline, calculate the sentiment score
11. for column in headline_columns:
12.     news[column + '_sentiment'] = news[column].apply(textblob_sentiment)
13. news['overall_sentiment'] = news[[col + '_sentiment' for col in
headline_columns]].mean(axis=1)
```

```
                                  ### Hugging Face ###

1. sentiment_pipeline = pipeline("sentiment-analysis")
2. def get_sentiment(row):
3.     sentiments = []
4.     for headline in row[1:]:  # Exclude 'Date' column
5.         if isinstance(headline, str):  # Check if the headline is a string
6.             sentiment = sentiment_pipeline(headline)[0]
7.             sentiments.append(sentiment)
8.         else:
9.             sentiments.append({'label': 'NEUTRAL', 'score': 0})  # Handle non-string
values
10.     return sentiments
11.
12. news['Sentiments'] = news.iloc[:, 1:].apply(get_sentiment, axis=1)  # Skip 'Date'
column for sentiment
```

```
# Function to calculate daily average sentiment score
def calculate_daily_avg_score(sentiment_list):
    scores = []
    for sentiment in sentiment_list:
        score = sentiment['score'] if sentiment['label'] == 'POSITIVE' else -
sentiment['score']
        scores.append(score)
    return sum(scores) / len(scores) if scores else 0
news['Daily_Avg_Sentiment_Score'] =
news['Sentiments'].apply(calculate_daily_avg_score)
```

Table 3

```
                                    ### RFE ###
model = LinearRegression()
rfe = RFE(model, n_features_to_select=10)
X_rfe = rfe.fit_transform(X, y)
ranking = rfe.ranking_
features = X.columns
rfe_results = pd.DataFrame({
    'Feature': features,
    'Ranking': ranking}).sort_values(by='Ranking')
```