Integrating Sentiment Analysis and Machine Learning for Robust Stock Price Prediction: A Comprehensive Study

Exploring the Synergy of Sentiment Data and Historical Stock Prices for Accurate and Reliable Market Predictions

Thesis by

[Anup Wilson]

Under the Guidance of

[Pierpaolo Dondio]

In Partial Fulfillment of the Requirements for the

Degree of

[Master of Science]

DUBLIN BUSINESS SCHOOL

Dublin, Ireland

[2024]

Defended [January 8, 2024]

DECLARATION

I, Anup Wilson, formally declare that the work presented in this thesis, "Integrating Sentiment Analysis and Machine Learning for Robust Stock Price Prediction: A Comprehensive Study," is solely my own. The study I did, the approaches I used, and the conclusions I came to are all a reflection of my independent academic work in Data Analytics.

I additionally confirm that all outside sources of data and concepts used in this thesis have been properly cited and referenced. The acknowledgments section contains the proper acknowledgement of any support received during the study process.

I acknowledge the value of maintaining academic integrity and declare that the information in this thesis has not been submitted in whole or in part for consideration toward any other degree or certification at any other university.

Name: Anup Wilson

Student ID: 10628141

Date: 8th January 2024

ACKNOWLEDGMENT

I would like to sincerely thank Pierpaolo Dondio, my supervisor, for his excellent mentorship, persistent support, and important advice over the entire study process. His advice and thoughts were quite helpful in guiding the direction of this thesis.

I want to express my sincere gratitude to my family for their constant encouragement and compassion during this difficult path. Their confidence in my skills has always been a source of motivation.

I am also grateful to my friends, who supported me through the highs and lows of my academic pursuit by offering advice, encouragement, and insights. Their friendship gave this study experience a valuable new aspect.

Finally, I would like to express my gratitude to everyone who helped me to successfully complete this thesis, whether directly or indirectly. Your assistance has been invaluable, and I sincerely appreciate the spirit of cooperation that has enhanced our academic endeavor.

ABSTRACT

The incorporation of sentiment analysis from Twitter data into stock price prediction models for well-known electric vehicle (EV) manufacturers, such as Tesla, Ford, Nio, and Xpeng, is investigated in this study. Sentiment scores are collected from tweets using natural language processing algorithms, and they are classified as positive, neutral, or negative. Subsequently, these sentiment scores are combined with historical stock price data to train various machine learning models, such as neural network, support vector machine (SVM), decision tree, random forest, MLP Regressor and linear regression models. The accuracy of each model is evaluated using metrics like mean squared error (MSE), mean absolute error (MAE), and R-squared value. The findings demonstrate how sentiment research may improve stock price predictions, with random forest models continuously beating other models. This indicates the ability to capture hidden nonlinear relationships between sentiment and stock price. The study highlights the growing significance of sentiment analysis in financial analysis and investment decision-making, not only in the electric vehicle industry but also providing investors with important market conditions information. The results add to the expanding amount of literature on sentiment analysis applications in finance, especially in the fast-paced electric vehicle (EV) sector. These applications may enhance overall stock market efficiency and provide guidance for strategic investment choices.

Keywords: Sentiment Analysis, Stock Price Prediction, Electric Vehicle (EV) Industry, Machine Learning Models, Random Forest, Investment Decision-making, Financial Analysis, Market Conditions.

Table of Contents

Table of Figures

CHAPTER 1

INTRODUCTION

1.1 Background

The stock market is a complex and ever-changing environment that is influenced by several variables such as economic data, company performance, and investor state of mind. Although stock price predictions have long been made using traditional financial analysis techniques, the rise of social media has opened a new channel for evaluating investor sentiment and may provide insightful information for stock market predictions.

Understanding public opinion and evaluating emotional reactions may be effectively achieved using sentiment analysis, which is the process of extracting and evaluating subjective information from text. Social media platforms, such as Twitter, have become a treasure trove of public sentiment, with millions of users expressing their thoughts and opinions on various topics, including the stock market. By analyzing the sentiment of tweets related to a particular company, we can gain valuable insights into investor perceptions and potentially predict future stock prices.

This thesis investigates the use of sentiment analysis in predicting the stock prices of electric vehicle (EV) companies. EVs have attracted the attention of investors, with Tesla leading the trend, followed by Ford, Nio, and Xpeng. The study uses data from these four companies, collecting sentiment from their own Twitter feeds. Tweets are categorized using sentiment analysis algorithms into three categories: positive, neutral, and negative. Sentiment scores are then calculated to determine how each company's sentiment is overall.

The obtained sentiment scores are then combined with historical stock price data to train various machine learning models. To investigate the relationship between sentiment and stock prices, methods such as linear regression, random forest, decision tree, support vector machine (SVM), and neural network are used. The accuracy of each model is evaluated based on the results of the analysis using metrics like mean squared error (MSE), mean absolute error (MAE), and R-Squared value.

The findings of this study indicate that sentiment analysis can be a useful method for predicting the stock prices of EV companies. Random forest models outperform other models on a consistent basis, suggesting their greater capacity to capture complicated nonlinear correlations between sentiment and market prices. It suggests that integrating sentiment analyses to regular financial evaluation approaches can enhance stock price prediction accuracy significantly.

The effects of this research extend beyond electric vehicle companies. Sentiment analysis is set to become more significant in stock market analysis and investing decision-making as social media keeps influencing the financial environment. Investors can obtain a greater understanding of the market's conditions and make good investment decisions by knowing the opinions of analysts, investors, and the public.

This thesis contributes to the increasing amount of research on sentiment analysis and its applications in finance. It offers valuable insight into whether sentiment research can predict the price of stocks, especially in the rapidly growing and dynamic electric vehicle (EV) industry. The study's conclusions could improve the stock market's overall efficiency and guide investing plans.

1.2 Research Project.

RQ1. Is it possible to effectively integrate sentiment analysis from Twitter data into stock price prediction models for major EV manufacturers?

RQ2. What is the performance of various machine learning models (e.g., Random Forest, Support Vector Machines,) in predicting stock prices using sentiment analysis and historical stock price data?

RQ3. Which machine learning model (e.g., Random Forest, Support Vector Machines) performs best in predicting EV stock values utilizing a combined sentiment analysis and historical stock price data approach?

RQ4. When compared to models that only use historical stock price data, will the integration of sentiment analysis using Twitter data considerably improve the prediction accuracy of stock prices for key electric vehicle manufacturers?

RQ5. How much can sentiment analysis from social media sites like Twitter be used to enhance stock market performance and guide smart investment choices for the electric vehicle (EV) sector?

RQ7. Does the use of cross-validation enhance the accuracy and usability of sentiment-based stock price prediction models? If so, what effect does the model's performance have on the cross-validation technique selected?

RQ8. Which feature selection techniques make the most use of sentiment data to improve stock price prediction models' accuracy?

1.3 Research Objectives

RO1. To find out whether it is possible to use sentiment analysis from Twitter data into stock price prediction models for major EV manufacturers, like Tesla, Ford, Nio, and Xpeng.

RO2. To use sentiment analysis and historical stock price data to assess how well different machine learning models predict stock prices.

RO3. To evaluate the performance of different machine learning models in predicting stock prices using sentiment analysis and historical stock price data.

RO4. To identify the most effective machine learning model for predicting stock prices in the electric vehicle (EV) industry utilizing sentiment analysis and historical stock price data.

RO5. To determine whether sentiment analysis may be used to improve stock market efficiency and offer advice on strategic investment decisions.

RO6. To examine the impact of cross-validation on the accuracy and generalizability of sentiment-based stock price prediction models.

RO7. Examine how the feature selection process affects the models' capacity to represent nonlinear correlations between sentiment and stock price.

RO8. To assess the efficiency of various feature selection strategies in improving the accuracy of sentiment-based stock price prediction models.

1.4 Scope and Limitations.

1.4.1 Scope

The study's findings shed insight on how sentiment analysis might improve stock price prediction, especially in the volatile EV market. The results seek to contribute to a wider knowledge of sentiment analysis applications in finance. The focus on key EV manufacturers offers insights into investor views. The study's conclusions, which highlight the importance of sentiment monitoring in financial analysis, may potentially increase overall stock market performance and provide direction for strategic investment decisions.

1.4.2 Limitations

- **Concerns about Overfitting:** The research admits that there may be overfitting in predictive models, which is particularly evident given the high R-squared values that are getting close to 1. A balance between generalization and accuracy may be necessary, which could affect the models' performance.
- **Data Source Limitations:** Sentiment analysis in the study relies on Twitter data, and its effectiveness depends on how diverse and representative the opinions shared on this network are. The generalizability of the model may be impacted by differences in sentiment expression on various social media platforms.
- **Model Dependency:** The assumptions made throughout the modeling process influence how well machine learning models, such as the Random Forest Regressor, perform. The chosen models might not be able to fully represent the complex relationships and patterns present in the data.

- **External Factors:** A variety of external factors, such as economic conditions and world events, influence stock prices. There could be differences in the models' forecasts if these externalities are not fully considered.
- **Evaluation Metrices:** Even if MSE, MAE, and R-squared offer insightful information, not all details of model performance may be captured by them. A more thorough assessment might be provided by additional metrics or additional validation techniques.
- **Dynamic Nature of Stock Market:** Since the stock market is constantly dynamic, changes in the market environment could affect the study's conclusions. As market trends change, predictive models might need to be adjusted on a regular basis.

CHAPTER 2

LITERATURE SURVEY

The efficiency of the stock market has been a long-standing debate among economists and financial experts. The relationship between implied stock prices generated by the Black-Scholes option pricing model and actual stock prices to assess market efficiency. The market is efficient, as implied and actual stock prices closely correspond. However, a simulated trading strategy based on the largest discrepancies between actual and implied prices does demonstrate some success, hinting at potential profitable opportunities. The importance of careful data analysis and consideration of timing differences in studying the relationship between option prices and stock prices(Brown and Shevlin, 2001). In a 10-day forecast period, neural networks have shown positive results in stock price prediction, reaching high accuracy of up to 90%. To enable its practical application for certain prediction tasks, neural networks' theoretical underpinnings must yet be strengthened. Although ADALINE-like algorithms have been utilized in statistical prediction, their overall understanding is limited by the lack of efficient neural network design approaches. Complex neural network analysis has been made easier by recent developments in neurocomputers and computer hardware, but more testing and methodological improvements are still required to fully realize the potential of neural networks in stock price prediction(Schöneburg, 1990). To increase the accuracy of stock price predictions, a neuro-genetic stock prediction system makes use of financial correlation between companies. To choose informative input features to analyze the correlation between companies, a recurrent neural network is used. The buy-and-hold strategy and other conventional approaches are outperformed by the system. To enhance recurrent neural networks' stock trading performance, a feature-selection genetic method is developed. Compared to standard strategies, the algorithm performs significantly better by generating potential input variables from other organizations that show correlation. To forecast stock values, a hybrid genetic algorithm combines recurrent neural networks with genetic algorithms. Using financial correlation to identify important input features, this algorithm enhances prediction accuracy significantly. These findings show that using neuro-genetic techniques can increase stock price predictions by leveraging financial correlations (Kwon et al., 2005). The integration of Twitter as an essential data source for predicting and modelling the movements of the stock market by extracting information from tweets about both facts and opinions. Support Vector Regression (SVR) and natural language processing methods showed positive results in present and future stock price prediction. Twitter's ability as a quick source of information demonstrates its potential uses outside of stock market analysis, like anticipating the spread of pandemics and serving as an earthquake early warning system. Twitter is the best social media channel for real-value price prediction according to comparisons with other sources. the field of stock market prediction by utilizing text mining and natural language processing (NLP) tools to improve understanding of the complex relationship between sentiment on social media and stock market activity (Sebastian and Wolfram, 2010). The potential of predicting stock market trends through sentiment analysis on Twitter. An algorithm for data mining was created that included sentiment analysis methods, and it was able to predict stock prices for a selected group of companies with an amazing average accuracy of 76.12%. Using this method, it was possible to find a significant relationship between actual stock price fluctuations and Twitter sentiment, especially for companies that are listed on the NASDAQ and the New York Stock Exchange. Additionally, using undetermined textual twitter data, natural language processing (NLP) was used to extract and describe public sentiment. This analysis highlighted the differences in prediction between different companies and highlighted the promise of using Twitter data to forecast stock market changes. It also recommended a three-day delay for assessing the effectiveness of event management on social media in predicting stock movements (Bing et al., 2014). The complex relationship between market movements and public sentiment as expressed on Twitter, with a focus on stock price dynamics. To find connections between Twitter sentiment and stock market volatility, the methodology makes use of supervised machine learning and sentiment analysis. Tweets are precisely and highly accurately classified as positive negative, or neutral by a sentiment analyser. Complex understandings of the relationship between sentiment and stock prices are obtained using two different text representations: Word2vec and Ngram. The analysis shows a strong relationship between the attitude indicated in Twitter messages and changes in stock prices, both positively and negatively. Unfortunately, Word2vec's inability to be dynamically optimized at times when certain tasks must be completed is one of its drawbacks. Since Word2vec is a static model, Ngram is mainly only helpful in situations when a large amount of data is available. The model may not always be able to capture every occurrence in features, despite having an extensive data set (Sasank Pagolu et al., 2016). the application of Support Vector Machines (SVM) to compute sentiment scores for tweets taken from Stock Twits, and the integration of sentiment analysis on social media, specifically Twitter, in predicting stock price changes. These sentiment scores are used together with market index data to build an SVM model that predicts the movement of stocks for the next day with an excellent 76.65% accuracy. the direct effect of news perception on stock prices, highlighting the strong influence of public opinion on financial market decision-making. Sentiment analysis is a useful technique that gives investors information about the relationship between sentiment and stock prices. The potential of sentiment analysis is a helpful tool for investor decision-making, especially when it comes to anticipating stock movements. 1.7 million newspaper stories from The Guardian between 2000 and 2016 are used to examine how sentiment analysis affects stock market patterns. Sentiment dictionaries are used to classify the sentiment of different stock market indexes. (Bhandari, 2017) notes the difficulties in accurately extracting sentiment indicators from digital documents and highlights the complex nature of sentiment analysis and natural language processing in relation to stock forecasts. He also recommends testing several models and using a larger training dataset to increase prediction accuracy. In addition, it explores sophisticated techniques like Bivariate Granger Causality Analysis and the Self-Organizing Fuzzy Neural Network model, as well as the application of lexical dictionary-based methodologies for sentiment categorization. (Cristescu et al., 2022) the importance of correctly assessing the movement of the stock market, especially in times of crisis, and how sentiment analysis might improve regression models used for market forecasting. He highlights the influence of investor opinions on financial markets and claims for the addition of sentiment analysis as an independent feature in regression models. It focuses on the weaknesses of investment risk-management models in addressing systemic risks during crises. Evaluations show that when sentiment analysis is added, linear autoregression models perform better in terms of goodness of fit, but polynomial autoregressions—more specifically, quadratic and cubic models—perform better than linear models. (Cristescu et al., 2022) offers insightful information on the relationship between stock prices and news opinions, indicating that news reports could align with market Application of sentiment analysis to identify patterns in customer reviews and predict stock market movement. The main objective is to improve prediction performance by critically reviewing current issues in sentiment analysis-based stock market prediction. The approach is divided into sections that explain various prediction methodologies. which methodically examines a variety of statistical and econometric approaches used for stock market prediction with sentiment analysis. This also highlights the impact of classification accuracy on prediction performance and offers insightful information about the dependability of stock market indicators. This approach aims to improve our understanding of stock market behaviour through investigating the influence of customer reviews, it also suggests employing sentiment analysis techniques to increase the accuracy of predictions. This survey lays the groundwork for future developments in the field by providing a thorough overview of the state of sentiment analysis in stock market prediction (Rajendiran and Priyadarsini, 2023). Investigation of the correlation between news reports and changes in the stock market by the application of sentiment classification to financial news stories. (Agarwal et al., 2023) intends to enhance information regarding the critical role of news sentiment in anticipating stock market trends and highlights the importance of sentiment analysis in the field of stock price prediction, with a particular focus on projecting a company's future stock trend. He carefully examines the effectiveness of several algorithms, such as Count Vectorizer, TF-IDF Vectorizer, and Naive Bayes, in sentiment analysis of product evaluations using an Amazon dataset by utilizing data mining and artificial intelligence techniques. This approach meticulously preprocesses data to eliminate noise, transforms textual information into numerical features through vectorization techniques, and applies classifiers like Naive Bayes to predict sentiment. Approach revealed that both the Random Forest algorithm with Count Vectorizer and the Naive Bayes algorithm outperformed the Random Forest algorithm with TF-IDF Vectorizer when applied to news articles. (Agarwal et al., 2023) evaluates algorithm effectiveness using metrics such as accuracy, precision, recall, and F1-score, visualizing results through confusion matrices.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Data Flow Cycle.

The 1999 publication represented the release of the first version of the Cross-Industry Standard Process for data processing. According to numerous studies, it is still the most extensively used analytic methodology today. Because electronic devices and sensors are so widely available, the number of data processing projects that are available has increased significantly. Over the past two decades, data has been commercialized in numerous ways due to an abundance of new applications and business models. The area of extracting value from data has grown significantly in size and complexity while also becoming considerably more exploratory under the oversight of knowledge research (Martinez-Plumed et al., 2021). Strict adherence to project procedures may prove challenging for teams engaged in data science projects. Process models such as the Analytics Life Cycle and CRISPDM can benefit from and be enhanced by agile approaches. In current data science projects, process models, despite their long history, are rarely extensively utilized (Schröer et al., 2021). Under the discipline of data science, the topic of extracting the most value from data has become much more exploratory while simultaneously increasing in volume and complexity. Whereas the traditional data mining process starts with stated business goals that translate into a specific data mining activity that finally turns "data into knowledge," in the latter case, data-driven and expertise-driven phases interact. Said another way, the methods for extracting value from data have evolved to reflect the changing characteristics of data. CRISP-DM is a data mining process model that appears to be independent of industry. As shown in Figure 1, there are six iterative stages from business knowledge to its operation.

Drawing exclusively from the CRISP-DM, Figure provides a concise overview of the key idea, assignments, and results of each stage.

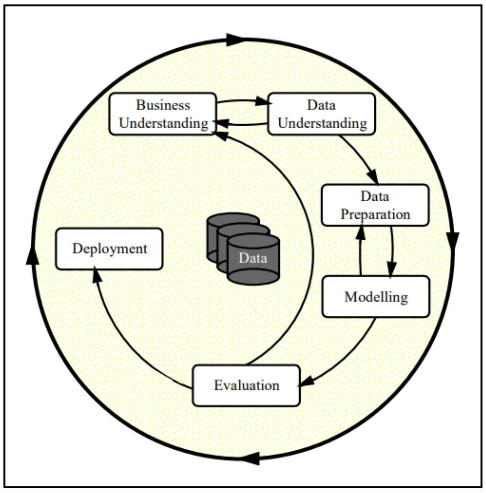


Figure 1 Phases of Crisp-DM Methodology data flow cycle. (Wirth and Hipp, 2000)

Step 1: Business Understanding.

The primary objective of the business understanding phase is to comprehend the goals as well as requirements of the work being done. The remaining three tasks in this phase, except for task number three, are basic project management tasks that are relevant for a wide range of projects.

- **The business's goals are:** The aim of the company in our set situation is to develop a stock price prediction model that can accurately identify stock price for the EV industry by utilizing past stock data and the sentiment of tweets.
- **Accessing:** Conduct a cost-benefit analysis and determine the resources that are available, the project's needs, risks, and backup plans.
- **Data mining**: what being successful suggests in terms of both business objectives and technology data mining.
- **Project Plan:** Identify methods and instruments and develop detailed plans for each project phase.

Step 2: Data Understanding.

Data understanding is the next stage. identifying, collecting, and evaluating data sets that might help you accomplish the business objective is the focus here, as it also strengthens the foundation of business expertise. This phase comprises the following four tasks:

- **Getting data:** Gathering the information required and using the right tools for data analysis.
- **Data understanding:** Examine the data and make a note of its surface attributes, like fields, record counts, and data types.
- **Data exploration:** A more thorough examination of the data. The data may be discovered, examined and correlations between them can be discovered.
- **Quality:** To what extent is the data impure or pure? Observe any issues with quality.

Step 3: Preparing the Data.

In data mining, data preparation is one of the most important and time-consuming tasks. It's estimated that data preparation makes up between 50 and 70 percent of the time and work required for a project. Although this cost can be decreased by investing sufficient time and effort to the first stages of learning about company and data, much work would still need to be done to prepare the data for mining.

Step 4: Modelling the data.

Modelling is usually done in several iterations. Usually, data miners run many models with the default parameters and then adjust them or go back to the data preparation phase to make any adjustments required by their preferred model. Often, an organization cannot effectively solve a data mining problem with a single model and one implementation. The fact that there are several approaches you may use to investigate a given problem is what makes data mining interesting.

Step 5: Evaluation.

Make sure the models developed during the modelling phase meet the previously specified technical as well as practical requirements outlined in the data mining success criteria. However, before proceeding, you should examine the results of your efforts using the project's initial set of business objectives. This is necessary to guarantee that your business can use the insights you have produced. There are two types of results from data mining:

- The final CRISP-DM models selected in the first round.
- Any conclusions or discoveries that come from the models themselves and the data mining procedure. We refer to them as discoveries.

Step 6: Deployment

Deployment is the process for applying what you have learned to make improvements to your business. Another definition of deployment is the process of implementing changes throughout your company by use of data mining findings. These results will surely be useful for planning and decision-making around marketing, even if they aren't officially integrated into your information systems. There are two main types of activities that make up the CRISP-DM deployment phase:

- Preparing and tracking the outcomes of the application.
- Completing tasks, such as completing a project evaluation and creating a final report.

3.2 Methodology

1. **Twitter Data Collection and Preprocessing.**

The first stage of the data flow cycle involves collecting Twitter data. This can be easily done by utilizing the Twitter API, which provides real-time access to essential Twitter data. Following the data collecting process, the most important stages are data cleaning and processing. This involves eliminating any unnecessary or disruptive data with care, resulting in a cleaned dataset. Moreover, an important part of this stage is formatting the data in a way that satisfies the needs of the sentiment prediction model.

2. **Twitter Data Preprocessing.**

Moving forward in the Twitter data processing pipeline, the next task is the necessary preparation of the acquired data. Stemming or lemmatizing the words, removing stop words from the text, and other activities are all included in this crucial step. Reducing the size of the data is the main objective of this preprocessing stage, which will help the sentiment prediction model learn more effectively.

3. Sentiment Labelling

Sentiment labelling plays a critical role in the data flow cycle when using Twitter data for predicting stock prices. Tools like TextBlob and VaderSentiment analyzer can be used to complete this. These tools are useful for classifying tweets into sentiment categories: positive, negative, or neutral. They produce sentiment scores in addition to labels, providing more in-depth understanding of the general sentiment expressed in tweets. In addition to classifying tweets, this sentiment labelling procedure helps to provide a useful dataset that is necessary for training the sentiment prediction model.

4. Exploratory Data Analysis.

Following the preprocessing of Twitter data, an important step is looking into exploratory data analysis (EDA) to gain knowledge about the dataset. To identify patterns and trends, this requires a thorough analysis of the data using a variety of statistical methods and visualization techniques. Gaining an in-depth understanding of the data is the main objective here, since it will enable one to make well-informed decisions in the following stages.

5. Sentiment Model Prediction Training and Evaluation.

The next step is to train a sentiment prediction model. Using various machine learning algorithms, such as logistic regression, Naive Bayes, or support vector machines (SVMs), is required for this stage. The main aim of this model is to predict the sentiment included in a tweet to determine whether it is positive, negative, or neutral. The sentiment prediction model basically aims to identify and categorize the state of mind that each tweet expresses.

6. Collection of Historical Stock Data.

As we go to the following stage, the collection of information regarding stock prices is necessary. Using different financial data sources, such Bloomberg or Yahoo Finance, will help obtain this. The necessary stock price data should include all relevant information, such as Date, Open price, High price, Low price, Close price, Volume, Adj close price.

7. Data preprocessing of Stock data and understanding the stock data.

Preparing the raw financial data for analysis and improving it is the main goal of the preprocessing stage of stock data. To make sure the data is accurate, consistent, and prepared for use in analytical models, there are several necessary steps that must be taken. Managing missing or outlier numbers, standardizing data to a common scale, and resolving any possible timestamp problems are examples of common preprocessing activities. Furthermore, it is crucial for understanding the complex details of stock data. This understanding includes identifying important elements like stock prices, trade volumes, and other relevant information. Moreover, utilizing methods such as exploratory data analysis (EDA) to uncover patterns and trends in the stock data allows for a thorough comprehension of market behaviour and establishes a foundation for wise decision-making in later phases.

8. Merging the dataset.

Next challenge is to combine two datasets, one produced from sentiment predictions and the other from market prices. The ticker symbols and dates that are present in both datasets can be used to carry out this merging operation. In short, combining the data from various disparate sources to provide a single dataset that includes stock price information as well as sentiment-related insights. We create an integrated dataset by comparing relevant dates and symbols, which may provide a more complete view for further analysis and modelling.

9. Feature Selection.

It is critical to identify the most important features from the combined data of historical stock and twitter data to accurately predict stock prices. Feature selection is the process of finding and eliminating features that are unnecessary or unwanted that might affect the prediction model's performance. To find the most suitable set of features for the stock price prediction model, a variety of feature selection strategies was used.

10. Model Building and evaluation

Various regression approaches were used to build effective stock price prediction models, including linear regression, random forest regression, decision tree regression, support vector regression, neural networks, and multilayer perceptron regression. Models were initially built without cross-validation to evaluate their fundamental performance. To assess each model's prediction accuracy, the Mean Absolute Error (MAE), R-squared, and Root Mean Squared Error (RMSE) values were computed.

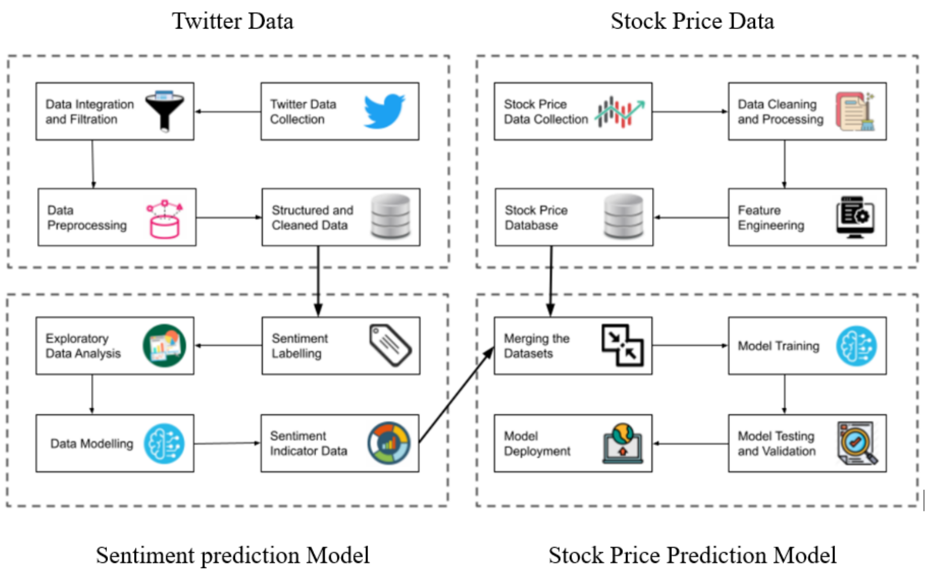

Twitter Data
Stock Price Data
Data Integration and Filtration
Twitter Data Collection
Data Preprocessing
Structured and Cleaned Data
Stock Price Data Collection
Data Cleaning and Processing
Stock Price Database
Feature Engineering
Exploratory Data Analysis
Sentiment Labelling
Data Modelling
Sentiment Indicator Data
Merging the Datasets
Model Training
Model Deployment
Model Testing and Validation
Sentiment prediction Model
Stock Price Prediction Model

Figure 2 Methodology of stock price prediction model using Stock and sentiment data.

CHAPTER 4

RESULTS, EVALUATION AND DISCUSSION

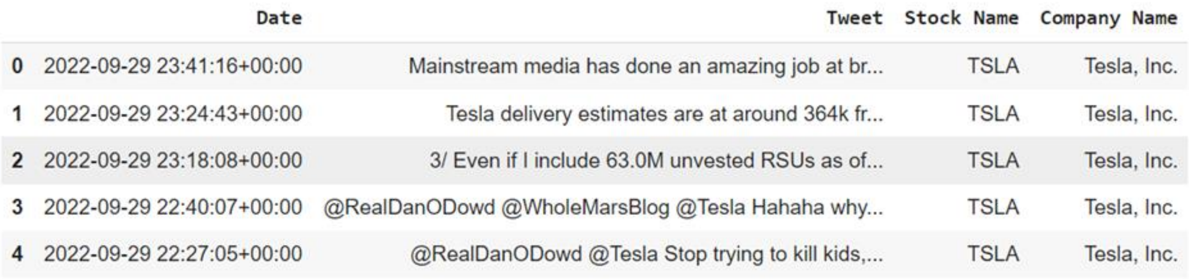
4.1 Twitter data Collection

A comprehensive web scraping operation was conducted to extract tweets from four renowned automobile manufacturers, Tesla, Ford, Nio, and Xpeng. Utilizing Selenium's powerful features, a while loop was created to systematically check Twitter's search results, extracting crucial information like the tweet's body, timestamp, stock ticker symbol, business name, retweets, and likes. Selenium's dynamic nature allows it to adapt to changes in web page structure, ensuring a quick and efficient data collection process.

The data was retrieved iteratively and organized using a systematic methodology. The filtered dataset was saved in a CSV file called "stock_tweets_selected_automobile.csv," protecting data integrity and providing a framework for future investigation, analysis, and visualization. This Selenium-powered web scraping technique is useful for understanding public opinions on stock prices and attitudes towards Tesla, Ford, Nio, and Xpeng brands.

Selenium is a versatile open-source framework used for web application automation, offering cross-platform and cross-browser flexibility. Its client-server architecture allows developers and testers to dynamically navigate online pages, imitate user activities, and interact with web components. Selenium supports multiple programming languages like Java, Python, and C#, handles dynamic web pages, and is compatible with popular browsers like Chrome, Firefox, and Safari. Its flexibility makes it an ideal tool for developing online applications, automation testing, and web scraping. Developers can use Selenium to create scalable and reliable automation scripts for various industries and applications. Using machine learning techniques, the Selenium automation framework automates web scrapping, allowing for autonomous scrapping in later iterations and the utilization of collected data for marketing or customer support applications (Mehta and Pandi (Jain), 2019)

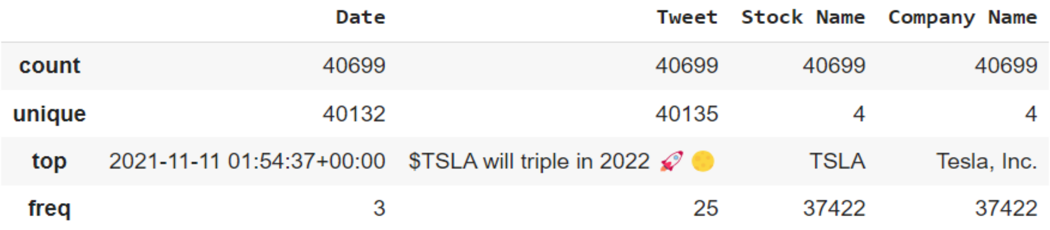
4.2 Data Understanding and EDA for Tweet data.



	Date	Tweet	Stock Name	Company Name
0	2022-09-29 23:41:16+00:00	Mainstream media has done an amazing job at br...	TSLA	Tesla, Inc.
1	2022-09-29 23:24:43+00:00	Tesla delivery estimates are at around 364k fr...	TSLA	Tesla, Inc.
2	2022-09-29 23:18:08+00:00	3/ Even if I include 63.0M unvested RSUs as of...	TSLA	Tesla, Inc.
3	2022-09-29 22:40:07+00:00	@RealDanODowd @WholeMarsBlog @Tesla Hahaha why...	TSLA	Tesla, Inc.
4	2022-09-29 22:27:05+00:00	@RealDanODowd @Tesla Stop trying to kill kids,...	TSLA	Tesla, Inc.

Figure 3 Top 5 records of tweet dataset.

The obtained dataset consists of four different columns that have been carefully organized to offer a comprehensive summary: Date, which includes the tweets' timestamp; Tweet, which includes the textual content of the tweets; Stock Name, which indicates the stock symbol that is associated with the tweet; and company Name, which specifies the name of the company that is connected. With 40,699 rows, the dataset is large and includes a significant amount of sentiment and discussion in real time about the stock market performance of four major players in the automobile industry- Tesla, Ford, Nio, and Xpeng. In addition to ensuring the dataset's truthfulness, this systematic methodology sets the stage for a comprehensive and data-driven investigation of the influence of the social media ecosystem on the stock market movements of these major performers.



	Date	Tweet	Stock Name	Company Name
count	40699	40699	40699	40699
unique	40132	40135	4	4
top	2021-11-11 01:54:37+00:00	$TSLA will triple in 2022 🚀 🌕	TSLA	Tesla, Inc.
freq	3	25	37422	37422

Figure 4 Descriptive Statistics for Tweet Data.

The dataset obtained through Twitter scraping comprises 40,699 entries, comprising 40,132 unique timestamped entries, 40,135 unique tweet texts, and a categorization of four unique stock symbols and their corresponding company names: TSLA (Tesla, Inc.), F (Ford Motor Company), NIO (NIO Inc.), and XPEV (XPeng Inc.). The tweet "$TSLA will triple in 2022 🚀 🌕" is the most common, appearing 25 times, while the date "2021-11-11 01:54:37+00:00" is the most frequent, happening three times. These observations demonstrate the frequency of certain timestamps and tweet content. Additionally, the information highlights Tesla, Inc.'s supremacy with 37,422 entries linked to its stock symbol on a regular basis.

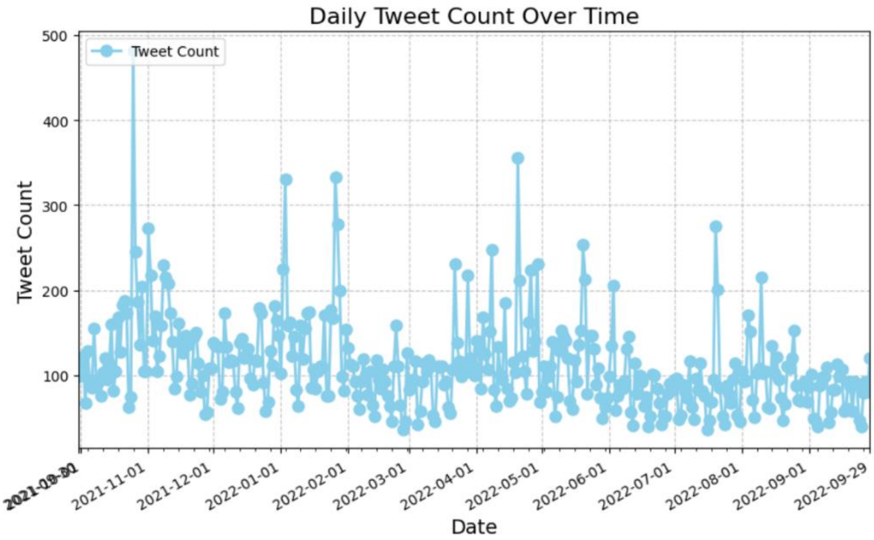


Figure 5 Distribution of tweets count over time.

This figure 5 shows the periodic distribution of tweet counts, with a special focus on the discussions involving Tesla, Ford, NIO, and XPeng. There has been a noticeable increase in twitter activity, which suggests that people are more interested in and have been talking about these companies frequently, as was previously discussed. One thing to notice is that there are noticeable variations all along the period that is shown in figure 5 The significant peaks found between November 2021 and May 2022, defined by daily tweet counts ranging from 490 to 370, are particularly significant, indicating increased involvement during these periods. On the other hand, September 2022 marks the lowest point in terms of recorded tweets, indicating a brief decline in social media conversation. This complex historical investigation highlights how opinions and conversations about the companies under investigation are ever-changing, which is important background information for the thorough assessment of their share of the market.

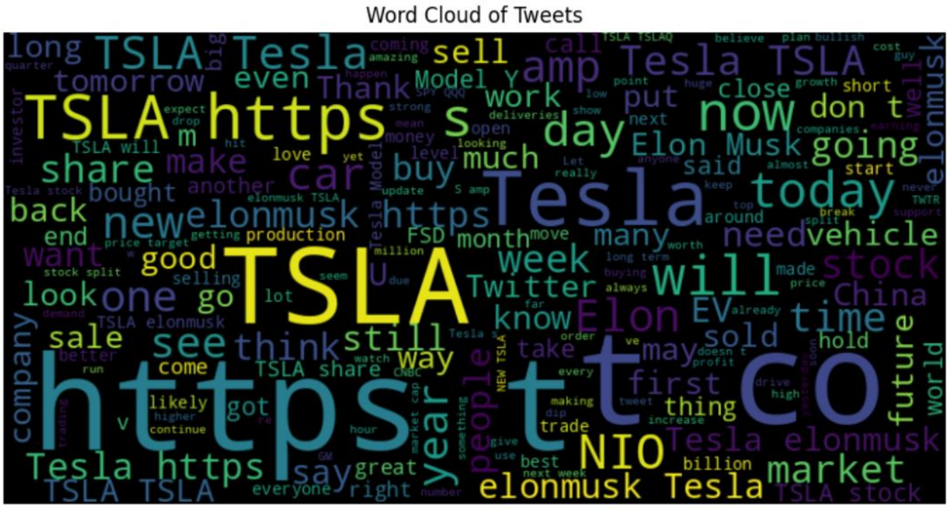


Figure 6 word cloud of Tweets.

The word cloud that is displayed in figure 6 visually summarizes the most prevalent topics discussed on Twitter around Tesla, Ford, NIO, and XPeng issues. Interestingly, the increased frequency of words like "Tesla," "TSLA," "stock," "Elon Musk," and "car" suggests that conversations mostly focused on Tesla, highlighting important details like its stock price, CEO, and list of products. A closer look finds words related to finance that are often used on Twitter, such as "bullish," "bearish," "long," and "short," suggesting that users are very interested in the companies' stock performance.

Among the words which stand out are negative ones like "sell," "put," and "loss," which indicate a wide range of sentiments and highlight the fact that not all Twitter users have a positive opinion of the stocks of the companies that are being discussed. When taken as a whole, the word cloud provides a concise summary of the most talked about subjects on Twitter, presenting Tesla as a contentious and highly discussed company with its stock price at the center of the conversation. This visual understanding considerably increases understanding of the complex nature of social media discussions focused on major manufacturers.

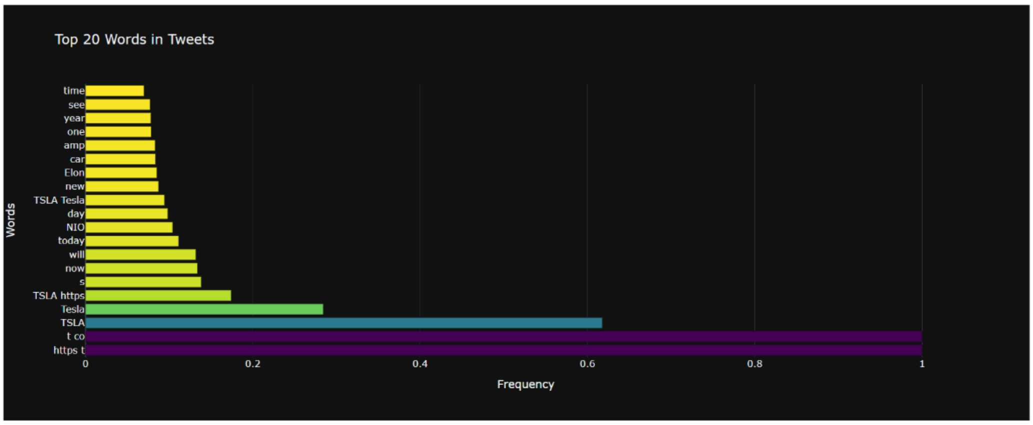


Figure 7 Top 20 words used in the tweets.

This Fig 7 shows the top 20 terms discussed on Twitter arranged by frequency, with "Tesla" being the most often used word and "time" being the least. Some of the most frequently used words that appear are "Elon Musk," "Tesla," "new," "year," "onc" (once), "day," "now," "time," and "see." This distribution suggests that the most talked about subjects on Twitter are current events, technology, and famous people, with Elon Musk and Tesla holding an important position in the discussion.

Figure 7 also emphasizes how common acronyms and abbreviations are used, such as "onc" for "once," "day" for "today," and "now." This is probably because tweets on the website are limited to 280 characters. By using concise language, Twitter users can transmit a variety of opinions on current events in a compact manner, which enhances our understanding of the complex and multidimensional nature of social media discussions.

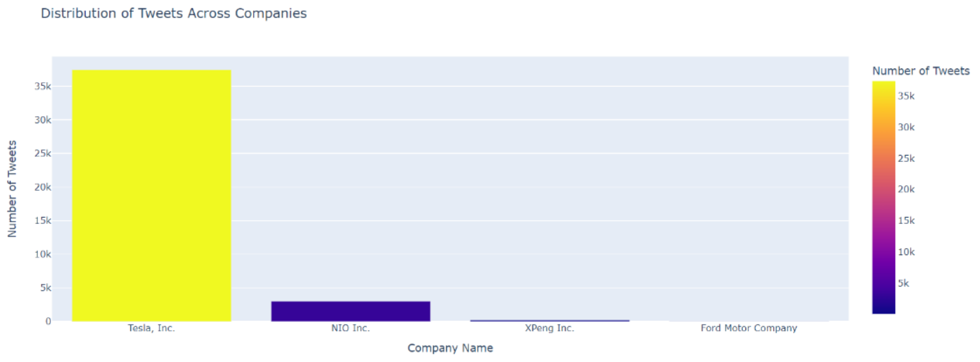
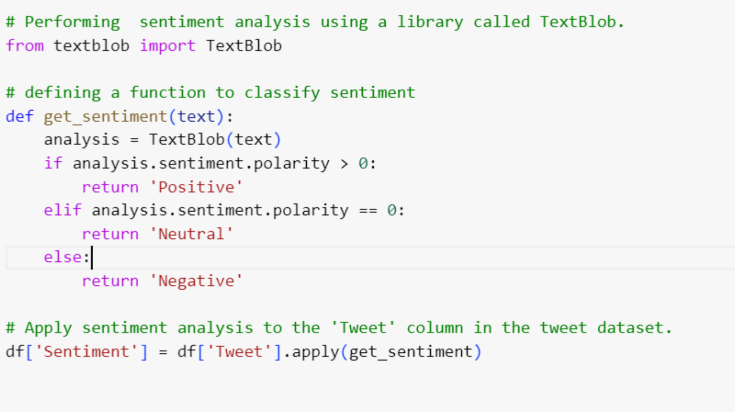


Figure 8 Distribution of tweets across companies.

Figure 8 illustrates how tweets from different electric car companies are distributed, with Tesla, Inc. at the top of the discussion, followed by NIO Inc., XPeng Inc., and Ford Motor Company. Interestingly, Tesla has more than 10 times as many tweets as Ford has with 37422 tweets. This difference might be explained by the fact that Tesla is the leader in the electric car market, that it uses Twitter often, and that it has an engaged audience that increases the number of retweets and shares. The pattern of total tweets increasing over time indicates a growing public interest in electric vehicles, possibly driven by factors like rising petrol prices, increasing accessibility to electric vehicles, and rising awareness of the environment. To summarize, the graph highlights the growing public interest in electric vehicles and highlights Tesla's presence on Twitter.

4.3 Data Preprocessing of Twitter Data.



```
# Performing  sentiment analysis using a library called TextBlob.
from textblob import TextBlob

# defining a function to classify sentiment
def get_sentiment(text):
    analysis = TextBlob(text)
    if analysis.sentiment.polarity > 0:
        return 'Positive'
    elif analysis.sentiment.polarity == 0:
        return 'Neutral'
    else:
        return 'Negative'

# Apply sentiment analysis to the 'Tweet' column in the tweet dataset.
df['Sentiment'] = df['Tweet'].apply(get_sentiment)
```

Figure 9 Sentiment Analysis on tweet data.

The TextBlob package was used to make natural language processing relatively easier when sentiment analysis was performed on a twitter dataset. The main function, called "get_sentiment," was carefully defined to receive a text input, which was reportedly a tweet. This prompted the creation of a TextBlob object specifically for sentiment analysis. TextBlob is a natural language processing package that is well-known for its adaptability in text-related activities. In this case, it simplified the sentiment analysis procedure. The sentiment.polarity property, an essential feature of the TextBlob architecture, was used to determine sentiment polarity within the textual content. Sentiment analysis on a dataset of tweets using TextBlob analyzer provides insightful information about the emotional responses of Twitter users. This TextBlob program improved the analysis of sentiment patterns in the data collected (Abiola et al., 2023)

As a result, the emotion of the text was determined by determining if the polarity exceeded zero, indicating that the text was 'Positive.' When the polarity was equal to zero, the emotion was appropriately labeled as "Neutral," but polarities below zero resulted in the classification as "Negative." The results of this sentiment analysis were carefully merged into a newly formed column called 'Sentiment' within the dataframe, selected as 'df.' This analytical technique, which is highly effective, provides a quick and insightful tool for assessing the sentiment that is present in the corpus of tweets that is being examined.

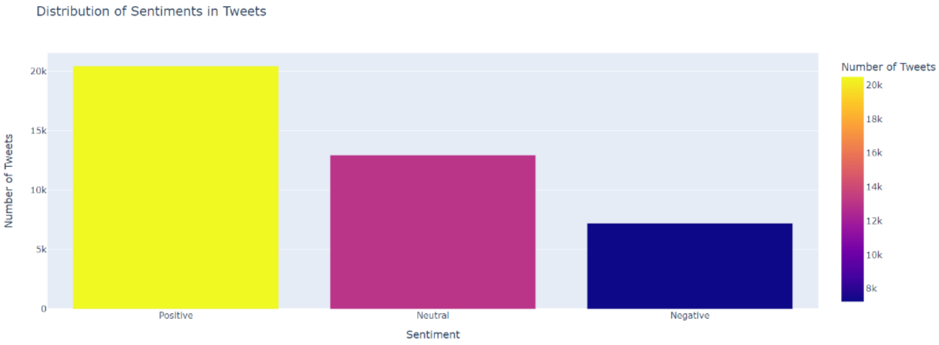
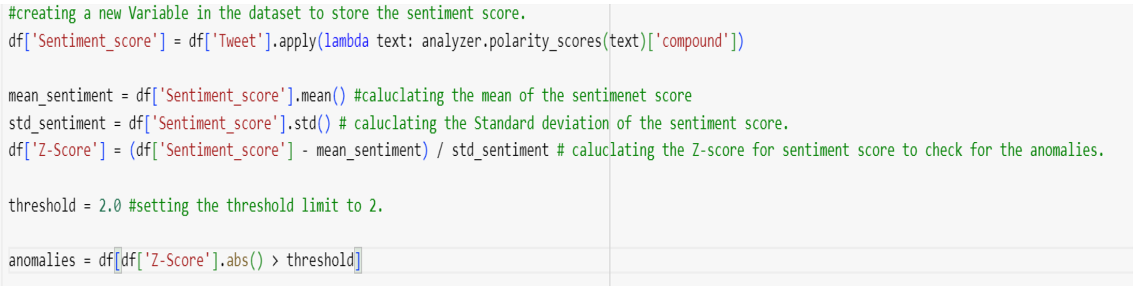


Figure 10 Distribution of sentiment in tweets.

The sentiment distribution that is contained in the tweet dataset can be seen graphically in figure 10, providing a thorough summary of the main feelings. There is a clear trend that indicates that there are more positive tweets in the dataset between 20,000 to 22,000, to be exact. Simultaneously, there is a notable number of Neutral tweets, with a range of 13,000 to 15,000 tweets. On the other hand, negative tweets appear in a range of 5,000 to 8,000 tweets. One important finding from this quantitative analysis is that there is a noticeable difference in the sentiment categories within the dataset. In particular, the dominance of positive attitudes could introduce bias into later categorization models or investigations.

A detailed comprehension of this disparity is essential because it highlights the need for focused approaches, such resampling methods or precision-weighted algorithms, to reduce possible biases and improve the adaptability of following investigations. Therefore, this understanding of the sentiment distribution of the dataset offers an essential basis for improving the approach and ensuring the accuracy of future research and classification models.



```
#creating a new Variable in the dataset to store the sentiment score.
df['Sentiment_score'] = df['Tweet'].apply(lambda text: analyzer.polarity_scores(text)['compound'])

mean_sentiment = df['Sentiment_score'].mean() #caluclating the mean of the sentimenet score
std_sentiment = df['Sentiment_score'].std() # caluclating the Standard deviation of the sentiment score.
df['Z-Score'] = (df['Sentiment_score'] - mean_sentiment) / std_sentiment # caluclating the Z-score for sentiment score to check for the anomalies.

threshold = 2.0 #setting the threshold limit to 2.

anomalies = df[df['Z-Score'].abs() > threshold]
```

Figure 11 calculating sentiment score using Vader sentiment analyzer.

'Sentiment_score' is a new variable that was added in the context of sentiment analysis applied to a twitter dataset. It captures the combination of sentiment polarity scores obtained from Vader sentiment Analyzer. The Valence Aware Dictionary for sEntiment Reasoner (VADER) is a robust multi-classification system for Twitter sentiment analysis, demonstrating high accuracy in detecting diverse sentiment classes (Elbagir and Yang, 2019). To provide a baseline for the sentiment distribution of the dataset, statistical metrics were then calculated, such as the mean and standard deviation of the sentiment scores. Z-scores for each sentiment score were computed, which made it easier to identify anomalies. These are those cases where the absolute Z-score is greater than a preset threshold of 2.0.

This method of analysis provides a reliable way to recognize and separate out outliers in the sentiment scores, which helps to provide a deeper understanding of the sentiment landscape inside the dataset. In the larger context of sentiment-based investigations, these anomalies which are indicative of deviations from the established sentiment norms play a critical role in improving the precision and dependability of future analyses and interpretations.

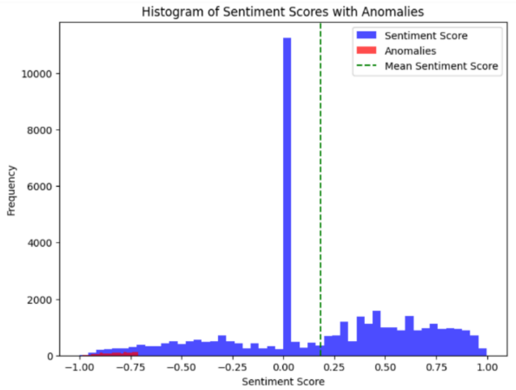


Figure 12 Distribution of Sentiment Scores with Anomalies.

The distribution of sentiment scores with anomalies is displayed in figure 12. While there are a few outliers at the negative end, the histogram indicates that sentiment ratings tend to be positive. The graph also displays the mean sentiment score, which is positive.

4.4 Building and evaluating Text classification Models.

- **4.4.1 Multinominal Naive Bayes Classifier**

A Naive Bayes classifier is used to build a text classification model for tweets. Using the train_test_split function, the dataset is divided into training and testing sets, with 80% of the data being used for training and 20% for testing. The TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is used to turn text data into numerical features by converting the raw text into a matrix of TF-IDF features. Common English stop words and phrases that appear in more than 70% of the documents are ignored by the vectorizer. Following the TF-IDF transformation of the training data, the Multinomial Naive Bayes technique is used to assess the model on the testing set. Using the accuracy_score function, the accuracy of the Naive Bayes classifier on the test set is calculated and the model showed the accuracy of 0.538. This accuracy score is a statistic used to evaluate how well the text classification algorithm predicts tweet sentiment labels.

- **4.4.2 Random Forest Classifier.**

A Random Forest classifier is used to construct a text classification model for tweets. For recurrence the RandomForestClassifier from the scikit-learn module is used with 100 decision trees and a fixed random state. The Random Forest model is trained on the TF-IDF (Term Frequency-Inverse Document Frequency) vectorized training data. The learned model is then applied to the test data that has been processed using the TF-IDF. Using the accuracy_score function, the model's predictions are compared against the testing set's actual sentiment labels. Model showed a accuracy score of 0.756. As an evaluation metric, this accuracy score shows how well the Random Forest classifier performs in predicting tweet sentiment labels based on the provided features.

- **4.4.3 Support Vector Machine Classifier.**

A Support Vector Machine (SVM) classifier is used to build a text classification model for tweets. The regularization parameter C is set to 1, and the linear kernel is used. The SVM model is trained using vectorized training data called TF-IDF (Term Frequency-Inverse Document Frequency). The learned model is then applied to test data that has been processed using TF-IDF. Using the accuracy_score function, the SVM model's predictions are compared against the testing set's actual sentiment labels. The accuracy that results is 0.802.

- **4.4.4 Decision Tree Classifier.**

A Decision Tree classifier is used in the development of a text classification model for tweets. For recurrence a fixed random state is employed using the scikit-learn DecisionTreeClassifier. The Decision Tree classifier is trained on the vectorized training data of TF-IDF (Term Frequency-Inverse Document Frequency) and subsequently used to the test data that has undergone TF-IDF transformation. 0.696 is the accuracy obtained by using the accuracy_score function to compare the model's predictions with the actual sentiment labels in the testing set.

- **4.4.5 K-Nearest Neighbors Classifier.**

A K-Nearest Neighbors (KNN) classifier is used to build a text classification model for tweets. Using the scikit-learn KNeighborsClassifier with a value of k=5, the model makes predictions by considering the labels of its five closest neighbors. TF-IDF (Term Frequency-Inverse Document Frequency) vectorized training data is used in the training process, and the learned KNN classifier is then applied to the test data that has undergone TF-IDF transformation. The accuracy_score function is used to evaluate the model's predictions to the actual sentiment labels in the testing set, and the resultant accuracy is 0.347.

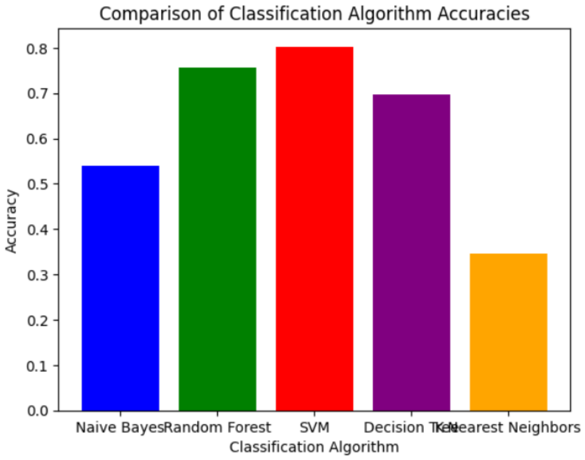
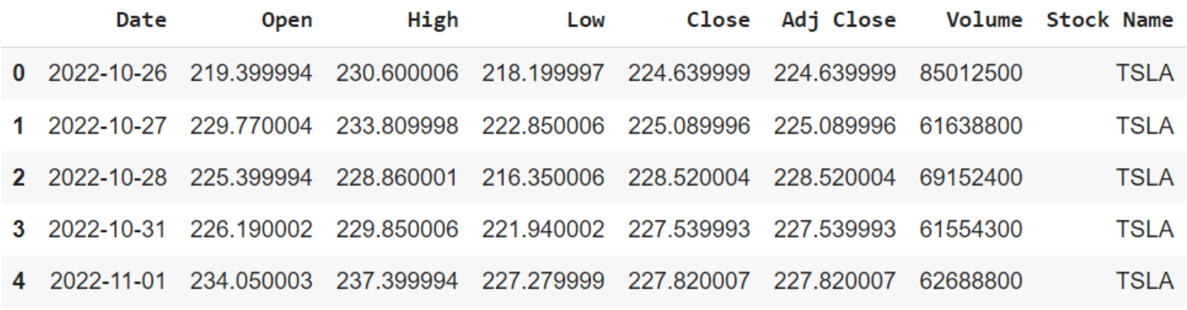


Figure 13 Performance metrices of text classification models.

The Support Vector Machine (SVM) classifier performed well in correctly identifying the test tweets, as demonstrated by the data shown in Figure 13. With an impressive accuracy rate of 0.802, the model shows that it is capable of correctly classifying tweets according to their sentiment. This result highlights the SVM classifier's effectiveness in accurately addressing the complex nature of categorization of sentiment.

4.5 Data understanding and EDA of Historical Stock data.



	Date	Open	High	Low	Close	Adj Close	Volume	Stock Name
0	2022-10-26	219.399994	230.600006	218.199997	224.639999	224.639999	85012500	TSLA
1	2022-10-27	229.770004	233.809998	222.850006	225.089996	225.089996	61638800	TSLA
2	2022-10-28	225.399994	228.860001	216.350006	228.520004	228.520004	69152400	TSLA
3	2022-10-31	226.190002	229.850006	221.940002	227.539993	227.539993	61554300	TSLA
4	2022-11-01	234.050003	237.399994	227.279999	227.820007	227.820007	62688800	TSLA

Figure 14 Top five rows of Historical Stock dataset.

The dataset was carefully taken from Yahoo Finance and consists of 4723 rows and 8 columns. The data includes a wide variety of financial data that is necessary for an in-depth investigation of the stock market. A periodic series of historical stock performance is provided by each row, which relates to a particular date. Following are the column names that make up the structure of the dataset:

- **Date:** The day on which the stock prices were noted.
- **Open:** The stock's opening price on the given date.
- **High:** The price at which the stock hit its peak during the trading session.
- **Low:** The stock's closing price for the trading day.
- **Closing:** The stock's price at the end of the trading day.

- **Adj Close:** The adjusted closing price after taking into consideration business decisions like stock splits and dividends.
- **Volume:** The total number of shares traded on the specified day.
- **Stock Name:** The name or symbol of the stock that is linked to the prices that have been recorded.

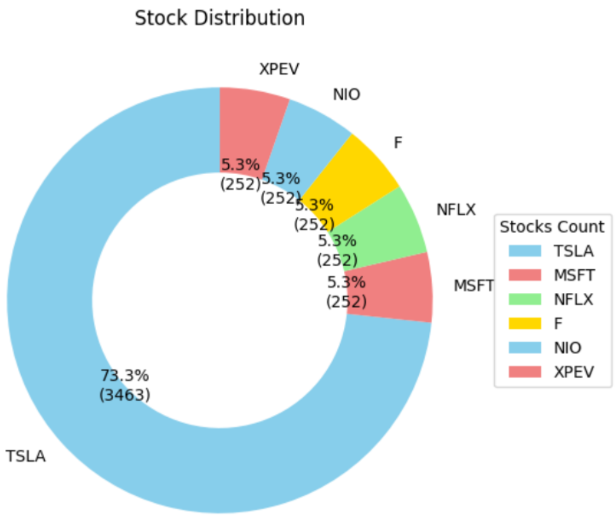
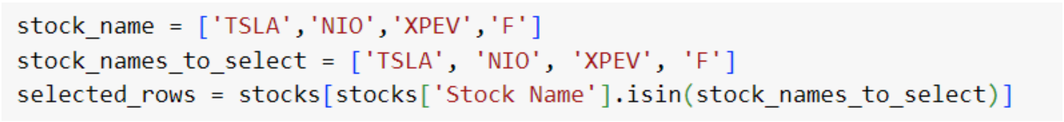


Figure 15 Stock Distribution Across Companies

Figure 15 shows the distribution of stocks, with a focus on six major companies: TSLA, MSFT, NFLX, F, XPEV, NIO. TSLA is the most valuable stock in this portfolio, accounting for 73.3% of the entire distribution. With a contribution of 5.3% each, MSFT, NFLX, F, and XPEV, NIO all contributes equally to the remaining 26.7%. Looking at the share counts, 3463 shares is the most that TSLA controls, indicating that it is a significant component of the portfolio. XPEV, NFLX, F, and MSFT all have 252 shares.

4.6 Preprocessing of Historical Stock Data.

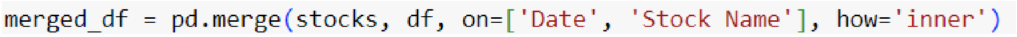


```
stock_name = ['TSLA','NIO','XPEV','F']
stock_names_to_select = ['TSLA', 'NIO', 'XPEV', 'F']
selected_rows = stocks[stocks['Stock Name'].isin(stock_names_to_select)]
```

Figure 16 Removal of non-Relevant stocks from the datasets

During stock data analysis, the dataset was generated with a concentration on the automotive sector, which includes stock data related to Tesla, Nio, Ford, and Xpeng. Understanding that Netflix and Microsoft do not support our main goal of forecasting stock prices in the automotive industry, these data points were intentionally eliminated from our dataset to improve the precision and relevance of the analysis. As a result, the filtered dataset which consists only of vehicle stock data has a simple structure with dimensions (4219, 8). This calculated removal of non-essential stocks guarantees a more relevant and focused investigation of trends and patterns in the automotive sector for our forecasting.

4.7 Integrating sentiment Tweet data and Historical Stock Data.

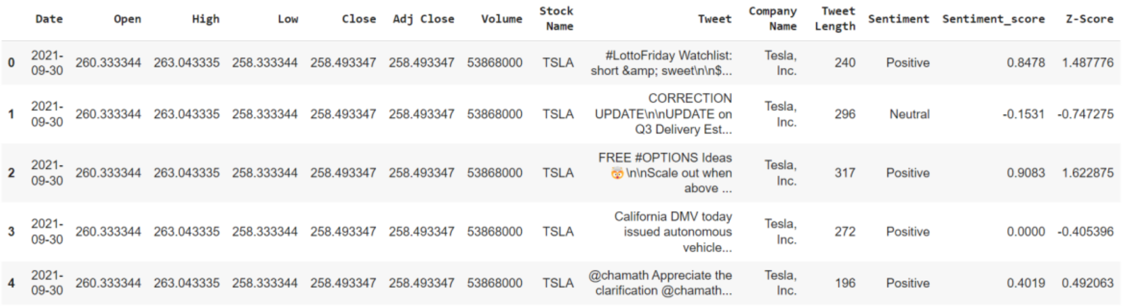


```
merged_df = pd.merge(stocks, df, on=['Date', 'Stock Name'], how='inner')
```

Figure 17 Merging of sentiment and Historical Stock data.

Utilizing an inner join ('how='inner"), two datasets, "stocks' and "df," are combined based on the common fields "Date" and "Stock Name." The 'Date' columns in both datasets were already converted into datetime format before the merging. 'merged_df,' the resulting dataframe, shows where the two datasets meet and only includes the rows where the values for 'Date' and 'Stock Name' match. By integrating important data points into a single dataframe, this merging process helps with analysis and is helpful for combining information from various datasets.

4.8 Understanding of Merged Data.



	Date	Open	High	Low	Close	Adj Close	Volume	Stock Name	Tweet	Company Name	Tweet Length	Sentiment	Sentiment_score	Z-Score
0	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA	#LottoFriday Watchlist: short &amp; sweet\n\n$...	Tesla, Inc.	240	Positive	0.8478	1.487776
1	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA	CORRECTION UPDATE\n\nUPDATE on Q3 Delivery Est...	Tesla, Inc.	296	Neutral	-0.1531	-0.747275
2	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA	FREE #OPTIONS Ideas 🦉 \n\nScale out when above ...	Tesla, Inc.	317	Positive	0.9083	1.622875
3	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA	California DMV today issued autonomous vehicle...	Tesla, Inc.	272	Positive	0.0000	-0.405396
4	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA	@chamath Appreciate the clarification @chamath...	Tesla, Inc.	196	Positive	0.4019	0.492063

Figure 18 Top five records of the merged Dataset.

The combined dataset, which consists of 48,300 rows and 14 columns, has an extensive collection of sentiment and financial data. Important financial indicators like "Date," "Open," "High," "Low," "Close," "Adj Close," and "Volume" are included in the columns, offering a thorough understanding of stock market trends. Furthermore, sentiment analysis features are included in the dataset; these variables include columns labeled "Tweet," "Company Name," "Tweet Length," "Sentiment," "Sentiment_score," and "Z-Score". The integration of sentiment and financial data enables an in-depth investigation of the correlation between market performance and the sentiment expressed through tweets. This provides insightful information for strategic planning and thorough decision-making within the financial sector.

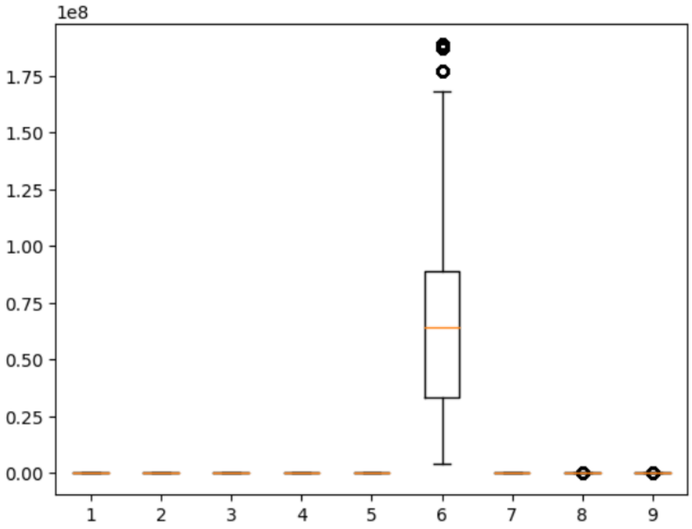


Figure 19 Boxplot to check for the outliers.

Figure 19 shows that the combined dataset contains outliers. However, the lack of data scaling has prevented an accurate identification of these outliers. The lack of normalization or standardization in the dataset makes it difficult to identify the individual cases that considerably differ from the distribution of data. To analyze and identify outliers more accurately, scaling is necessary to bring the numerical features into a uniform range. By addressing this scaling issue, outlier identification methods will become stronger and help us understand the distributional properties of the dataset in more detail.

4.9 Preprocessing of merged data.

As part of the dataset preparation, the 'Date' column was converted to a datetime format. Following that, two more columns were created called "Day of Week" and "Month" to extract the relevant day and month data from the "Date" column. A more detailed examination of the temporal patterns in the data is made possible by this temporal breakdown. A traditional scalar transformation was utilized to fix scale differences among the numerical variables. Notably, standardization was applied to the target variable along with the other numerical characteristics. The target variable had a significantly greater scale (200–300) than for the independent variables. By ensuring that all variables are listed on a uniform scale, this standardization procedure promotes comparability and lessens the effect of scale variances in following investigations or modeling tasks.

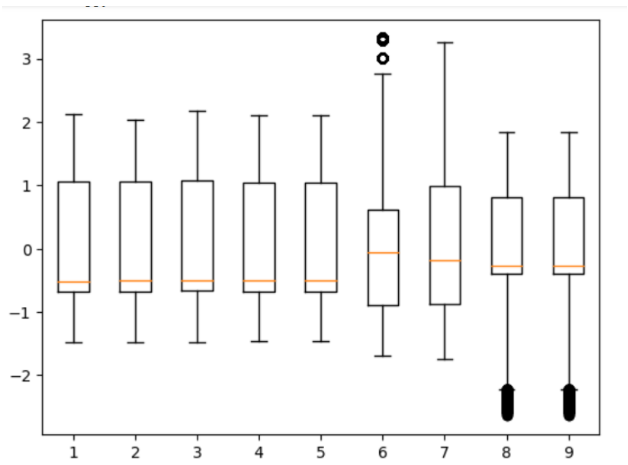
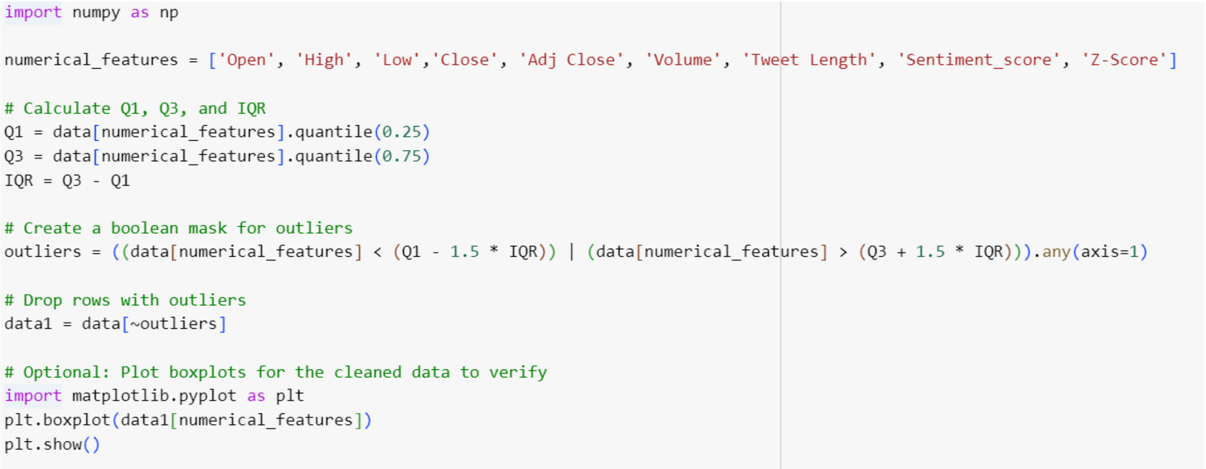


Figure 20 Boxplot to check outliers on scaled data.

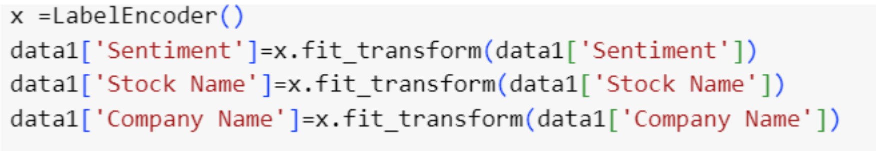
By looking at figure 20 columns 6, 8, and 9 have outliers. To maximize model performance and produce a more precise and trustworthy prediction, it is essential to address these outliers. Using outlier treatment techniques in these columns will strengthen the analytical framework.



```
import numpy as np

numerical_features = ['Open', 'High', 'Low','Close', 'Adj Close', 'Volume', 'Tweet Length', 'Sentiment_score', 'Z-Score']

# Calculate Q1, Q3, and IQR
Q1 = data[numerical_features].quantile(0.25)
Q3 = data[numerical_features].quantile(0.75)
IQR = Q3 - Q1

# Create a boolean mask for outliers
outliers = ((data[numerical_features] < (Q1 - 1.5 * IQR)) | (data[numerical_features] > (Q3 + 1.5 * IQR))).any(axis=1)

# Drop rows with outliers
data1 = data[~outliers]

# Optional: Plot boxplots for the cleaned data to verify
import matplotlib.pyplot as plt
plt.boxplot(data1[numerical_features])
plt.show()
```

Figure 21 Removing outliers from the dataset by using Boolean Mask for Outliers.

Identification and elimination of outliers for a given subset of the dataset's numerical characteristics. For the given numerical columns, the first quartile (Q1), third quartile (Q3), and interquartile range (IQR) are first calculated. The rows with outliers are then identified using a Boolean filter based on the 1.5 times IQR criteria. Following this, rows that contain identified outliers are removed from the dataset, giving a refined dataset known as 'data1.' For verification, boxplots of the cleaned data have optionally been displayed. By treating outliers methodically, the dataset's integrity is improved, which encourages robustness in later analysis.



```
x =LabelEncoder()
data1['Sentiment']=x.fit_transform(data1['Sentiment'])
data1['Stock Name']=x.fit_transform(data1['Stock Name'])
data1['Company Name']=x.fit_transform(data1['Company Name'])
```

Figure 22 Label Encoding of numerical columns.

To convert category columns into numerical representations, label encoding was used. Through this method, distinct categorical values are given separate number labels, allowing categorical data to be easily integrated into quantitative analysis. The conversion improves the overall understanding and efficiency of the analytical process and ensures compatibility with machine learning methods.

4.10 Feature Selection.

4.10.1 Feature Selection using Random Forest Regressor with Recursive Feature Elimination (RFE)

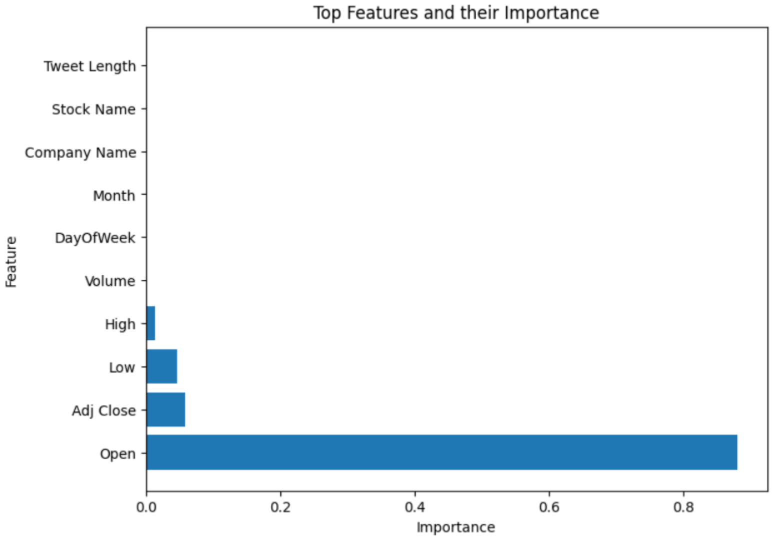


Figure 23 Feature selection using Random Forest Regressor with RFE

By utilizing a Random Forest Regressor combined with Recursive Feature Elimination, the method systematically identified the top 10 characteristics that are essential for building an effective stock price prediction model. By focusing on important predictors, this systematic selection procedure strengthens the model's robustness and increases its effectiveness in identifying complex patterns in the combined sentiment data from tweets and historical stock data. Figure 23 indicates that the important features Open, Adj Close, Low, and High that were obtained using Random Forest Regressor with Recursive Feature Elimination (RFE) are clear. This decision highlights the importance of these characteristics in building a strong stock price prediction model, offering an obvious focus for model improvement and interpretability.

4.10.2 Feature Selection using Mutual Info Regressor.

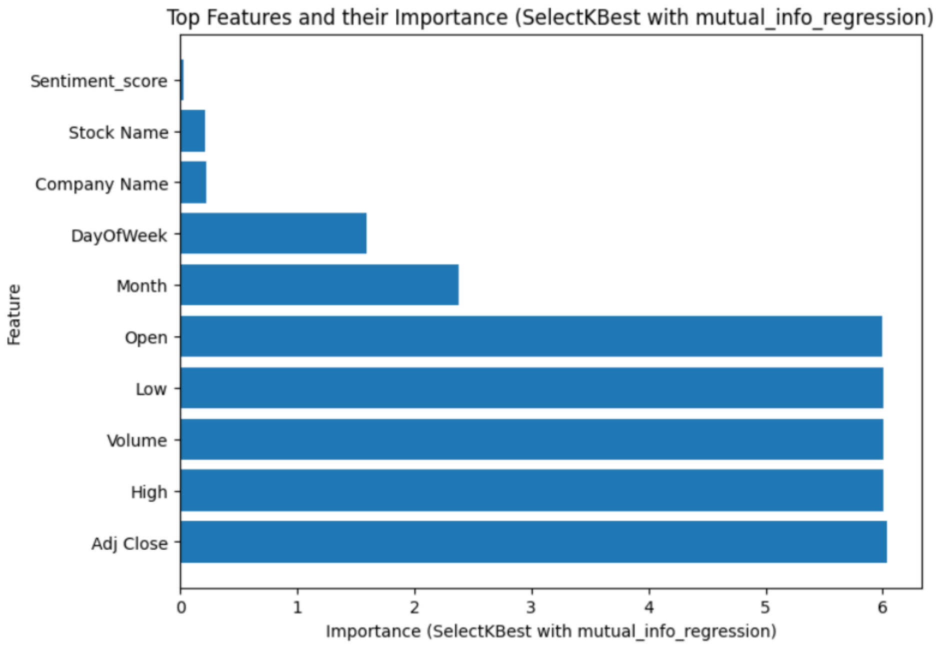


Figure 24 Feature Selection using Mutual Info Regressor.

Figure 24 shows how important it is to use 10 crucial variables when creating an effective stock price prediction model: adj close, high, volume, low, open, month, day of week, company name, stock name, and sentiment score. The technique uses the Mutual Info Regressor to measure the mutual information between these features and the target variable, highlighting their significance. In addition to identifying the fundamental connections between attributes and stock prices, this complete method highlights the model's ability to identify minor connections, resulting in a prediction model that is more accurate and understandable. Sentiment scores are an additional feature that improves the model's overall prediction power by combining sentiment analysis with financial metrics.

4.10.3 Feature Importance Using Lasso.

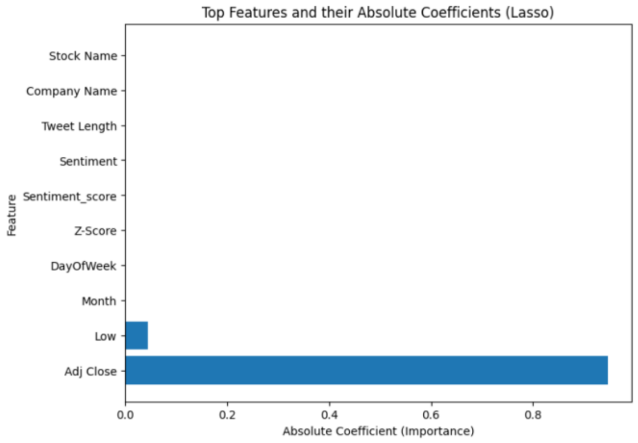


Figure 25 Feature Selection using Lasso.

According to figure 25, Adj Close and Low are the primary differentiators for building a successful stock price prediction model when using the Lasso feature selection approach. A regularization technique called Lasso intentionally places limits on the coefficients, causing the model to give preference to a small number of significant features. The order of importance for Adj Close and Low in this context shows how important they are for reducing noise and improving the interpretability of the model. In addition to simplifying the predictive framework, this careful feature selection demonstrates Lasso's contribution to maximizing model simplicity for better generalization and performance.

4.10.4 Feature Selection using Selecting the Kbest with F-classif.

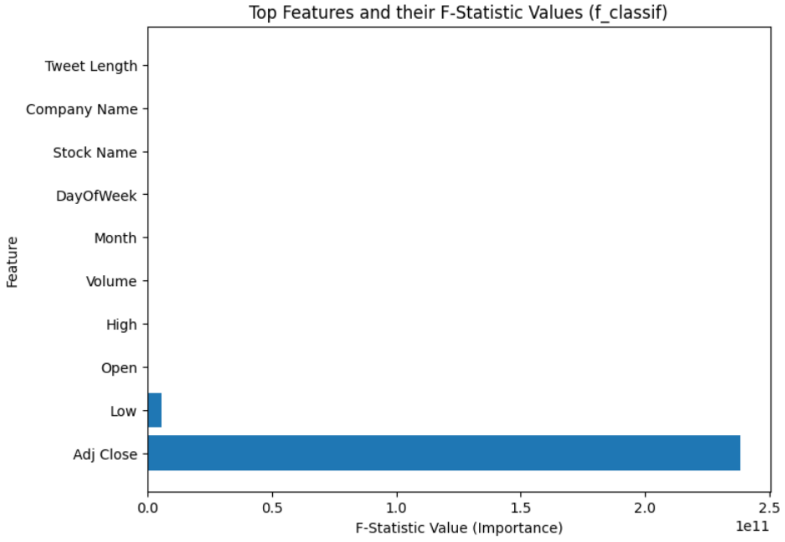


Figure 26 Feature Selection using kbest with f_classif.

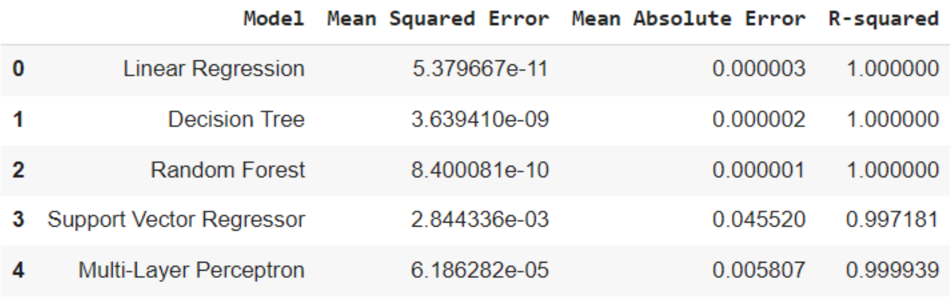
Figure 26 clearly illustrates that, using the SelectKBest technique with F_classif, the two most important attributes for creating a strong stock price prediction model are Adj Close and Low. To retain the most relevant variables, SelectKBest with F_classif systematically assesses and chooses features according to how well they distinguish across categories. Adj Close and Low serve as leading instances of this careful selection, which not only highlights their statistical significance but also demonstrates how effective this approach is in maximizing feature relevance. SelectKBest with F_classif, which prioritizes discriminative power, improves the model's accuracy and provides a more sophisticated prediction framework that can lead to more precise stock price predictions.

Using crucial data found using Mutual Info Regressor Adj Close, High, Volume, Low, Open, Month, Day of Week, Company Name, Stock Name, and Sentiment Score are the important features to build a strong stock price prediction model. In addition to capturing related relationships, this strategic feature selection approach incorporates sentiment analysis and financial measures to provide a prediction framework that is both accurate and comprehensive.

For contemporary feature selection techniques to be effective, mutual complementarity, or cooperation, must be considered in addition to applicability and redundancy (Vergara and Estévez, 2014)

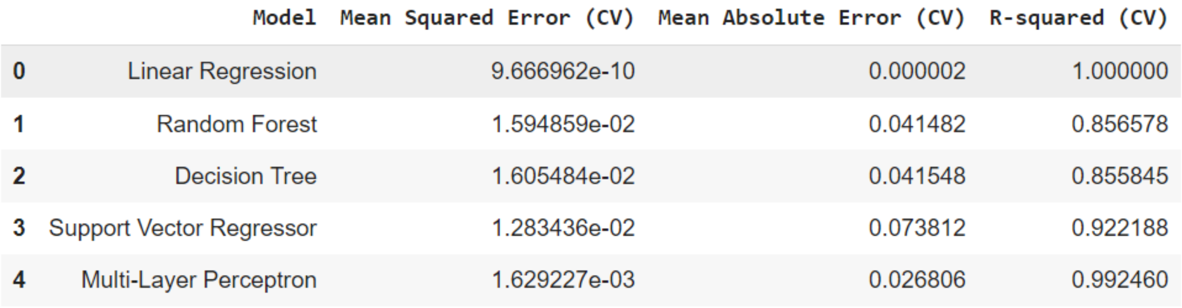
4.11 Model Building with and without Cross Validation technique.

Several regression models were implemented, each optimized with the top attributes found by Mutual Info Regressor, to conduct a thorough prediction analysis. The models include Decision Trees, Support Vector Regressors, Random Forest Regressors, Linear Regressions, and Multilayer Perceptron’s, Neural Network. By using the identified useful features to evaluate and compare the models' abilities to capture the complex patterns present in the dataset, this technique ensures a comprehensive analysis of the predictive landscape. A combination of methods like this improves the predictive framework's robustness and provides a more detailed knowledge of the correlations between the chosen features and stock price movements.



	Model	Mean Squared Error	Mean Absolute Error	R-squared
0	Linear Regression	5.379667e-11	0.000003	1.000000
1	Decision Tree	3.639410e-09	0.000002	1.000000
2	Random Forest	8.400081e-10	0.000001	1.000000
3	Support Vector Regressor	2.844336e-03	0.045520	0.997181
4	Multi-Layer Perceptron	6.186282e-05	0.005807	0.999939

Figure 27 performance of the models without cross validation technique.



	Model	Mean Squared Error (CV)	Mean Absolute Error (CV)	R-squared (CV)
0	Linear Regression	9.666962e-10	0.000002	1.000000
1	Random Forest	1.594859e-02	0.041482	0.856578
2	Decision Tree	1.605484e-02	0.041548	0.855845
3	Support Vector Regressor	1.283436e-02	0.073812	0.922188
4	Multi-Layer Perceptron	1.629227e-03	0.026806	0.992460

Figure 28 Performance of the models with cross validation technique.

4.11.1 Linear Regression.

The Linear Regression model achieves impressive performance measures as shown in figure 28, including a mean absolute error (MAE) of 0.000003 and a mean squared error (MSE) of 5.379667e-11, demonstrating high predictive precision. An almost perfect match is shown by its exceptional R-squared value of 0.9999999999466892, which indicates an exceptional alignment with the dataset. But there are concerns about possible overfitting considering how close the R-squared number is near 1. By figure 28 it is evident that Overfitting issues are successfully addressed by using strong generalization techniques like cross-validation. The Linear Regression model maintains its ability to predict via cross-validation, providing an R-squared value of 1, strengthening its accuracy in predicting the price of stocks. This highlights the model's ability to find an appropriate balance between generalization and accuracy, guaranteeing its reliable performance across a variety of datasets.

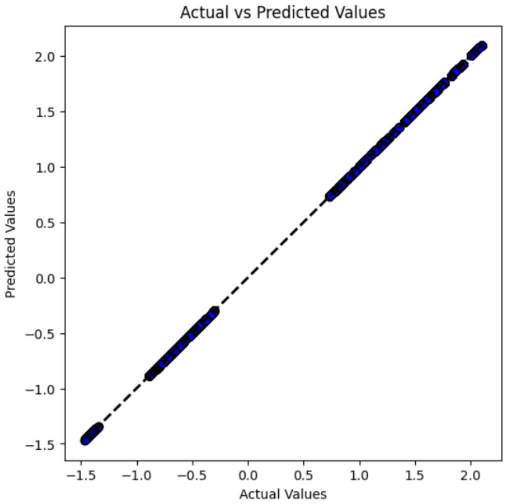


Figure 29 Actual VS predicted value linear regression with cross validation technique.

4.11.2 Decision Tree

The Decision Tree model's low error rates and impressive R-squared value of 0.9999999963934584 illustrate its remarkable ability to identify complex patterns in the dataset. With mean absolute error (MAE) of 1.694e-6 and mean squared error (MSE) of 3.639e-9, these measures highlight how accurate and useful the model is at capturing complex relationships. But overfitting is a possibility, especially when the R-squared value approaches 1, which indicates a nearly perfect fit. To tackle this issue, the model was subjected to a thorough assessment using cross-validation, which led to a moderate decrease in the R-squared value to 0.855. By carefully using generalization approaches, this improves the model's robustness, prevents overfitting, and guarantees consistent performance on a variety of datasets.

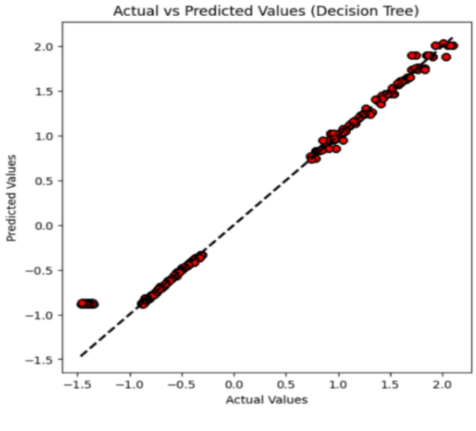


Figure 30 Actual VS predicted value of Decision Tree Regressor with cross validation technique.

4.11.3 Random Forest.

With an extremely low mean squared error (MSE) of 8.400081302908498e-10 and mean absolute error (MAE) of 1.2178382709735516e-6, the Random Forest Regressor significantly improves predictive accuracy. The model remarkably achieves a very high R-squared value of 0.9999999991675781, highlighting its remarkable ability to generalize to new data. These low error rates show how well the Random Forest Regressor captures complex correlations in the data, making it a dependable tool for precise and trustworthy stock price predictions. The near-perfect R-squared value raises worries about overfitting, which are mitigated by a careful application of cross-validation techniques. With a lower R-squared value of 0.856, the improved model produced by this calculated approach increases the Random Forest Regressor's robustness and guarantees its applicability to a variety of datasets.

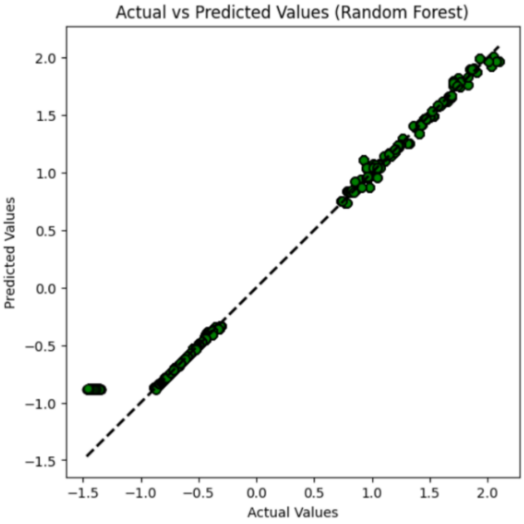


Figure 31 Actual VS predicted value of Random Forest Regressor with cross validation technique.

4.11.4 Support Vector Regressor.

The high R-squared value of 0.9971813516276729 indicates that the Support Vector Regressor is a highly effective predictor of underlying patterns in the dataset. With a mean squared error (MSE) of 0.002844336030140915 and a mean absolute error (MAE) of 0.04551962890124118—slightly higher errors than ensemble models—the model skillfully handles these difficulties while maintaining high predicted accuracy. A useful technique that achieves an acceptable balance between accurate forecasting and an advanced understanding of the complex relationships involved in stock price prediction is the Support Vector Regressor. Its performance is impressive, but because the R-squared value is getting close to one, care should be taken to avoid any overfitting. By carefully refining the model with cross-validation procedures, the model's R-squared value is lowered to 0.922, greatly improving its generalization capabilities.

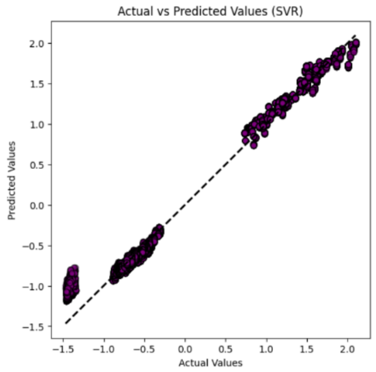


Figure 32 Actual VS predicted value of Support Vector Regressor with cross validation technique.

4.11.5 Multi- Layer Perceptron.

The Multi-Layer Perceptron model has outstanding prediction accuracy; its mean absolute error (MAE) is 0.005806802547602894, and its mean squared error (MSE) is 6.186281884124661e-5. Its exceptional performance is confirmed by an excellent R-squared value of 0.9999386958742614, which demonstrates its ability in identifying complex relationships in the data. With its near-perfect R-squared value and negligible error, the Multi-Layer Perceptron model is a useful tool for accurate stock price predictions. It can model and anticipate financial data with remarkable precision and dependability due to its advanced understanding of complex relationships. Since the R-squared value is getting closer to one, which indicates that overfitting may occur, cross-validation procedures are strategically applied. After careful consideration, an altered model with a slightly lower R-squared value of 0.992 is produced, preserving good predictive performance but preventing overfitting and strengthening the model's robustness.

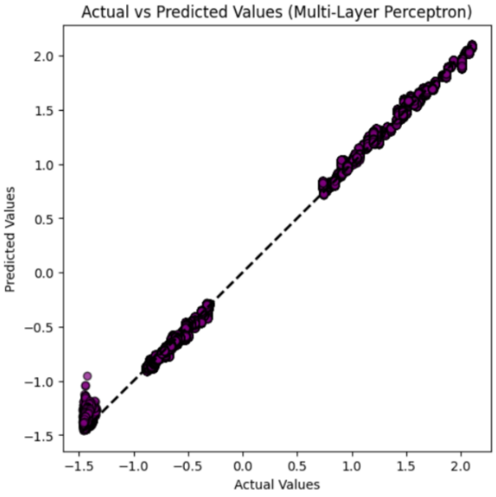


Figure 33 Actual VS predicted value of Multi-Layer Perceptron with cross validation technique.

In summary, the Mutual Info Regressor's top features enable the regression models to perform remarkably well in terms of prediction. The extremely high R-squared values, which are getting close to 1, are important because they may indicate overfitting, in which case the models might be capturing noise in the training set. Even while Multi-Layer Perceptrons, Random Forests, Decision Trees, and Linear Regressions demonstrate exceptional accuracy, more investigation is necessary to assess how well they transfer to new data. Even with slightly greater errors, the Support Vector Regressor performs well. Maintaining a balance between generalization and model complexity will be important to address overfitting issues and guarantee the accuracy of stock price predictions.

4.12 Evaluation.

The model evaluation section uses important measures to ensure reliability and precision while critically evaluating the performance of predictive models created for stock price prediction. The metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) serve as critical indicators that enable a thorough evaluation of prediction abilities. Greater model fit is indicated by a higher R2 score, and stronger alignment between projected and actual scores is indicated by lower MSE and MAE values. Using sentiment analysis and stock data, this thorough examination of many models offers insightful information about their advantages and disadvantages, facilitating the selection of the most accurate and ideal models for stock price prediction.

a) Mean Squared Error:

Purpose: MSE measures prediction accuracy, but it also penalizes higher errors with higher penalties than MAE. Squaring the differences between expected and actual numbers highlights the significance of outliers.

Calculation: The mean square error (MSE) can be computed by taking the square of the difference (y) between each predicted and actual value, adding these squared differences, and dividing the result by the total number of data points (n).

MSE = $\Sigma(\hat{y} - y)2 / n$

b) Mean Absolute Error:

Purpose: The average absolute difference between expected and actual data is determined using the mean absolute error (MAE) method. It provides an easy way to assess the precision of the model.

Calculation: By summing the absolute differences between each predicted value () and the corresponding actual value (y), dividing by the total number of data points (n), and then adding the results, the MAE is computed as follows:

MAE = $\Sigma|\hat{y} - y| / n$

c) R Squared Value:

Purpose: R2 calculates how well the regression model matches the observed data. It shows the amount of the variance of the dependent variable (y) that can be explained by the independent variables or predictors in the model.

Calculation: For determining the R2 value, the variance of the actual values (y) is compared to the variance of the expected values ():

R2 = 1 - (Σ(ŷ - y)2 / Σ(y - ȳ)2)

The MSE, MAE and R-squared Value are shown in figure 34.

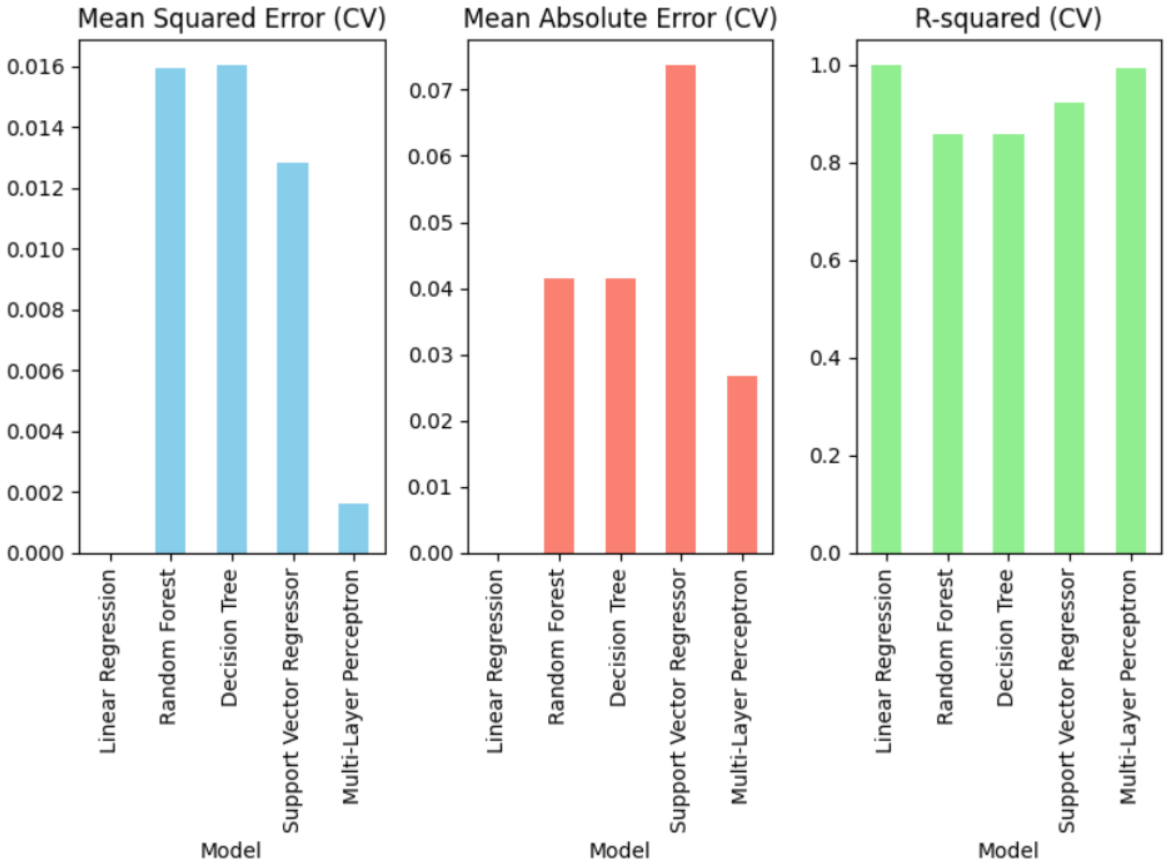


Figure 34 performance metrices of the models.

4.12.1 Result and Outcome.

- When it comes to stock price prediction, the Linear Regression model and the Multilayer Perceptron model both perform well, with low Mean Squared Error (MSE) and Mean Absolute Error (MAE) values. Remarkably, both models attain an exceptional R-squared value of 1, indicating a perfect compliance with the fundamental data. These findings highlight the strong predictive capabilities of both models and show how well they can forecast stock prices with little error. Their effectiveness is further enhanced by their combination of low MSE, MAE, and a maximal R-squared value, which makes them useful instruments for accurate and trustworthy stock price forecasts.

- The Random Forest Regressor, Decision Tree Regressor, and Support Vector Regressor models perform well in predicting stock values, although with slightly higher Mean Squared Error (MSE) and Mean Absolute Error (MAE) compared to the Linear Regression model. These models' R-squared values, which vary from 0.85 to 0.92, show a strong capacity to identify underlying patterns in the data, even though they fall short of the ideal range of a 1. These models provide an effective compromise between accuracy and interpretability, demonstrating strong forecasting skills and contributing to a full understanding of stock price dynamics despite the slightly larger error metrics.

Concluding the final model:

The best model for predicting stock prices using sentiment and historical stock data is the Random Forest Regressor. Its capacity to manage overfitting is a major strength, despite having slightly greater Mean Squared Error (MSE) and Mean Absolute Error (MAE) and a considerably lower R-squared value of 0.856 compared to other models. In addition to reducing the possibility of overfitting and guaranteeing reliable performance in a range of situations, the Random Forest model excels in generalizing to a wide range of datasets. In the setting of historical stock and sentiment data, the Random Forest Regressor is the best option for precise and trustworthy stock price predictions due to its careful balance between generalization and predictive capacity. Overfitting reduces during training when data and feature samples are randomly chosen, and tree ensembles produce accurate and reliable predictions (Salam et al., 2021)

CHAPTER 5

CONCLUSION

5.1 Research Overview.

To advance the field of financial forecasting, this research project was carefully planned to examine how sentiment analysis taken from Twitter data is incorporated into stock price prediction models, focusing on well-known electric vehicle (EV) manufacturers such as Tesla, Ford, Nio, and Xpeng. The primary goal of this study was to evaluate the feasibility and effectiveness of utilizing sentiment scores obtained from social media sentiments, specifically those originating from Twitter, to enhance the accuracy and consistency of stock price predictions in the dynamic and rapidly changing electric vehicle industry. By addressing an essential aspect of modern financial research, the study aimed to provide insightful information on the possible relationship between sentiment patterns in social media discussion and subsequent stock price changes. The project was to explain the complex relationship between market dynamics and online sentiments, providing a comprehensive viewpoint that could be useful in improving financial decision-making and prediction algorithms.

5.2 Problem Definition.

In the complex and dynamic environment of the stock market, a wide range of factors impact its direction, with public opinion emerging as an important factor. This study aims to investigate the implications of the relationship between stock price projections and sentiment analysis against the background of the rapidly expanding electric vehicle industry. The key problem addressed here is determining the true value of sentiment data as a distinct indication for anticipating stock price fluctuations.

5.3 Design / Evaluation and Results.

A comprehensive study approach was used, which included sentiment analysis with Natural Language Processing (NLP), historical stock price data, and a variety of machine learning models, including Multi-Layer Perceptron (MLP), Decision Trees, Random Forest, Linear Regression, and Support Vector Machines (SVM). Notably, TextBlob and Valence Aware Dictionary for Sentiment Reasoner (VADER) were also included for effective sentiment analysis. Notably, Mutual Information Regressor successfully identified sentiment score as a significant predictor for stock price prediction, Support Vector Machines (SVM) showed better performance in identifying tweet data pertaining to stocks. Through the application of cross-validation procedures, the study effectively mitigated data overfitting and proved the Random

Forest model's persistent outperformance, offering important insights into varying model performances. The study also indicated enhanced accuracy and generalizability.

5.4 Contribution and Impact.

This study advances the field of sentiment analysis in stock price prediction in a number of important ways. First, it proves that major EV manufacturers' models can successfully include sentiment analysis from Twitter data. According to the study, Random Forest is the best model for making these kinds of forecasts, highlighting its capacity to capture complex nonlinear correlations between feeling and stock values. The results highlight sentiment analysis's potential to increase stock market efficiency and provide direction for calculated investing decisions.

5.4.1 Guidance for Investors and Traders.

The established models of sentiment-based stock price prediction are a useful resource for traders and investors looking for better decision support. Sentiment analysis from social media, especially Twitter, can provide investors with up-to-date information on how the public feels about different electric vehicle manufacturers. In the fast-paced and moving world of the stock market, this information can assist stakeholders make better-informed decisions by guiding investment plans.

5.4.2 Risk Mitigation for Financial Analysts

The results of this study could help investors and financial analysts by enabling them to use sentiment analysis as an extra risk reduction tool. Analysts have an extra layer of data to evaluate market mood and possible dangers when sentiment scores are added to traditional financial models. This advanced understanding can help analysts provide more precise forecasts, which will ultimately strengthen risk management procedures in the financial sector.

5.5 Future Work and Recommendations

This study offers insightful information, however there are still questions to be answered. Improved sentiment analysis algorithms, new feature explorations, or the addition of more detailed sentiment categories are all potential areas for future research. Enhancing the ability to be generalized of the results could also involve looking into how well the developed models apply to various industries and market situations. It is recommended that future researchers investigate these directions to improve our understanding of sentiment-based stock price predictions.

APPENDIX

Web scrapping for Twitter Data.

```
EMAIL="YOUR_TWITTER_EMAIL_ID"
HANDLE="YOUR_TWITTER_HANDLE"
PASSWORD="YOUR_TWITTER_PASSWORD"
COMPANY="COMPANY_TO_SCRAP"
POSTS="NUMBER_OF_POST_TO_SCRAP"
import pandas as pd
from selenium.webdriver import Chrome
from selenium.webdriver.common.by import By
from seleniumbase import Driver
from selenium.webdriver.common.keys import Keys
from dotenv import load_dotenv
import os
from time import sleep

load_dotenv()
EMAIL=os.environ.get("EMAIL")
HANDLE=os.environ.get("HANDLE")
PASSWORD=os.environ.get("PASSWORD")
COMPANY=os.environ.get("COMPANY")
POSTS=int(os.environ.get("POSTS"))

browser = Driver(browser="chrome",headless=False,headless2=False,headed=True)
browser.get(r"https://twitter.com/explore")
sleep(5)
username = browser.find_element(By.XPATH,r'//*[@class="r-30o5oe r-1niwhzg r-17gur6a r-1yadl64 r-deolkf r-homxoj r-poiln3 r-7cikom r-1ny4l3l r-t60dpp r-1dz5y72 r-fdjqy7 r-13qz1uu"]')
username.send_keys(EMAIL)
```

```
next = browser.find_element(By.XPATH,r'//*[@class="css-18t94o4 css-1dbjc4n r-sdzlij r-1phboty r-rs99b7 r-ywje51 r-usiww2 r-2yi16 r-1qi8awa r-1ny4l3l r-ymttw5 r-o7ynqc r-6416eg r-lrvibr r-13qz1uu"]')

next.click()

sleep(5)

username = browser.find_element(By.XPATH,r'//*[@class="r-30o5oe r-1niwhzg r-17gur6a r-1yadl64 r-deolkf r-homxoj r-poiln3 r-7cikom r-1ny4l3l r-t60dpp r-1dz5y72 r-fdjqy7 r-13qz1uu"]')

username.send_keys(HANDLE)

next = browser.find_elements(By.XPATH,r'//*[@role="button"]')[-1]

next.click()

sleep(5)

password = browser.find_element(By.XPATH,r'//*[@class="r-30o5oe r-1niwhzg r-17gur6a r-1yadl64 r-deolkf r-homxoj r-poiln3 r-7cikom r-1ny4l3l r-t60dpp r-1dz5y72 r-fdjqy7 r-13qz1uu"]')

password.send_keys(PASSWORD)

next = browser.find_elements(By.XPATH,r'//*[@role="button"]')[-2]

next.click()

sleep(5)

search = browser.find_element(By.XPATH,r'//*[@class="r-30o5oe r-1niwhzg r-17gur6a r-1yadl64 r-deolkf r-homxoj r-poiln3 r-7cikom r-1ny4l3l r-xyw6el r-13rk5gd r-1dz5y72 r-fdjqy7 r-13qz1uu"]')

search.send_keys(COMPANY)

search.send_keys(Keys.ENTER)


data = set()

nodes = set()

sleep(5)

scroll = 2160

while len(data)<POSTS:

    tweets = browser.find_elements(By.XPATH,r'.//*[@class="css-1dbjc4n r-1iusvr4 r-16y2uox r-1777fci r-kzbkwu"]')

    added = False

    sleep(5)

    for tweet in tweets:
        if tweet in nodes:
            continue
        added=True
        try:
            date = tweet.find_element(By.TAG_NAME,"time").get_attribute("datetime")
            content = tweet.find_element(By.XPATH,r".//*[@class='css-901oao css-cens5h r-1nao33i r-37j5jr r-a023e6 r-16dba41 r-rjixqe r-bcqeeo r-bnwqim r-qvutc0']").text
            comments,retweets,likes,views = tweet.find_element(By.XPATH,r".//*[@class='css-1dbjc4n r-1kbdv8c r-18u37iz r-1wtj0ep r-1s2bzr4 r-1ye8kvj']").text.split()

            nodes.add(tweet)
            data.add((content,date,comments,retweets,likes,views))
        except Exception:
            continue
    df = pd.DataFrame(data,columns=["content","date","comments","retweets","likes","views"])
    df.to_csv(f"{COMPANY}.csv")
    browser.execute_script(f"window.scrollTo(0, {scroll})")
    scroll+=2160
    if not added:
        break
```

Web Scrapping for Stock Data.

```
import yfinance as yf
import random
from datetime import datetime, timedelta

def get_random_date(year, month):
    # Generate random day within the given year and month
    last_day_of_month = (datetime(year, month, 28) + timedelta(days=4)).replace(day=1) - timedelta(days=1)
    random_day = random.randint(1, last_day_of_month.day)
    return datetime(year, month, random_day)

def get_stock_data(stock_symbol, year):
    try:
        # Create an empty dataframe to store the combined data for the year
        combined_data_year = None

        # Fetch data for each month in the year with random start and end dates
        for month in range(1, 13):
            start_date = get_random_date(year, month)

            # Calculate the last day of the month for the start_date
            last_day_of_month = (start_date.replace(day=28) + timedelta(days=4)).replace(day=1) - timedelta(days=1)

            # Get random end date within the same month and year
            random_end_day = random.randint(1, last_day_of_month.day)
            end_date = datetime(year, month, random_end_day)

            # Download stock data for the specific month
            stock_data_month = yf.download(stock_symbol, start=start_date, end=end_date)

            # Add the company and stock names to the dataframe
            stock_data_month['Company'] = stock_symbol
            stock_data_month['Stock_Name'] = yf.Ticker(stock_symbol).info['shortName']

            # Concatenate data for the month to the combined_data_year
            if combined_data_year is None:
                combined_data_year = stock_data_month.copy()
            else:
                combined_data_year = combined_data_year.append(stock_data_month)
        return combined_data_year
    except Exception as e:
        print(f"Error fetching data for {stock_symbol} in {year}: {e}")
        return None

def main():
    # Define the stocks
    stocks = ['TSLA', 'NIO', 'F', 'XPEV']

    # Create an empty dataframe to store the combined data for all years
    combined_data_all_years = None

    # Fetch data for each stock with random start and end dates for each month in every year
    for stock_symbol in stocks:
        for year in range(2010, 2024):
            stock_data_year = get_stock_data(stock_symbol, year)
            if stock_data_year is not None:
                # Concatenate data for the year to the combined_data_all_years
                if combined_data_all_years is None:
                    combined_data_all_years = stock_data_year.copy()
                else:
                    combined_data_all_years = combined_data_all_years.append(stock_data_year)

    # Reset the index and Save the combined data to a CSV file
    combined_data_all_years.reset_index(inplace=True)
    combined_data_all_years.to_csv('random_stock_data_each_month_all_years.csv',
index=False)

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

BIBILOGRAPHY

Abiola, O., Abayomi-Alli, A., Tale, O.A., Misra, S., Abayomi-Alli, O., 2023. Sentiment analysis of COVID-19 tweets from selected hashtags in Nigeria using VADER and Text Blob analyser. Journal of Electrical Systems and Information Technology 10. https://doi.org/10.1186/s43067-023-00070-9

Agarwal, A., Vats, S., Agarwal, R., Ratra, A., Sharma, V., Gopal, L., 2023. Sentiment Analysis in Stock Price Prediction: A Comparative Study of Algorithms. https://doi.org/https://doi.org/10.1007/978-981-16-4284-5_11

Bhandari, N., 2017. Stock Market Trend Prediction Using Sentiment Analysis Senior Project.

Bing, L., Chan, K.C.C., Ou, C., 2014. Public sentiment analysis in twitter data for prediction of a company's stock price movements, in: Proceedings - 11th IEEE International Conference on E-Business Engineering, ICEBE 2014 - Including 10th Workshop on Service-Oriented Applications, Integration and Collaboration, SOAIC 2014 and 1st Workshop on E-Commerce Engineering, ECE 2014. Institute of Electrical and Electronics Engineers Inc., pp. 232–239. https://doi.org/10.1109/ICEBE.2014.47

Brown, R.L., Shevlin, T.J., 2001. STOCK MARKET EFFICIENCY AND PRICE PREDICTIONS IMPLICIT IN OPTION TRADING by.

Cristescu, M.P., Nerisanu, R.A., Mara, D.A., Oprea, S.V., 2022. Using Market News Sentiment Analysis for Stock Market Prediction. Mathematics 10. https://doi.org/10.3390/math10224255

Elbagir, S., Yang, J., 2019. Twitter Sentiment Analysis Using Natural Language Toolkit and VADER Sentiment.

Kwon, Y.-K., Choi, S.-S., Moon, B.-R., 2005. Stock Prediction Based on Financial Correlation. Washington, DC, USA,. https://doi.org/https://doi.org/10.1145/1068009.1068351

Martinez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernandez-Orallo, J., Kull, M., Lachiche, N., Ramirez-Quintana, M.J., Flach, P., 2021. CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. IEEE Trans Knowl Data Eng 33, 3048–3061. https://doi.org/10.1109/TKDE.2019.2962680

Mehta, S., Pandi (Jain), G., 2019. An_Improving_Approach_for_Fast_Web_Scrap. International Journal ofAdvanced Research in Computer Engineering & Technology (IJARCET) 8, 434–438.

Rajendiran, P., Priyadarsini, P.L.K., 2023. Survival study on stock market prediction techniques using sentimental analysis. Mater Today Proc 80, 3229–3234. https://doi.org/10.1016/j.matpr.2021.07.217

Salam, M.A., Azar, A.T., Elgendy, M.S., Fouad, K.M., 2021. The Effect of Different Dimensionality Reduction Techniques on Machine Learning Overfitting Problem. International Journal of Advanced Computer Science and Applications 12, 641–655. https://doi.org/10.14569/IJACSA.2021.0120480

Sasank Pagolu, V., Reddy Challa, K.N., Panda, G., Majhi, B., 2016. Sentiment Analysis of Twitter Data for Predicting Stock Market Movements proceedings. International conference on

Signal Processing, Communication, Power and Embedded System (SCOPES)-2016. https://doi.org/10.1109/SCOPES.2016.7955659

Schöneburg, E., 1990. Stock price prediction using neural networks: A project report, Neurocomputing.

Schröer, C., Kruse, F., Gómez, J.M., 2021. A systematic literature review on applying CRISP-DM process model, in: Procedia Computer Science. Elsevier B.V., pp. 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Sebastian, M., Wolfram, A., 2010. Modelling the Stock Market using Twitter.

Vergara, J.R., Estévez, P.A., 2014. A review of feature selection methods based on mutual information. Neural Comput Appl. https://doi.org/10.1007/s00521-013-1368-0