OPTIMIZING STOCK PRICE PREDICTION USING LONG SHORT-TERM MEMORY (LSTM) AND SENTIMENT ANALYSIS WITH LARGE SOCIAL MEDIA DATA

A dissertation
presented to the Faculty of Huddersfield University
in partial fulfillment of the requirements for the degree of
Master of Science (M.Sc.) in Computing

by

Imamuddin Shaik

Huddersfield, UK
01, 2024

ABSTRACT

Predicting stock prices is a difficult task that is influenced by dynamic financial forces. The integration of sentiment analysis of a large social media dataset, namely Dow30 Stocks, with advanced Natural Language Processing (NLP) methods, such as VADER and TextBlob, and a machine learning model, specifically Long Short-Term Memory (LSTM), is investigated in this paper. The research questions seek to understand how a large amount of sentiment data analysis improves stock market predictions, the specific improvements obtained by combining sentiment analysis and LSTM models, and the extent to which optimising large-scale data processing techniques improves accuracy.

This thesis proposes a methodology that combines two datasets from the same day, tweets, and stock data. Over 100,000 tweets for each stock were collected from Dow Stocks 30 dataset. The examination of Twitter data, followed by preparation steps. Text cleaning and tokenization are performed using multiple Python library tools, followed by feature extraction from Twitter tweets. Using two Python tools, VADER and TextBlob, the cleaned tweets are then labelled with sentiment scores. The acquired scores are weighted and summed to get an overall sentiment score, which is then fed into an LSTM model with predefined hyper parameters.

Our LSTM-based model, which we propose, demonstrates superior performance compared to existing models, specifically in forecasting McDonald's (MCD) stock prices, with a focus on the Dow Stocks 30. It obtained reduced MSE and RMSE by adjusting particular hyperparameters, indicating superior accuracy. The utilisation of cross-validation additionally enhances the R2 value for McDonald's stock from 0.821 to 0.916.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all those who have contributed to the successful completion of this dissertation. Above all, I would like to express my sincere gratitude to Dr. Zhijie Xu, my academic supervisor, for his important advice, encouragement, and insights over the whole study process. The direction and standards of this dissertation were significantly influenced by this encouragement and expertise. Furthermore, I wish to express my gratitude to the faculty members who offered valuable counsel and input throughout the different phases of my research endeavour. With sincere appreciation extended to my family and friends for their steadfast support and comprehension throughout the arduous phases of this scholarly expedition. A constant source of inspiration has been their support. In conclusion, I would like to express my gratitude to the University of Huddersfield for its provision of facilities and resources that were crucial to the accomplishment of this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

Predicting stock market is a difficult challenge due to the complex nature of the financial landscape, where numerous factors dynamically influence asset prices (Ghosh, 2022). In addition to past price changes, economic indicators, trading volumes, company earnings reports, and an ever-expanding array of supplemental data sources, financial analysts also must manage a massive dataset. In order to address this challenge, sophisticated data analytics approaches, and machine learning algorithms are applied to a large dataset. These techniques aid in the spotting of potential market trends (Ghosh, 2022).

The LSTM (Long Short-Term Memory) model, a sophisticated variant of the Recurrent Neural Network (RNN), has shown promise in handling time series data, thereby making it well-suited for predicting stock market trends (Liu et al., 2022). Moreover, stock price prediction algorithms have included sentiment analysis to capture market mood and its influence on stock prices. (Huang et al., 2021).

Several studies have investigated the use of sentiment analysis and LSTM together to predict stock prices. The use of lexicon-based approaches and sentiment analysis in stock price prediction was investigated by Fazlija & Harder (2022). To forecast stock prices based on the tone of financial news or social media content, they have employed traditional machine learning models (Fazlija & Harder, 2022). In the same way, Valle-Cruz et al. (2021) have improved a lexicon-based method by using shifted correlation analysis to find latent correlations in data, increasing the predictive performance.

The application of sentiment analysis to the prediction of stock prices has also been studied. Machine learning forecasting models, such as Attention-Integrated LSTM, were

combined with statistical indicators and trading methods to forecast stock movements based on market sentiment (Bagga & Patel, 2022). Furthermore, Huang et al. (2021) show that combining LSTM with sentiment analysis outperforms traditional machine learning models in predicting stock values.

In addition, the integration of LSTM with other techniques such as attention mechanisms and multi-feature fusion has been investigated to improve stock trend prediction models (Qiu et al., 2020). These approaches seek to denoise historical stock data, extract relevant features, and improve stock price prediction accuracy.

Combining sentiment research with stock market data has the potential to improve model predictiveness by providing a deep understanding of investor sentiment through the collection of opinions and emotional tones embedded in textual data (Guo & Li, 2019).

This thesis aims to analyse large amounts of textual data from social media posts in order to capture various opinions and emotions influencing market sentiment. Using advanced Natural Language Processing (NLP) algorithms such as VADER and TextBlob, as well as machine learning models such as LSTM.

This study aims to fill the gap between large textual data processing using Lexicon-Based Natural Language Processing (NLP) algorithms to obtain a sentimental score of text and merging the sentimental data with stock data to advance the current state of the art by achieving higher accuracy.

1.1 Motivation

The study by Pegah Eslamieh et al. (2023) utilizing the User2Vec method for stock market prediction underscores the potential impact of incorporating sentiment analysis from social

media, particularly Twitter. While their approach is successful, there is a noteworthy lack of exact accuracy values, allowing space for improvement. This creates an incentive to investigate large-scale data processing approaches, particularly those based on advanced models such as LSTM, in order to improve stock price forecast accuracy. Recognising the significance of sentiment analysis in understanding market attitudes, particularly through massive dataset analysis, is a practical and logical approach. We hope to utilise the capabilities of both techniques by integrating sentiment analysis and Long Short-Term Memory (LSTM) models. This integration aims to establish a more reliable forecasting framework by taking into account the effect of market sentiment on stock prices.

Finally, the motivation for using deep neural networks like LSTM and sentiment analysis in stock market prediction stems from their demonstrated ability to extract extensive features from data, incorporate stock sentiments, and achieve superior performance in forecasting stock prices. The merging of huge data sentiment analysis with deep learning models has yielded encouraging results, bolstering the case for utilising these sophisticated approaches in stock market prediction (Pegah Eslamieh et al., 2023).

1.2 Research Questions and objectives

Research Question 1:

How does the integration of sentiment analysis from social media, specifically Twitter, enhance the accuracy of stock market predictions?

Research Question 2:

What specific improvements in accuracy can be obtained by using sentiment analysis and Long Short-Term Memory (LSTM) models in stock price prediction?

Research Question 3:

To what extent can optimising large-scale data processing techniques for broad textual data analysis, notably sentiment analysis with LSTM models, contribute to improving the accuracy of stock market prediction predictive frameworks?

1.1.1 Objectives:

- Assess the impact of integrating Twitter sentiment analysis on stock market prediction accuracy.
- Look at the relationship between social media mood and stock price movement.
- Quantify and assess specific accuracy gains by adding sentiment analysis and Long Short-Term Memory (LSTM) models into stock price prediction.
- Compare stock market predictions with and without sentiment analysis in terms of performance metrics.
- Create and deploy optimised large-scale data processing strategies for massive textual data analysis, with an emphasis on sentiment analysis utilising LSTM models.
- Assess the effect of optimised data processing techniques on the accuracy of stock market prediction predictive frameworks.
- Identify and analyse significant aspects that contribute to observed improvements, shedding light on the relationship between data processing optimisation and forecast accuracy.

These research questions and objectives seek to provide a thorough knowledge of the function of sentiment analysis in improving stock market price prediction, as well as shed light on its possible benefits, problems, and consequences for financial decision-making.

1.3 Limitations of Study

This study is limited to the use of Long Short-Term Memory (LSTM) models in conjunction with Lexicon-Based Sentiment Analysis methodologies for stock market trend prediction. The study limits its investigation to these specific strategies on purpose, emphasising their potential.

Furthermore, the study is limited by the use of Lexicon-Based Sentiment Analysis, which recognises its limits in capturing complex qualitative settings. Furthermore, the study focuses entirely on social media data for sentiment analysis, noting that sentiments expressed on these platforms may be influenced by external influences and may not fully represent general market mood.

These limits define the study's exact boundaries, providing clarity on the constraints to which its conclusions apply. The project's scope is defined by an intentional concentration on LSTM models, Lexicon-Based Sentiment Analysis, and large social media data, emphasising the importance of cautious interpretation within this constrained context.

1.4 Findings of the Research

The outcomes of the study show a comprehensive current machine learning model Long Short-Term Memory (LSTM) model based on textual data analysis with Lexicon-Based Sentiment Analyzers and merging it with stock data with deep feature analysis for stock price prediction. The study concentrated on well-known corporations such as IBM, McDonald's (MCD), Procter & Gamble Co, and Nike. Initial evaluations demonstrate the models' performance, highlighting metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R2) values. The performance of the revised models shows that

incorporating sentiment scores considerably increases predicted accuracy. Cross-validation appears as a critical strategy for improving model robustness and dependability across varied datasets.

Notably, McDonald's (MCD) stock model stands out with improved accuracy, displaying progress over previous attempts. These findings add to a better understanding of the intricacies of stock price prediction, giving useful insights for both academic and practical applications.

1.5 Learnings from the Project

Working on this project has given me experience integrating multiple datasets, specifically combining sentiment analysis from Twitter with stock data to provide insights into financial market dynamics. I honed my skills in advanced Natural Language Processing techniques, employing tools such as VADER and TextBlob for large-scale sentiment analysis.

The project sharpened my skills to preprocess unstructured textual material as well as design and optimise a Long Short-Term Memory (LSTM) model, changing hyperparameters for increased accuracy. Formulating and researching research topics helped me better comprehend the relationship between sentiment analysis and stock market forecasting. Furthermore, using cross-validation approaches improved model performance, resulting in a well-rounded skill set in data science, machine learning, and research methodology.

1.6 Thesis Outline

The thesis begins with Chapter 1, which introduces the research topics, discusses objectives, and establishes the organisation. In Chapter 2, the Background Study and Literature Review navigates historical viewpoints, explores the evolution of sentiment analysis, and assesses existing models, finding research needs. Chapter 3 describes the research approach,

sentiment and stock data processing, and the development of an LSTM model with hyper parameters.

Following this, Chapter 4 offers empirical results, evaluates stock price predictions, and analyses optimisation impact with cross validation. Following that, we will compare the sentiment score insights to the stock close price. The fourth chapter concludes with a comparison of the suggested model to a related study.

Finally, Chapter 5 Conclusion summarises major findings, analyses implications, recognises limits, and offers future research areas. It concludes the theory by providing practical insights for stock market analysts.

CHAPTER 2 : BACKGROUND STUDY AND LITERATURE REVIEW

In this section, we will look at the background research and current stock market prediction methodologies. This chapter is broken into the sections listed below. Section 2.1 addresses the theoretical basis of stock markets, followed by Section 2.2, which investigates several methods for predicting stock values. Machine learning, brain networks, recurrent neural networks, and Long Short-Term Memory (LSTM) models are discussed in sections 2.3 to 2.6. Section 2.7 delves into the role of sentiment analysis, while Section 2.8 presents critical evaluation criteria. Sections 2.9 to 2.11 discuss relevant studies, offer problems, and expose the findings of the literature study, providing an organised roadmap for our investigation.

2.1 Stock market prediction hypothesis

2.1.1 Efficient Market Hypothesis

The Efficient Market Hypothesis (EMH) emphasises that stock prices, representing all available information, follow a random walk pattern in stock market forecast. EMH believes that consistently beating the market is unachievable, calling into question the use of historical stock prices for future predictions (Gramatovici & MORTICI, 2018).

2.1.2 Adaptive Markets Hypothesis

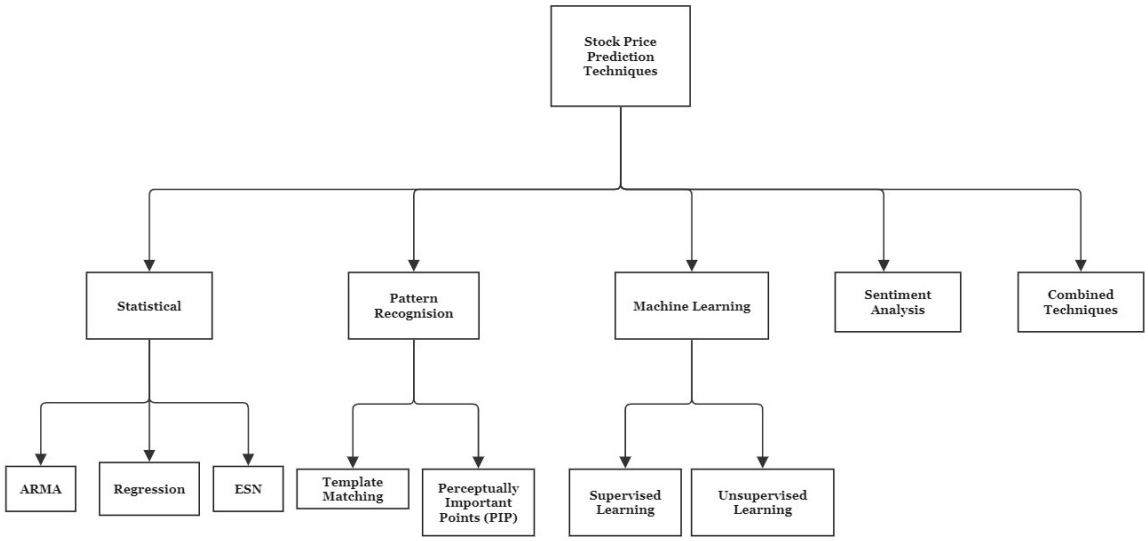
This hypothesis acknowledges the dynamic nature of stock markets as well as the impact of changing conditions on predictability (Pinar et al., 2019). It emphasises the cyclical nature of market efficiency, attributing it to factors such as market competition, profit possibilities, and participant adaptability (Trung and Quang, 2019).

2.2 Stock Price Prediction Techniques

Stock price prediction is a difficult task that requires the use of techniques from statistics, pattern recognition, machine learning, and sentiment analysis (Kara et al., 2011). Analysts develop models using past price data, technical indications, and market sentiment, as shown in Figure 1, in order to provide insight into possible price patterns. The overarching goal is to enable informed forecasting, allowing investors and traders to make strategic decisions in the volatile and unpredictable world of financial markets (García-Vega et al., 2020).

Figure 1

Stock Price Prediction Techniques



2.2.1 Limitations of Traditional Stock Price Prediction Techniques

Traditional methods of predicting stock market prices, such as fundamental and technical analysis, have been widely used in financial markets. Fundamental analysis is the process of

forecasting a company's future performance by examining its financial health, industry trends, and macroeconomic factors (Johnston, 2009).

Technical analysis, on the other hand, is based on reviewing past stock price charts and trade volumes to discover patterns and trends (Atmeh & Dobbs, 2006). These old methods, however, have several disadvantages. The following are the primary limitations:

- Limited Modeling Capability
- Market Efficiency Assumption
- Human Bias and Emotion
- Inability to Process Big Data

In contrast, the limits of traditional stock market forecast tools highlight the importance of implementing new and improved procedures. Because financial markets are dynamic and complicated, techniques that can adapt to changing conditions, identify detailed patterns, and leverage the power of enormous datasets are required (Qarni et al., 2019).

2.3 Machine Learning for Stock Price Prediction

2.3.1 Machine Learning Algorithms

Machine learning algorithms are a collection of technologies that allow computers to learn from data and make predictions or judgements without having to be explicitly programmed for each task. These algorithms are used to make sense of complex datasets and can generalise from examples (Athey & Imbens, 2019).

In machine learning, raw data is fed into algorithms, which then analyse the data to detect patterns. Once patterns are identified, the system can generate predictions based on new input, and the information acquired can be saved for future use or model refinement. This iterative

approach improves the system's capacity to anticipate accurately over time (Jordan & Mitchell, 2015).

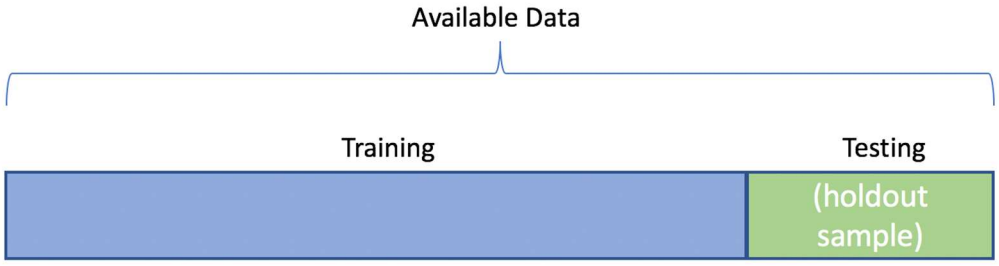
Figure 2

Machine Learning Algorithm Steps

As illustrated in Figure 3, machine learning models generally work in two phases: training and testing, which are used to determine the algorithm's performance. Each of these phases necessitates the collection of data sets (Deo, 2015).

Figure 3

Training and Test Data Split in Machine Learning



The goal of machine learning is to create systems that can learn and improve on their own. There are several types of machine learning algorithms, however they can be broadly divided into two categories: both guided and unsupervised learning.

2.3.2 Supervised Learning for Stock Price Prediction

Supervised learning includes training algorithms on labelled datasets, where input data is coupled with output labels. This allows the algorithm to generalise patterns from the training data in order to make accurate predictions on fresh, previously unknown data.

Linear Regression for regression tasks and classification techniques such as Support Vector Machines (SVM) and Neural Networks for categorization challenges are common supervised learning algorithms. There are various supervised learning algorithms with good stock price prediction scores. It demonstrates that these models can forecast with reasonable accuracy.

2.3.3 Unsupervised Learning

Unsupervised learning, on the other hand, involves unlabelled data and requires the algorithm to discover patterns and relationships in the input data without explicit instruction. The programme investigates the data's fundamental structure, grouping related data points or lowering the dataset's dimensionality.

2.4 Deep Neural Networks for Stock Price Prediction

2.4.1 What are Neural Networks?

Neural networks, whether organic or artificial, are based on their ability to recognise underlying relationships in data through a process that mimics the functions of the human brain (Tuna, 2019).

Figure 4

Neuron in Deep Neural Network (Rana, 2021)

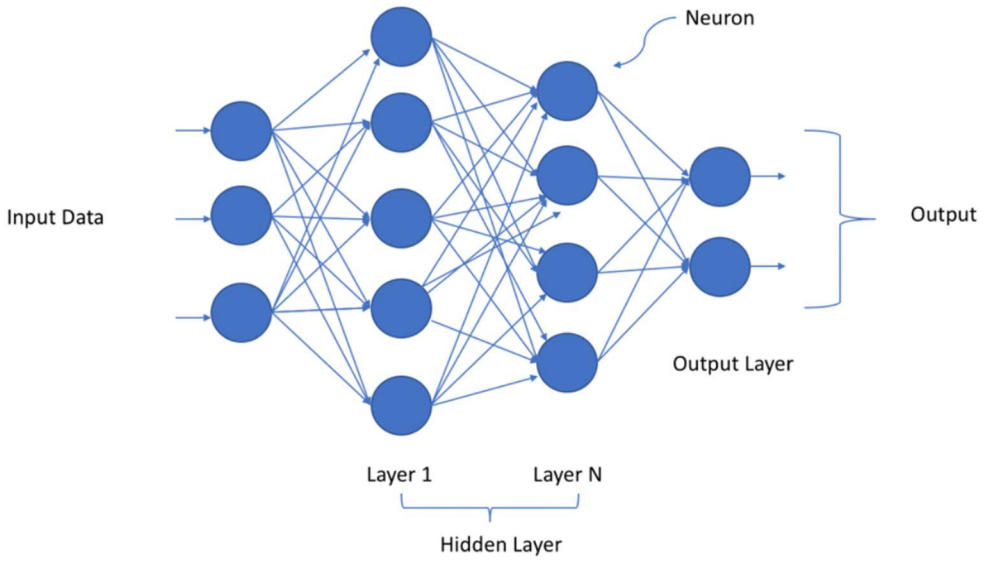


Figure 4 shows a simple feed-forward neural network. When calculating the outputs in this type of network, no information is ever transmitted backwards. The following components make up neural networks.

2.4.1.1 Neuron

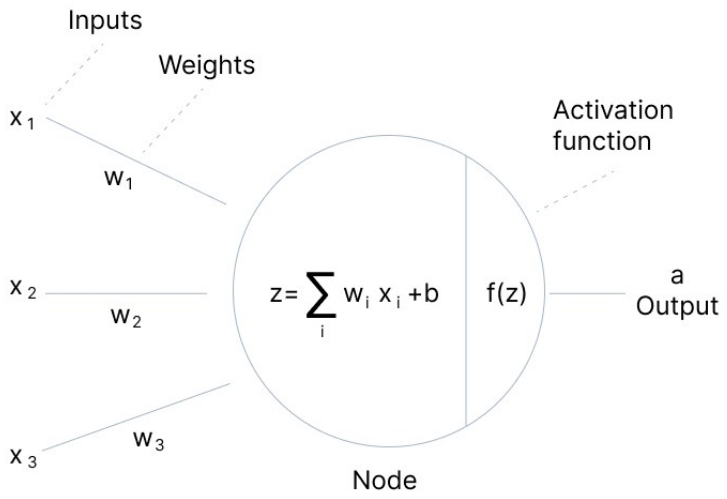
A neuron, also known as a node, is the most fundamental component of a neural network. It takes one or more inputs from other nodes or from a source and produces an output. It learns from data by modifying weights during training, allowing the network to recognise complicated patterns and predict future outcomes. The structure of a deep neural network for complex information processing involves layers of interconnected neurons.

2.4.1.2 Activation Function

Activation functions are important components of neural networks because they bring non-linearities to the model, allowing it to learn complicated patterns and relationships in data by creating functions that vary the different weights and inputs, as seen in Figure 5. By converting the neuron's output into the weighted sum of its inputs and bias. Sigmoid, tanh, and ReLU are examples of common activation functions. During training, the network's nonlinearity allows it to learn complicated patterns and relationships in the data.

Figure 5

Activation Function in Neural Network (*Baheti, 2021)*



Types of activation functions:

- Sigmoid Activation Function
- Hyperbolic Tangent (tanh) Activation Function
- Rectified Linear Unit (ReLU)

2.4.1.3 Weight

The strength of connections between inputs and neurons is represented by weights. These weights are modified during training to improve the model's performance. They calculate the effect of each input on the neuron's output, allowing the network to learn and adapt to data patterns.

2.4.1.4 Input Layer

Figure 5 depicts how a neural network's input layer gets raw data as input characteristics. Every neuron in the input layer corresponds to a distinct feature, resulting in a one-to-one mapping.

2.4.1.5 Hidden Layer

In Figure 5, the hidden layer of a neural network processes input from the input layer via weighted connections. Neurons in the buried layer use activation functions to convert weighted inputs into non-linear representations.

2.4.1.6 Output Layer

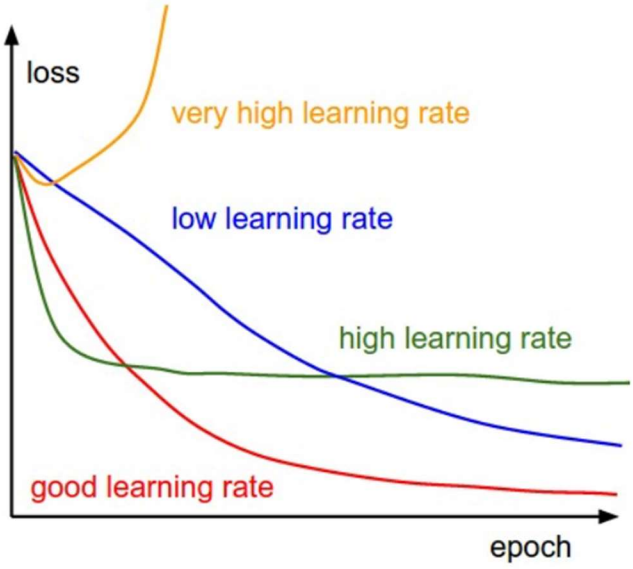
Figure 5 shows the output layer of a neural network, which generates the final predictions or classifications. Each neuron in the output layer represents a different class or prediction outcome.

2.4.1.7 Learning Rate

In a neural network, the learning rate is a hyperparameter that influences the size of steps taken during the optimisation process. It affects how rapidly the model adjusts its weights in response to computed gradients. Figure 6 depicts the various learning rates in a graph plot of training loss and number of epochs.

Figure 6

Various Learning Rates in Machine Learning (Dabbura, 2017)



Various Learning Rates in Machine Learning

2.5 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are distinguished from other neural networks by their feedback mechanism, which enables them to excel in processing sequential input with dependencies (Ohno & Kumagai, 2021). RNNs, unlike classic feedforward networks, have a feedback loop that maintains information, making them ideal for applications such as time-series analysis, speech recognition, and natural language processing (Ahmad et al., 2023).

The key concept of an RNN is its hidden state, which acts as memory and is updated at each sequence step to accrue knowledge from previous inputs (Rusch & Mishra, 2021). This

feedback mechanism enables the network to learn patterns, allowing for a more nuanced grasp of temporal connections and context in sequential input (Zhang et al., 2021).

2.5.1 Challenges in RNN

Hochreiter and Schmidhuber (1997) introduced the vanishing gradient problem in Recurrent Neural Networks (RNNs), limiting their ability to capture long-term dependencies. The ballooning gradient problem was discovered as a key challenge at the same time (Pascanu et al., 2013). To address these concerns, sophisticated topologies like as Long Short-Term Memory (LSTM) networks have been proposed. Hochreiter and Schmidhuber (1997) invented LSTMs, which feature a gating mechanism to selectively retain or discard input, overcoming the disappearing and ballooning gradient concerns in simple RNNs.

LSTMs outperformed typical RNNs in tasks involving long-term dependency capture, such as speech recognition and language modelling, according to Graves et al. (2013). This enhancement was ascribed to LSTMs' capacity to maintain stable gradients over time, thereby addressing the issues that basic RNNs faced.

Furthermore, new research by Greff et al. (2017) has shed light on the limitations of basic RNN architectures by providing insights into the underlying mechanics of vanishing and expanding gradients in RNNs. This paper emphasises the significance of employing advanced architectures such as LSTMs to overcome these issues and effectively describe long-term dependencies in sequential data.

2.6 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a version of a recurrent neural network (RNN) that has received a lot of interest because of its potential to solve the vanishing and exploding gradient

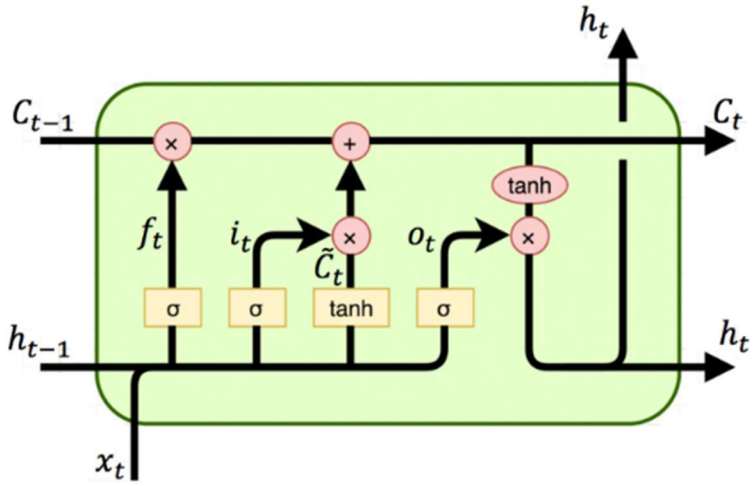
problems that regular RNNs have (Yu et al., 2019). It was developed to address the constraints of traditional RNNs in capturing and keeping long-term dependencies in sequential data (Berman, 2019). The LSTM architecture incorporates memory cells capable of retaining information over lengthy sequences, allowing it to capture long-range dependencies more effectively than typical RNNs (Hilal et al., 2023). This is accomplished through the use of gating mechanisms, which govern the flow of information into and out of memory cells, allowing the network to preserve critical information while discarding irrelevant facts (Yu et al., 2019).

One of the most significant advantages of LSTM over standard RNNs is its capacity to keep long-term memory, which is critical for tasks involving sequential data such as natural language processing, speech recognition, and time series prediction (Berman, 2019). The capacity of LSTM to store and retrieve information across long time lags makes it especially well-suited for applications requiring long-term dependencies (Hilal et al., 2023). Furthermore, LSTM networks are easier to train than normal RNNs because they are less vulnerable to the vanishing gradient problem, which can impede learning in typical RNNs (Berman, 2019).

2.6.1.1 Working of LSTM Architecture

Figure 7

Working of LSTM Architecture



f_t, i_t, O_t : *Forget gate, Input gate and Output gate vectors.*

h_t , C_t : *hidden layer and candidate layer vectors*

x_t : *input vector*

b_f , b_i , b_c, b_o : *bias vectors*

W_f , W_i , W_c, W_o : *represent weight matrices*

σ , tanh : *activation functions.*

The Forget Gate (f_t) in figure 7 starts the process by deciding whether to keep or discard information from the previous memory state (C_t-1) for the current time step. It works by

applying a sigmoid activation function to the concatenation of the current input (X) and the prior hidden state (H_t-1), resulting in a forget gate vector (f_t) with values ranging from 0 to 1.

Simultaneously, the Candidate Layer (C_t) constructs a candidate vector using the hyperbolic tangent (Tanh) activation function and the previous hidden state (H_t-1). C_t, the resulting candidate layer vector, is between -1 and 1.

The Input Gate (i_t) then decides which values from the candidate layer should be included into the memory state. It works similarly to the Forget Gate, employing a sigmoid activation function and providing an input gate vector (i_t) with values ranging from 0 to 1.

The Output Gate (O_t) regulates information exposure from the current memory state (C_t) to the next hidden state. It, like the previous gates, employs a sigmoid activation function, resulting in an output gate vector (O_t) with values ranging from 0 to 1.

The Memory State (C_t) is a dynamic storage location for context or memory from the input sequence. It proceeds through processes including element-wise multiplication and addition, which are impacted by the forget and input gates, to produce the current memory state (C_t).

Finally, the Hidden State (H_t) represents information distilled from the memory state by the output gate. The current hidden state (H_t) is acquired using element-wise multiplication and activation functions.

The working flow of the LSTM includes an initialization stage in which the initial hidden state (H_0) and memory state (C_0) are set. Following that, the LSTM takes inputs and computes outputs at each time step during the forward pass. The computed hidden and memory

states are then passed back into the next time step, resulting in a recursive loop that repeats for each sequence.

2.7 Sentimental Analysis for Stock Price Prediction

Sentiment analysis for stock price prediction entails analysing and interpreting market sentiment reflected in financial news, social media such as Twitter, or other textual sources. The theory is that public opinion and emotions expressed in writing can have an impact on stock prices.

Furthermore, as stated by Wang et al. (2021) and (Bai & Sun, 2022), the use of social media sentiment analysis in stock market trending analysis has contributed to greater accuracy of stock price predictions. Patil et al. (2021) proposed an approach for accurate stock market forecasting based on sentiment analysis and data analytics, emphasising the importance of sentiment analysis in financial forecasting.

2.7.1 Methods in Sentimental Analysis

In stock price prediction, sentiment analysis entails extracting sentiments from textual data such as financial news and social media. Among the most important methods are:

1. **Text Mining Techniques:**
    - ***Text Preprocessing:*** Remove noise and standardise format by doing activities like as tokenization and stemming.
    - *Bag-of-Words (BoW):* Text should be represented as an unordered set of words, with word frequency as a feature.
2. **Machine Learning Models:**

- *Supervised Learning:* Models can be trained on labelled datasets using techniques such as SVM, Naive Bayes, and logistic regression.
- *Deep Learning:* Use neural network topologies (for example, RNNs and LSTMs) to capture detailed patterns in sequential data.

3. **Lexicon-Based Approaches:**
    - *Sentiment Lexicons:* Sentiment lexicons are collections of phrases and words labelled with the sentiment (positive, negative, or neutral) associated with them to facilitate sentiment analysis in natural language processing.
    - *VADER:* A lexicon-based tool that is particularly useful for analysing feelings in social media text.
4. **Hybrid Approaches:**
    - *Combining Text and Numerical Data:* Integrating sentiment scores with standard financial indicators improves forecasting ability.
    - *Ensemble Methods:* Combine predictions from several models, such as sentiment analysis and technical or fundamental analysis, to improve forecasting accuracy.

2.7.2 Lexicon-Based Approaches

Natural language processing (NLP) techniques based on lexicons have grown in popularity due to their simplicity, computational efficiency, and flexibility to varied domains (Ding et al., 2008). These methods rely on preexisting dictionaries or lists of terms, which simplifies computations and requires less processing resources (Ding et al., 2008). They are ideal for scenarios with minimal labelled data or when advanced models are impracticable (Oliveira et al., 2016).

2.7.3 Text Blob

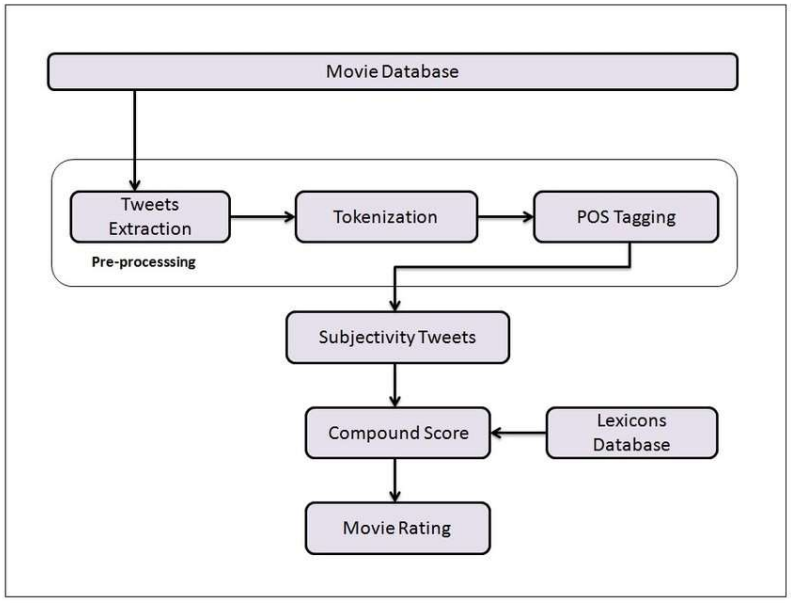
Text Blob is a Python module that provides a simple API for typical natural language processing (NLP) activities. It is based on NLTK (Natural Language Toolkit) and provides a user-friendly interface for tasks including part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, and translation.

2.7.4 VADER

VADER (Valence Aware Dictionary and Sentiment Reasoner) is an NLTK module sentiment analysis tool that uses a lexicon and rules to analyse social media text. It excels at analysing short and informal texts, making it useful for extracting sentiment from financial news, tweets, and other sources. VADER assigns a sentiment score to each piece of text based on its polarity (positive, negative, or neutral) and intensity. Figure 8 shows an example of working of VADER Sentimental Analyser with steps like Tokenization, POS tagging compound score etc.

Figure 8

Working of VADER Sentiment Analyser



Researchers can include the mood of financial news articles and social media posts into stock price prediction models by including VADER and Text Blob. VADER sentiment ratings are used as features in prediction models, assisting in capturing the market's emotional tone.

spaCy for Feature Extraction

spaCy is an open-source natural language processing (NLP) package developed for efficient and quick text processing. While spaCy is typically used for tasks like tokenization, part-of-speech tagging, named entity identification, and dependency parsing, it may also be utilised for sentiment analysis feature extraction.

2.8 Evaluation Metrics

The accuracy of the model must be evaluated at each stage of the machine learning process. The Mean Squared Error, Mean Absolute Error, Root Mean Squared Error, and R-Squared or Coefficient of Determination metrics are used in regression analysis to evaluate the model's performance. The following are the most commonly used metrics for evaluating regression models.

Mean Squared Error (MSE)

The average of the squared deviations between expected and actual values is calculated, demonstrating prediction accuracy.

$$MSE = n1 \sum i = 1n(yi - yi)2$$

Root Mean Squared Error (RMSE)

It is a regression model's performance indicator. It computes the average difference between the anticipated and actual values of a model. It gives an estimate of how well the model predicts the target value.

$$RMSE = n1 \sum i = 1n(yi - yi)2$$

R-squared (R^2)

Indicates the proportion of variance explained by the model in the dependent variable, with larger values indicating better fit.

$$R2 = 1 - \sum i = 1n(yi - y^{-})2 \sum i = 1n(yi - yi)2$$

Where:

- n is the number of observations.
- y_i is the actual value for observation *i*.
- $\overline{y_i}$ is the predicted value for observation *i*.
- $\bar{y}$ is the mean of the actual values.

2.9 Related work

Several research have found a strong relationship between sentiment analysis and market trends, emphasising the inadequacy of predicting stock prices only on historical or textual data (Yashmita & D, 2023). An and Chan (2016) presented a unique methodology for short-term stock price prediction based on limit order book dynamics, which performed rather well in data analysis (An & Chan, 2016).

Similarly, S* et al. (2020) presented a hybrid deep learning strategy to anticipate stock price trends that blends historic price-based trend forecasting with stock market opinions stated on Twitter (S* et al, 2020). These hybrid approaches demonstrate the possibility of combining several data sources and analytical tools to improve prediction accuracy.

Both Singamaneni et al. (2022) and Jaggi et al. (2021) used sentiment analysis models to link public sentiment to stock price changes. Natural Language Processing (NLP) techniques were used by Zhang et al. (2018) to extract indications of public mood from social media for stock price prediction. Similarly, Gite et al. (2021) proposed using machine learning techniques in conjunction with sentiment research to reliably anticipate stock values.

Fan and Chen (2022) discovered that combining news sentiment and social sentiment improved prediction accuracy. By combining investor sentiment and functional principal component analysis, the study used a novel approach to forecasting stock values in the real estate business. Based on stock bar comments and the Baidu search index, three real estate stocks—Wanke A, Hua Qiaocheng A, and Greenland Holdings—were chosen for sentiment research. The stock trade data comprised 11 indicators for each stock and spanned 243 trading days from January 1, 2021 to December 31, 2021. The process included assessing sentiment, using functional principal component analysis to reduce dimensionality, and using a convolutional neural network (CNN) to forecast stock price. The results of functional principal component analysis indicated that stock indicators were accurately represented. The proposed methodology sought to improve real estate stock price projections by taking both sentiment and functional data aspects into account.

Bouktif et al. (2020) used enhanced textual features to develop a unique technique to stock market prediction. They overcame the constraints of prior methods, outperforming sentiment-based approaches, including deep learning, with an astonishing 60% accuracy in predicting stock movements. The favourable contributions of augmenting textual elements to stock market prediction were obvious, leading to the conclusion that public opinion conveyed on Twitter could impact stock prices, highlighting the importance of sophisticated sentiment research methodologies. Future research directions emphasised the necessity of improving sentiment representation and investigating sophisticated machine learning methods, notably pushing for deep learning versions for sentiment feature engineering and prediction modelling.

Pagolu et al. (2016) sought to forecast stock market movements using sentiment analysis of Twitter data, with a specific focus on Microsoft. Their approach used machine learning using Word2vec and N-gram representations, resulting in a sentiment analyzer with sentiment classification accuracies of 70.18% (Word2vec) and 70.49% (N-gram). The study handled the analysis as a classification problem, correlating these attitudes with Microsoft's stock prices, and attained accuracies of 69.01% (Logistic Regression) and 71.82% (LibSVM), demonstrating a meaningful association between Twitter sentiments and stock market patterns.

Wang and Zhu (2023) were inspired to create an investor sentiment index based on deep learning and machine learning in order to improve market trend predictions. Over the course of six years, they processed data from the SSE 50 and CSI 300 Index Forums to generate a sentiment index, which greatly increased prediction accuracy, particularly with the suggested CNN-SVM model outperforming others. The study emphasised the importance of investor sentiment in stock forecasting. Future work ideas include expanding sentiment indexes to other

time series studies and applying them to fields such as carbon pricing and exchange rate forecasting. Although highlighted research limitations were not explicitly stated, prospective areas for further investigation were hinted at, such as sentiment analysis in specific market conditions or including more external components for a more thorough predictive model.

Pegah eslamieh et al. (2023) used the User2Vec approach for stock market prediction, incorporating Twitter sentiment research. The authors created the model by training it on financial market-related tweets using the Skip-gram algorithm. The User2Vec model excelled in forecasting stock trends and extracting insights from social media. The study used several machine learning techniques, including deep learning models, to improve prediction accuracy by emphasising user-level input. Although no exact accuracy statistics for User2Vec were supplied, the results demonstrated its efficacy, with AAPL achieving the highest accuracy and the mean accuracy for other stocks above 60%.

Heiden and Parpinelli (2021) investigated the impact of financial news on stock price by combining New York Times news data with sentiment analysis using the VADER framework, providing sentiment scores that enhance the features for LSTM model training. Over a 5-year period, the performance of the top 10 assets in the S&P 500 index is analysed. The results reveal that integrating feelings as features is clearly superior, with a reduced RMSE of 15.76 showing good model calibration and consistency. The model displays its ability to forecast stock prices up to 50 days in advance, while limitations emerge beyond this horizon, implying difficulties in forecasting prices with limited data points.

Table 1

Research Papers used ML models and Dataset Types for Stock Price Prediction

Research Papers	**Machine Learning Models**								**Dataset Type**		
	LSTM	**ARIMA**	**CNN**	**MLP**	**DBN**	**Random Forest**	**SVM**	**MLP**	**Stock Data**	**Social Media Date**	**News Data**
Hossain et al. (2018)	X								X		
Heiden & Parpinelli (2021)	X								X		X
Lv et al. (2019)					X		X		X		
Singamaneni et al. (2022)	X				X				X		
Ding et al. (201			X						X		X
Rather et al. (2015)	X	X							X		
Liu et al. (2020)	X		X						X	X	
Pang et al. (2020)	X								X		
Mittal et al. (2012)	X						X		X	X	
Bollen et al. (2011)							X		X	X	

Di Persio et.al (2017)	X								X		
Milosevic et al. (2016)							X		X		
Khashei & Hajirahimi (2017)	X							X	X		
Gite et al. (2021)	X								X		
Pegah eslamieh et al. (2023)	X		X			X			X	X	
Wang and Zhu (2023)			X						X	X	

According to the findings, the LSTM model was used in the majority of the publications' proposals. The LSTM has delivered outstanding results in stock price prediction. As a result, integrating the LSTM with sentimental analysis is being seriously investigated. This strategy will provide more precise predictions.

2.10 Challenges in Current Work

2.10.1 Research Gaps

According to the literature review, several machine learning algorithms produce good results, and LSTM is one of the best, but there are a few obstacles in the current state of the art in stock price prediction. They are described further below.

1) **Use of Large Sentiment Data with LSTM**: In the current study, combining a big sentiment dataset with Long Short-Term Memory (LSTM) models presents difficulties. For starters, the sheer volume of emotion data might increase computing complexity and resource demands. Training LSTM models on large data sets necessitates a significant amount of processing power and memory, potentially boosting the efficiency of the models described by Heiden and Parpinelli (2021).
2) **Implementation of Various Lexicon Based Analysis**: For sentiment analysis, it is necessary to properly leverage diverse lexicons. For starters, developing a cohesive framework requires maintaining and harmonising several lexicons with varied structures and scales. Integrating lexicons with different linguistic subtleties necessitates thorough preprocessing to assure compatibility, and meaningful synthesis contributes to the models proposed by Pegah eslamieh et al. (2023)

2.11 Outcome of Literature Review

In this chapter, we examined the necessary knowledge for stock price prediction as well as several machine learning algorithms, sentiment analysis, and their combined methodologies used to improve stock price prediction accuracy. However, our research found limitations in the

existing literature, particularly in the integration of huge sentiment datasets with LSTM models and the implementation of varied lexicon-based analysis for stock price prediction.

The study's findings highlight the vital relevance of addressing these problems, particularly in using huge data sets proposed by Pegah eslamieh et al. (2023) to be used for Lexicon Based Sentiment Analyzer and integrating the result with LSTM model. By filling these gaps, the study hopes to improve model accuracy, as proposed by Heiden & Parpinelli (2021) and Pegah eslamieh et al. (2023). Because a huge amount of data is employed, this strategy can assist achieve higher accuracy than based models.

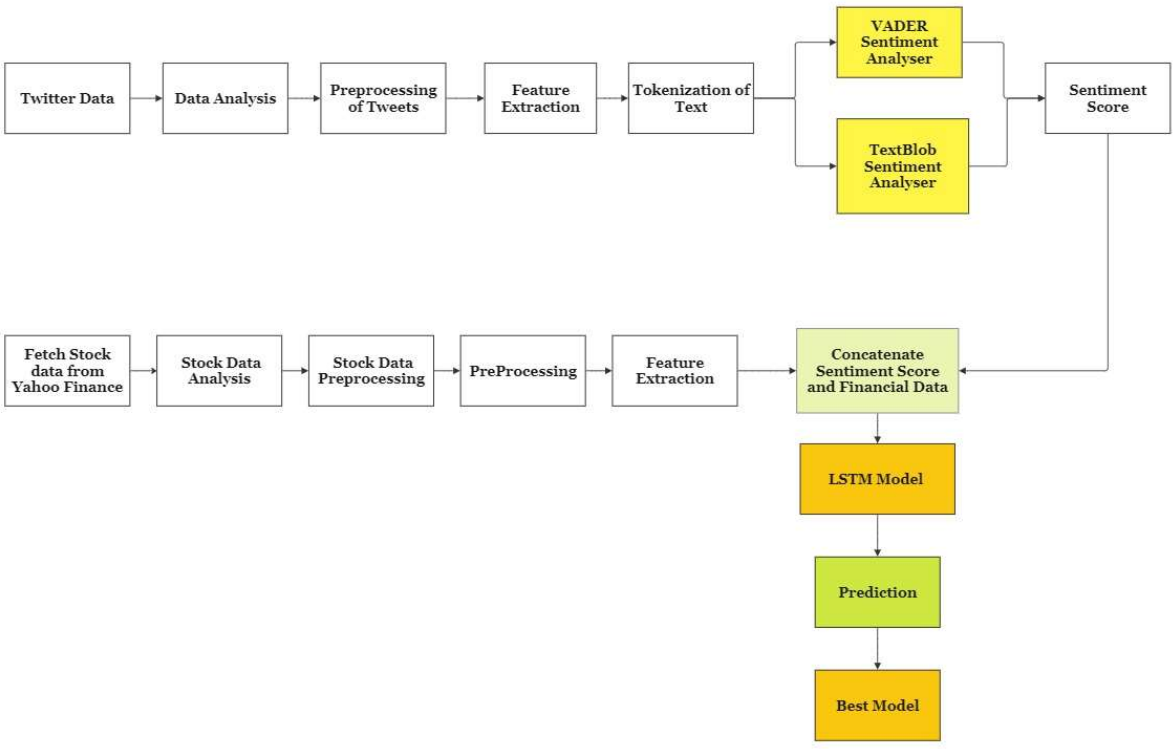
CHAPTER 3 : METHODOLOGY

The purpose of this chapter is to offer a detailed examination of the datasets and machine learning models that we employed in this thesis. The machine learning models utilised will be described in detail. This chapter is organised into five sections. Section 3.1 focuses on the social media dataset and the details of data cleaning and feature extraction; Section 3.2 focuses on the social media dataset's data feature selection; Section 3.3 discusses the technique used for dataset splitting and cross validation of the social media dataset; and Section 3.4 discusses the financial stock dataset.

3.1 Proposed Model

Figure 9

Proposed Model for Stock Price Prediction



The proposed model combines two datasets, namely tweets and stock data, both extracted on the same day. The Twitter dataset contains over 100,000 tweets for each stock during a two-year period. The investigation of Twitter data is the first step, followed by preparation measures. This involves text cleaning and tokenization using several tools from the Python library, followed by feature extraction from twitter messages. The cleaned tweets are then tagged with sentiment ratings using two Python tools, VADER and TextBlob. The obtained scores are weighted and averaged to produce an overall sentiment score.

On the other hand, stock data is acquired from the Yahoo Finance website utilizing the yfinance library within the Python ecosystem. Data analysis is performed to eliminate extraneous information, and relevant stock features are extracted.

Finally, based on the date, the stock and sentiment data are combined. Following that, correlation analysis and feature selection are carried out. The combined dataset is then placed into an LSTM (Long Short-Term Memory) model to forecast the close stock price.

3.2 Sentiment Data

3.2.1 Data Collection

As we discussed in Chapter 2 section 2.10, a vast amount of social media data is required to predict stock prices more effectively. Twitter is frequently referred to be a public platform, and numerous APIs have been developed to collect vast amounts of data. We chose Dow-30 Stocks tweets as our social media data because they comprise approximately 100,000 tweets for various stock companies (Eslamieh et al. 2021). This dataset was compiled between the years 2017 and 2019, and it includes a wide range of tweets linked to Dow Jones equities. The dataset

includes information such as the tweet ID, URL, username, text content, number of followers, date in UTC, retweets, favourites, mentions, hashtags, and geolocation.

We have chosen four stocks for our stock price prediction: MCD (McDonald's), IBM, PG (Procter & Gamble Co), and NKE (Nike). Using some basic sentimental analysing algorithms, the dataset already includes some attributes such as sentiment, polarity, and subjectivity. We will use the contemporary sentiment analysis tools VADER and Text Blob, which were mentioned in the previous chapter.

3.2.2 Data Exploration

The following columns were present in the initial data: author_id, Date, date_time, favourites, followers_count, friends_count, geo, hashtags, id, listed_count, mentions, name, permalink, polarity, replies, retweets, screen_name, sentiment, subjectivity, text. The majority of the columns were filled with nulls and 0s. Following the initial analysis, certain columns are removed.

Checking the Selected data:

Following the initial cleaning, 8 columns are chosen for study. Date, author_id, name, content, followers_count, sentiment, polarity, and subjectivity are among them.

As seen in the picture, the tweets generally provide information on stock shares and sales of those stocks for a stock business, which will aid in gaining insights into the stocks and making a good prediction.

Figure 10

Dataset top 5 rows analysis of Tweeter Dataset

```
[ ] df.head()
```

	Date	author_id	name	text	followers_count	sentiment	polarity	subjectivity
0	2017-01-03	4801718472	Food Processing News	Vatican #McDonalds divides opinion. Read more:...	501.0	1	0.500000	0.500000
1	2017-01-03	2840223957	Jakub Kapusnak	$MCD interesting level to watch, does it find ...	1182.0	1	0.233333	0.333333
2	2017-01-03	4885758668	markcarson16	$AMD Company Info Updated Tuesday, January 3, ...	226.0	0	0.000000	0.000000
3	2017-01-03	4739699834	32Trades us #WeMakeItRain	$MCD Clean Up Crew Gonna Be Busy http://www.32...	3235.0	1	0.233333	0.500000
4	2017-01-03	748817972970463232	finzine	$MCD Controversial McDonald's Opens on Vatican...	204.0	1	0.550000	0.950000

Checking for Data set Information:

The Data Frame has 485,739 entries and 8 columns, as seen in the picture. Each column is specified in terms of data type and non-null count. 'Date' (object), 'author_id' (int64), 'name' (object), 'text' (object), 'followers_count' (float64),'sentiment' (int64), 'polarity' (float64), and'subjectivity' (float64) are among the columns.

Figure 11

Dataset Information Analysis of Tweeter Dataset

```
#Checking the data types and null values in the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485739 entries, 0 to 485738
Data columns (total 8 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   Date             485739 non-null  object
 1   author_id        485739 non-null  int64
 2   name             485733 non-null  object
 3   text             485739 non-null  object
 4   followers_count  485739 non-null  float64
 5   sentiment        485739 non-null  int64
 6   polarity         485739 non-null  float64
 7   subjectivity     485739 non-null  float64
dtypes: float64(3), int64(2), object(3)
memory usage: 29.6+ MB
```

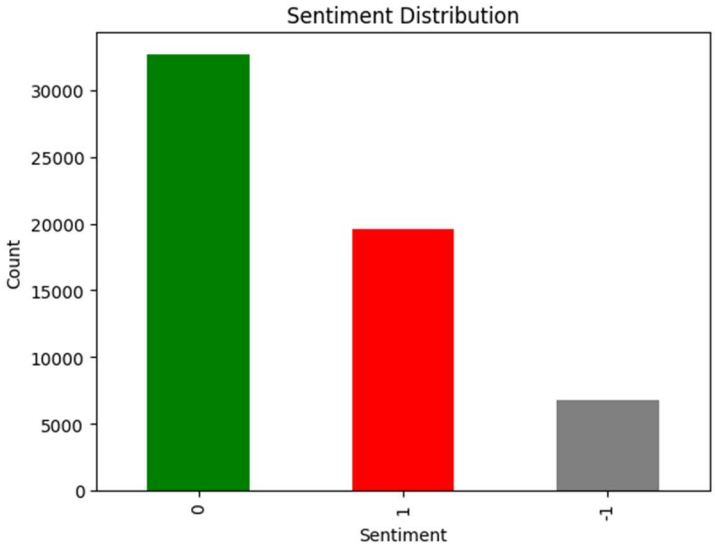
3.2.3 Data Visualization

Sentiment Distribution in Stock Tweets

The bar graph counts the number of positive (1), negative (-1), and neutral (0) sentiments. It is observed in the figure 12 that most of the dataset has the neutral labels and then it is followed by positive and then negative. Understanding sentiment distribution is critical for sentiment analysis in stock price prediction since it provides insights into market sentiment.

Figure 12

Sentiment Distribution of Tweeter Dataset

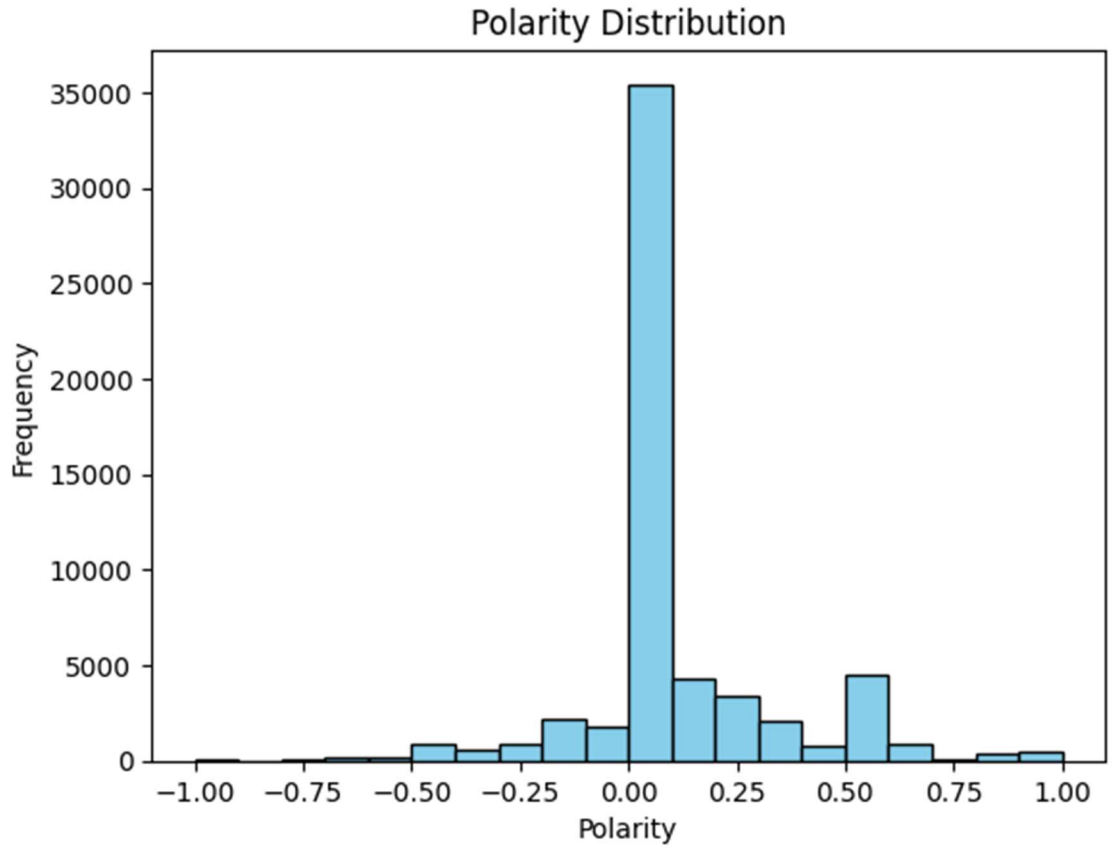


Polarity Distribution in Stock Tweets:

The polarity is mainly near zero in the figure 13, indicating the need to clean and preprocess data properly. The histogram, with bins set at 20, shows the frequency of various polarity levels, emphasising the dataset's overall sentiment tendency.

Figure 13

Polarity Distribution of Tweeter Dataset

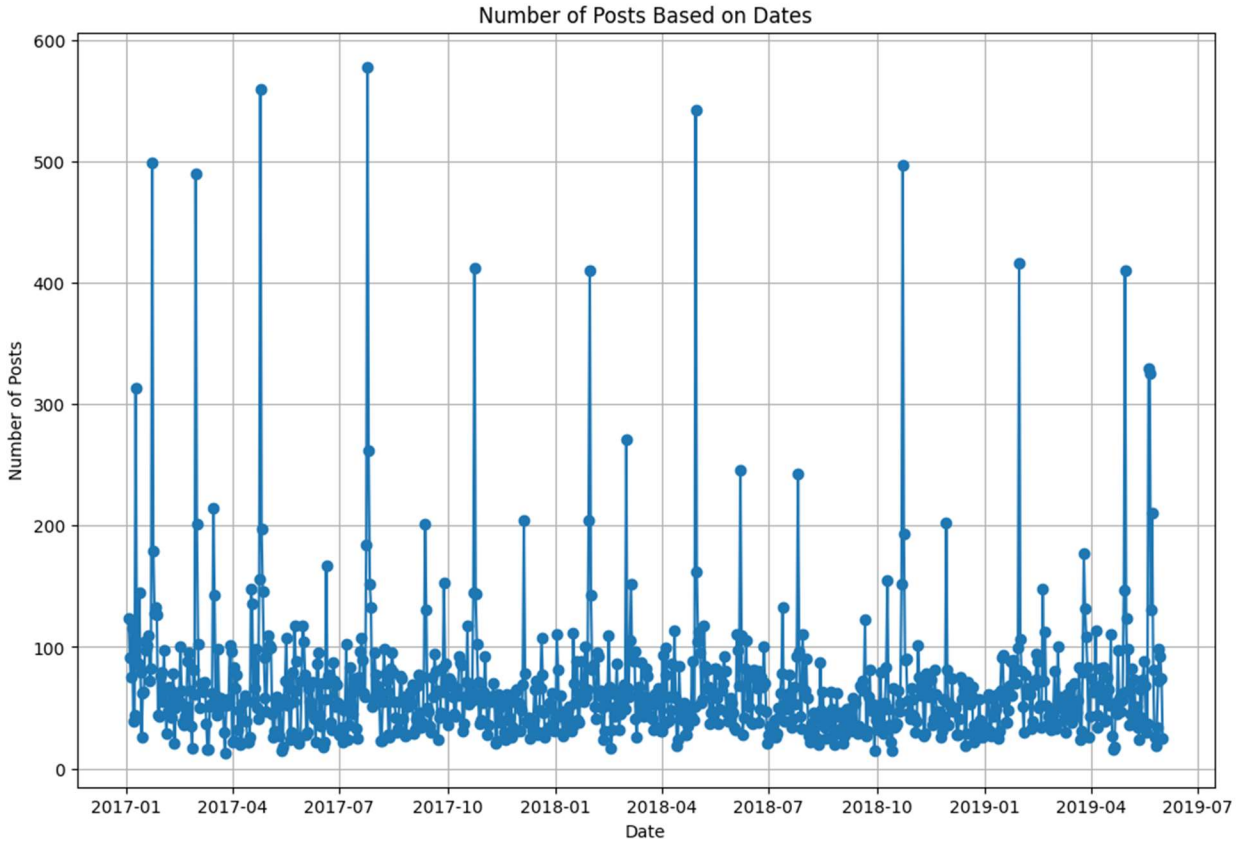


Posting Trends Over Time in Stock-Related Tweets

The resulting line plot from the figure 14 depicts the posting trends in stock-related tweets over time. The graph depicts a trend of abrupt increases in posts over a two-year period, which could aid in observing the stock in particular time periods for large changes in stocks. This data can help analysts better understand the temporal patterns of user involvement, potentially allowing them to correlate spikes or dips in post activity with specific market occurrences.

Figure 14

Number of Posts with Dates

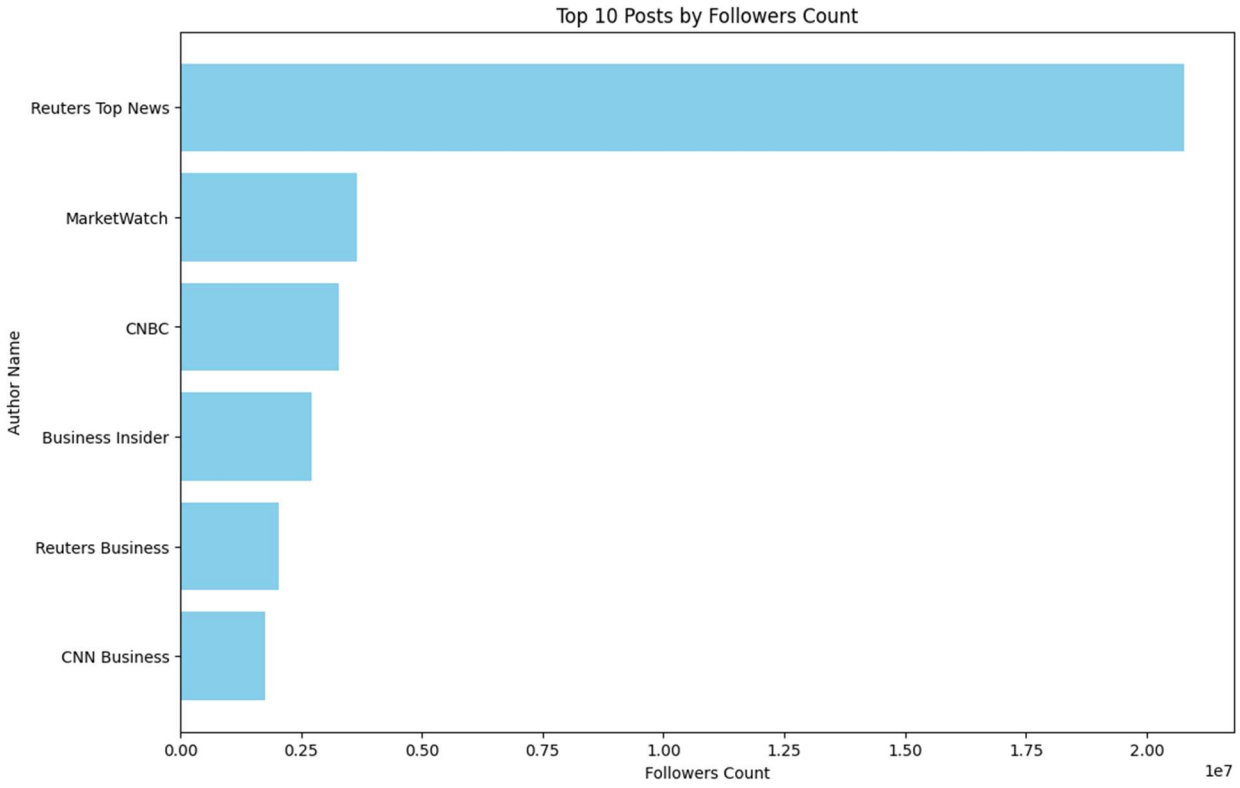


User Accounts with Highest Followers:

The horizontal bar chart in the figure 15 depicts the number of followers for the top ten users who have most of the followers. Based on the observations, it is concluded that all the top ten User accounts are news accounts like as CNBC, CNN Business, and Business Insider, among others, which guarantees that the information posted is legitimate and not an opinion tweet. This study will aid in comprehending the stock market conversations.

Figure 15

Top 10 User Accounts based on Followers Count



3.2.4 Data Pre-Processing

Check and Handle Null Values

Handling null data correctly, such as through imputation or elimination, is critical for developing accurate models and gaining valuable insights in the context of stock price prediction. The result in the figure 16 demonstrates that there are no null values in the dataset, indicating that better model predictions are possible.

Figure 16

Null Values Analysis in Tweeter Dataset

```
# Check for null values in the DataFrame
null_values = df.isnull().sum()
# Display the columns with null values (if any)
print("Columns with null values:")
print(null_values[null_values > 0])
```

```
Columns with null values:
Series([], dtype: int64)
```

Removing URLs and Unnecessary Patterns

We sanitise the raw social media data in many phases to provide a dataset suitable for feature extraction and the learning process. Hypertext Markup Language (HTML) is used to encode the data. As a result, all HTML encodings are removed. As shown in the figure 17, we remove HTML characters such as https+[text>]. & and currency marks such as $quot.

Figure 17

Removal of URLs from Tweeter Textual Data

```
[ ] # Function to remove URLs from the text
    def remove_urls_and_twitter(text):
        if pd.isna(text):
            return text

        # Remove URLs starting with http/https and Twitter-like patterns
        cleaned_text = re.sub(r'http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+|//twitter\.com/\S+|pic\.twitter\.com/\S+', '', text
        return cleaned_text.strip()

    # Apply the custom function to the 'text' column
    df['cleaned_text'] = df['text'].apply(remove_urls_and_twitter)
```

Removing Stock Tags and IDs:

The stock tags are also stripped from the text, as shown in the image, to obtain the fundamental form of the tweets. We erased information about the user ID and other sensitive information from the messages. We removed RT icons from messages. The Uniform Resource

Locators (URLs) were replaced by the keyword '_URL'. Similarly, we increased the length of the Hypertext Transfer Protocol (HTTP) and the World Wide Web (WWW).

Missing Value Analysis

We analysed any texts that were totally cleaned, leaving no data to process. The figure indicates that there are no missing values in the dataset, indicating that all of the data is in good shape.

Figure 18

Analysis of Missing values in Tweeter Data

```
[ ]  # Function for Missing Value/Empty Values in whole dataframe
     def missing_value_analysis(dataframe):
         # Check for missing values in the dataframe
         missing_values = dataframe.isnull().sum()

         # Print the missing values for each feature
         print("Missing Values Analysis:")
         print(missing_values[missing_values > 0])

     # Apply missing value analysis to your social media dataset
     missing_value_analysis(df)
```

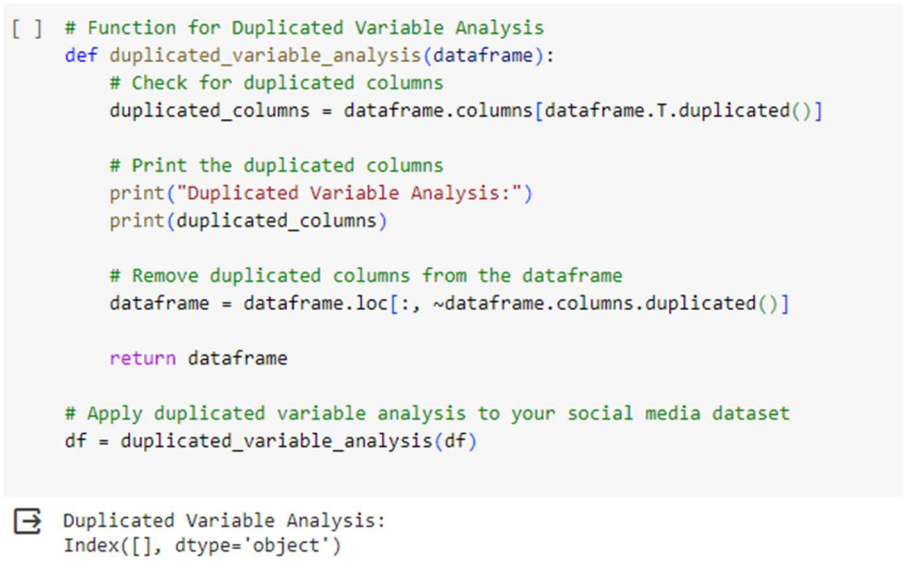
```
Missing Values Analysis:
Series([], dtype: int64)
```

Duplicated Variable Analysis

Sometimes a tweet is posted multiple times by mistake, and some retweets contain the original data. To delete this type of redundant material, we looked for duplicate sentences. After doing this analysis on the data, the results shown in the figure 19 indicated that all of the data in the dataset was unique.

Figure 19

Analysis of Duplicate Variables in Tweeter Data



```
[ ]  # Function for Duplicated Variable Analysis
     def duplicated_variable_analysis(dataframe):
         # Check for duplicated columns
         duplicated_columns = dataframe.columns[dataframe.T.duplicated()]

         # Print the duplicated columns
         print("Duplicated Variable Analysis:")
         print(duplicated_columns)

         # Remove duplicated columns from the dataframe
         dataframe = dataframe.loc[:, ~dataframe.columns.duplicated()]

         return dataframe

     # Apply duplicated variable analysis to your social media dataset
     df = duplicated_variable_analysis(df)
```

```
Duplicated Variable Analysis:
Index([], dtype='object')
```

Checking Spellings and Correcting words in Tweets

Correcting spelling errors, decreasing word elongation, fixing contractions, and cleaning mathematical symbols will aid in standardising the data. The goal is to improve the precision of sentiment analysis. This careful text cleaning method is critical for preparing the data for later analyses, making language understandable by Python tools like VADER, and obtaining the optimal sentiment score for each tweet. The end result is a refined and standardised text dataset that allows for more precise sentiment analysis, ultimately leading to more accurate stock price predictions. To standardise the text data, the following processes are taken.

- **Correct spelling mistakes:** Text Blob Library is used to correct words with incorrect spelling..
- **Correct word elongation:** Corrects word elongation by removing repeated characters (for example, 'victoryyyyy' becomes 'victory').

- **Correct contractions:** Corrects common word contractions (for example, "will not" to "won't").
- **Clean numeric symbols:** Removes extraneous apostrophes and punctuation splits from a sentence's last word. Additionally, it connects symbols to their respective words (for example, '$ AAPL' to '$AAPL'). Appendix A contains additional text data cleaning steps and words correction steps that were carried out in this process.

3.2.5 Feature Extraction

Core Meaning Extraction Using spaCy

The spaCy Python natural language processing (NLP) module was used to improve semantic understanding of stock-related text data. An English model spaCy ("en_core_web_sm") is downloaded and saved as model, which contains the analyzer for simplifying the sentences into simple form. The cleaned tweet data is then loaded into the model to extract the main meaning from the text. The spaCy model tokenizes, lemmatizes, and filters away non-essential features such as punctuation and spaces in each tweet.

The produced text represents the core semantic information of the original text, but in a more polished and meaningful manner. By focusing on the core semantic meaning inside the textual data, this semantic text processing with spaCy attempts to increase the accuracy of subsequent studies such as sentiment analysis and stock price prediction.

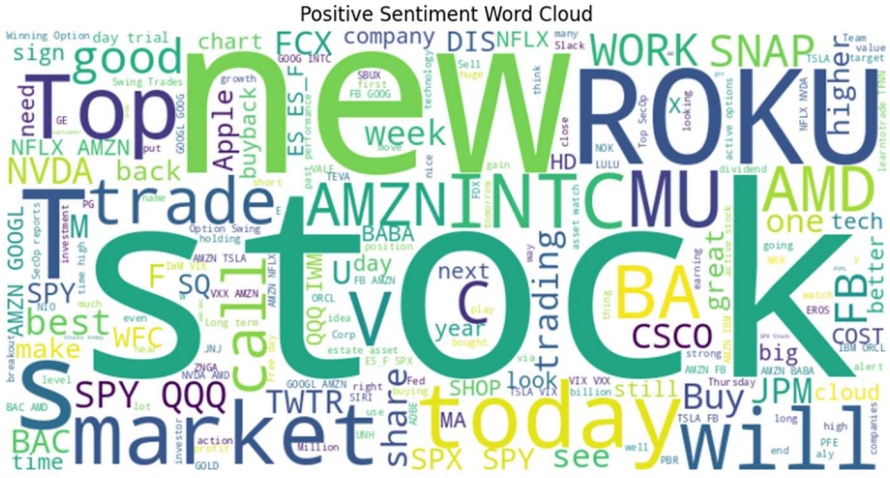
Extracting positive and negative words using Word Cloud

To visualise sentiment-driven word clouds for stock-related text data, use the Word Cloud library and the NLTK Sentiment Intensity Analyzer. It analyses the sentiment of each word in the text using VADER Lexicon. The positive and negative words are segregated based

on their sentiment ratings after downloading the 'vader_lexicon' for sentiment analysis. Using the WordCloud library, create positive and negative word clouds.

Figure 20

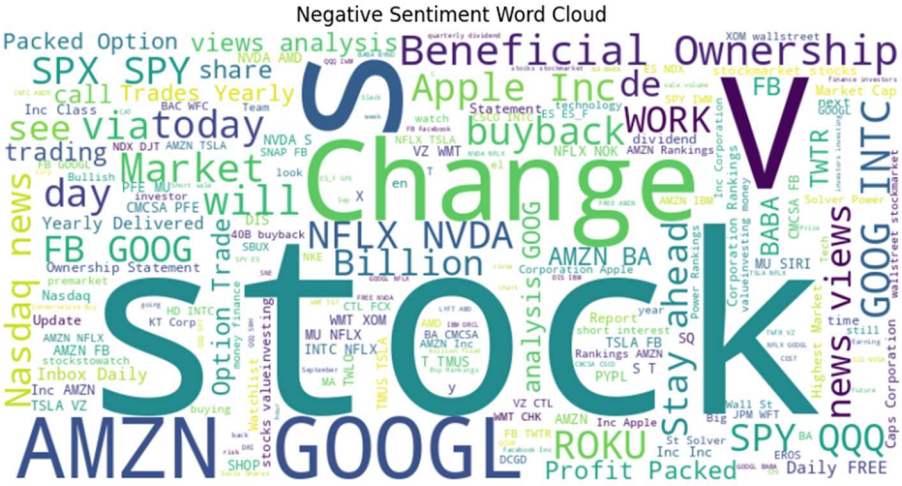
Analysis of Positive words in the Tweets



Certain indicators indicating the stock will rise are indicated in the figure 20, such as Buy, higher, best, look, buyback, fantastic, and huge, among others. In addition, indicators such as decline, packed, sell, and bullish are featured in the figure 21. These markers in a sentence aid in the distribution of each tweet's favourable and negative feelings.

Figure 21

Analysis of Positive words in the Tweets



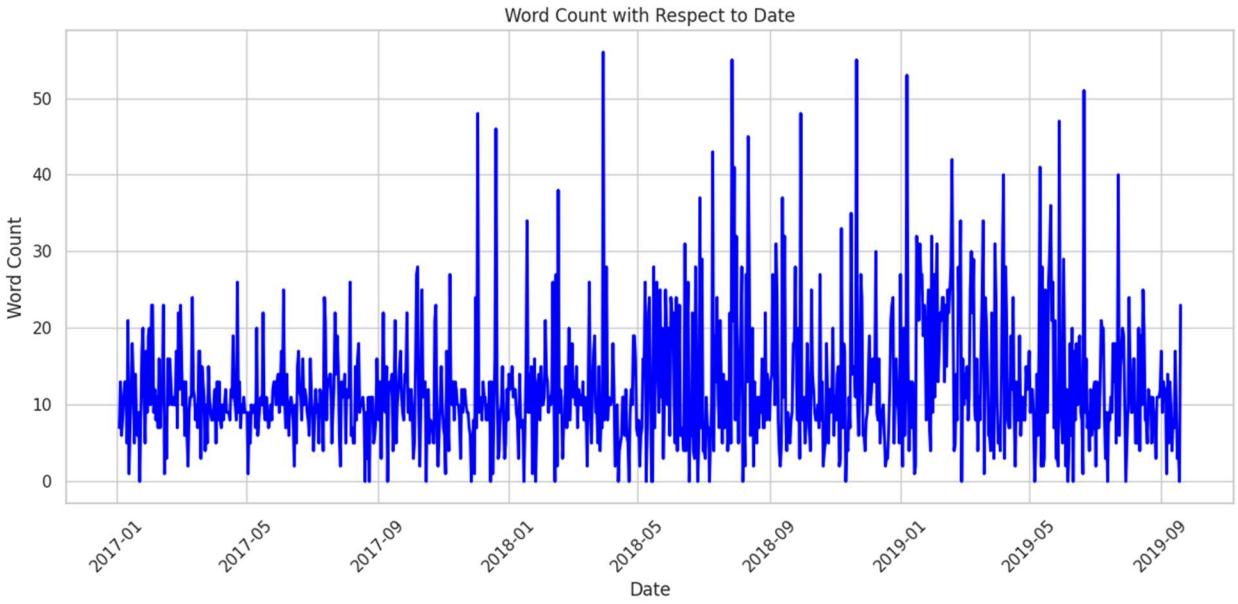
Integer Linguistic Features

Word Count Trends Over Time

Word count trends might provide information on the number and length of discussions. The figure 22 represents the evolution of the word count in stock-related writing throughout time. The x-axis displays dates, while the y-axis displays the associated word count. As shown in the graph, word counts increased dramatically between 2018-02 and 2018-04. This understanding can aid in the interpretation of sentiment dynamics.

Figure 22

Analysis of Words Count in Tweets Over Time

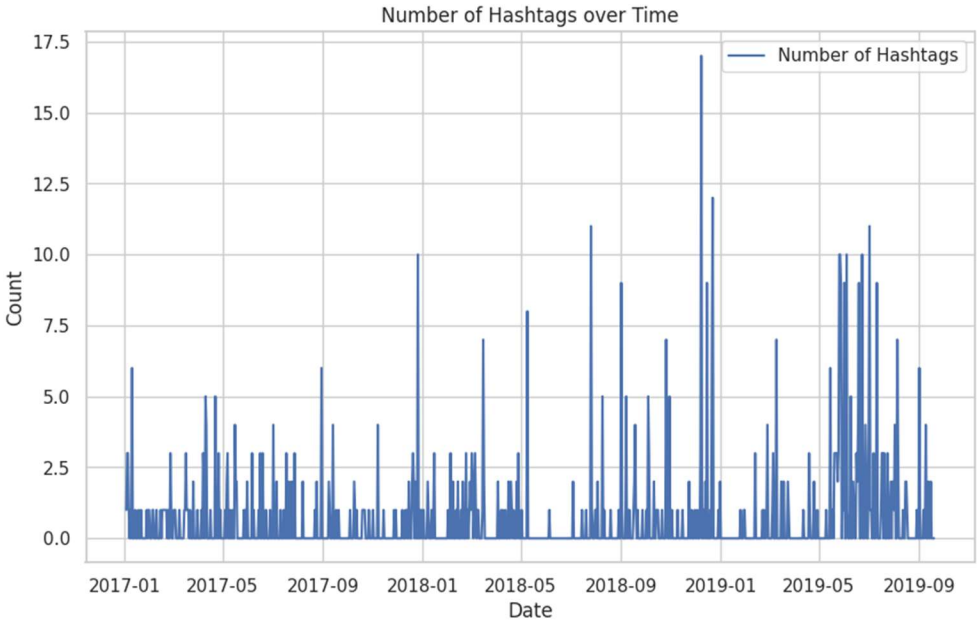


Hashtags Count

The quantity of hashtags in a post demonstrates adaptability and represents the stock's rapid up or down trend. The purpose of this analysis is to assess the variety of stock-related articles depending on the frequency with which hashtags are used. The figure 23 depicts a rapid increase in hashtag frequency in 2019-01. This element can aid in collecting awareness of that time frame in order to increase the emotion score's confidence.

Figure 23

Analysis of Hashtags Count Over Time



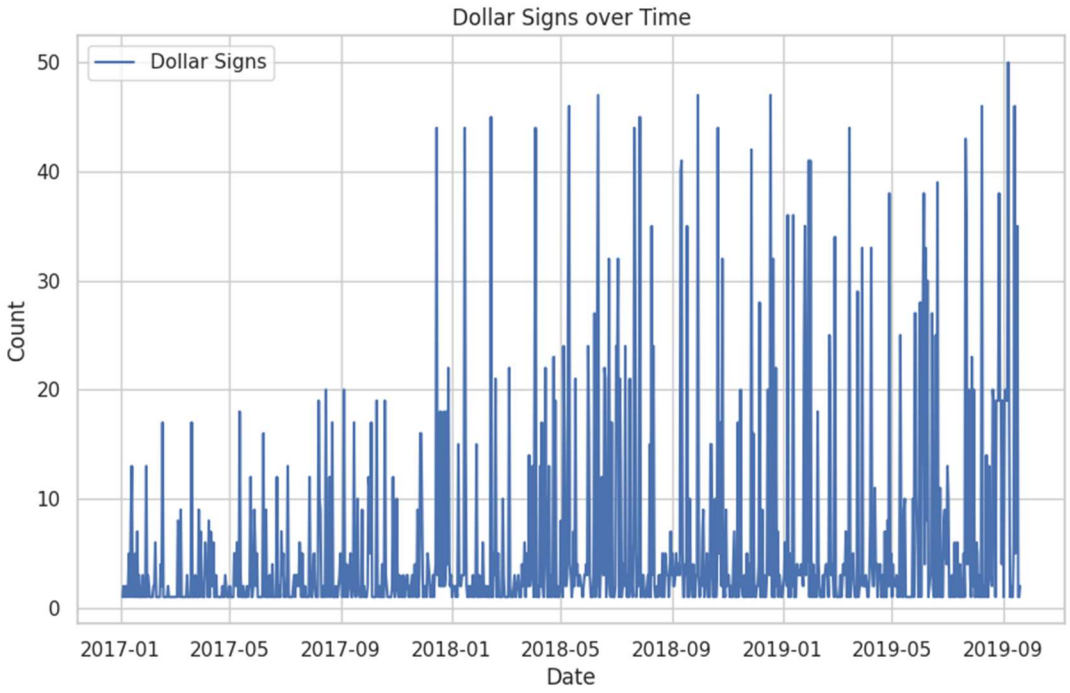
Dollar Signs

All dollar signs have a value that aids in recognising the enormous sum of money being invested or lost. Counting the dollar signs in the time series aids in determining the stock trend over various time intervals.

Figure 24 shows that the use of dollar signs more than doubled from 2018-01, indicating a new year's investments linked tweets that aid in analysing the trajectory of stock futures. The purpose of this analysis is to measure the financial emphasis inside stock-related posts, assuming that the presence of dollar signs indicates a focus on financial variables such as stock prices.

Figure 24

Analysis of Dollar Signs Count Over Time

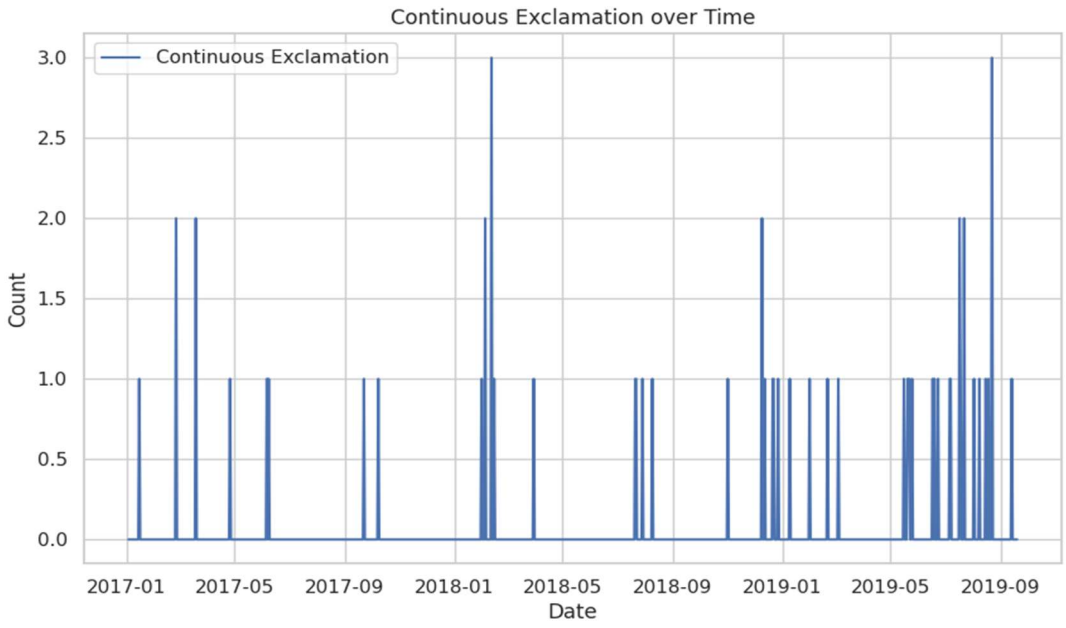


Continuous Exclamation mark

Exclamation marks are commonly used to convey excitement in sentences. An exclamation mark in a paragraph indicates that the stock is trending upward or downward. The figure plot depicts the use of regular expressions to identify and count sequences of exclamation marks in a data frame time period. The graph depicts an uncommon use of an exclamation mark in the data, implying that the tweets were tied to stock fluctuations on these dates. This feature aids in sentiment score analysis.

Figure 25

Analysis of Exclamation Count Over Time



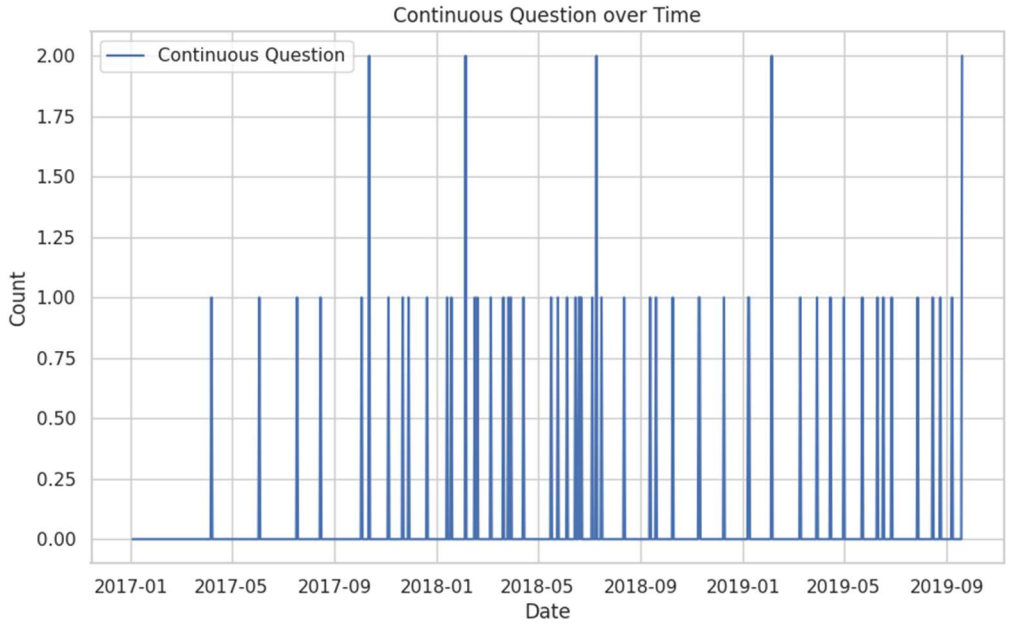
Question mark

The question mark indicates user participation. The number of question marks in tweets can reveal information about the level of inquiry or curiosity shown by individuals.

The figure 26 depicts the number of continuous question marks in a certain time period. Figure shows that consecutive use of question marks happened in the stock, indicating an unsettled sentiment.

Figure 26

Analysis of Question Marks Over Time



Log-Likelihood Ratio (LLR) scores

The Log-Likelihood Ratio (LLR) score is a statistical measure that compares the probability of seeing a specific collection of events to what would be predicted by chance. The LLR score is frequently used in natural language processing and collocation analysis to assess the strength of association between pairs of words known as bigrams.

We retrieved the LLR score feature by tokenizing the tweets into lowercase words, identifying and scoring bigrams with the Log-Likelihood Ratio (LLR) metric, and calculating the text's mean LLR score. It is used to handle probable mistakes and empty bigram lists.

3.2.6 Sentiment Score Extraction Using VADER

By analysing the data from the tweets with the VADER sentiment analyser, sentiment scores were generated. The SentimentIntensityAnalyzer from NLTK's VADER module examines the sentiment of each tweet, returning compound scores that describe the text's overall

sentiment polarity (positive, negative, or neutral). The distribution of sentiment scores from -1 to 1 ('-1' most negative, '0' neutral, and '1' most positive).

3.2.7 Sentiment Score Extraction Using TextBlob

TextBlob is a Python package for text processing. It provides a basic API for standard NLP operations such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. TextBlob is developed on top of NLTK (Natural Language Toolkit) and provides a user interface that is simple to use, making it accessible to those who do not have substantial expertise of NLP techniques.

Using the library, we derived TextBlob sentiment score, which will aid in comparison and combination with the VADER sentiment Analyzer. The sentiment scores obtained from both sentiment analysers are blended with weighted averages and saved as the final sentimental score.

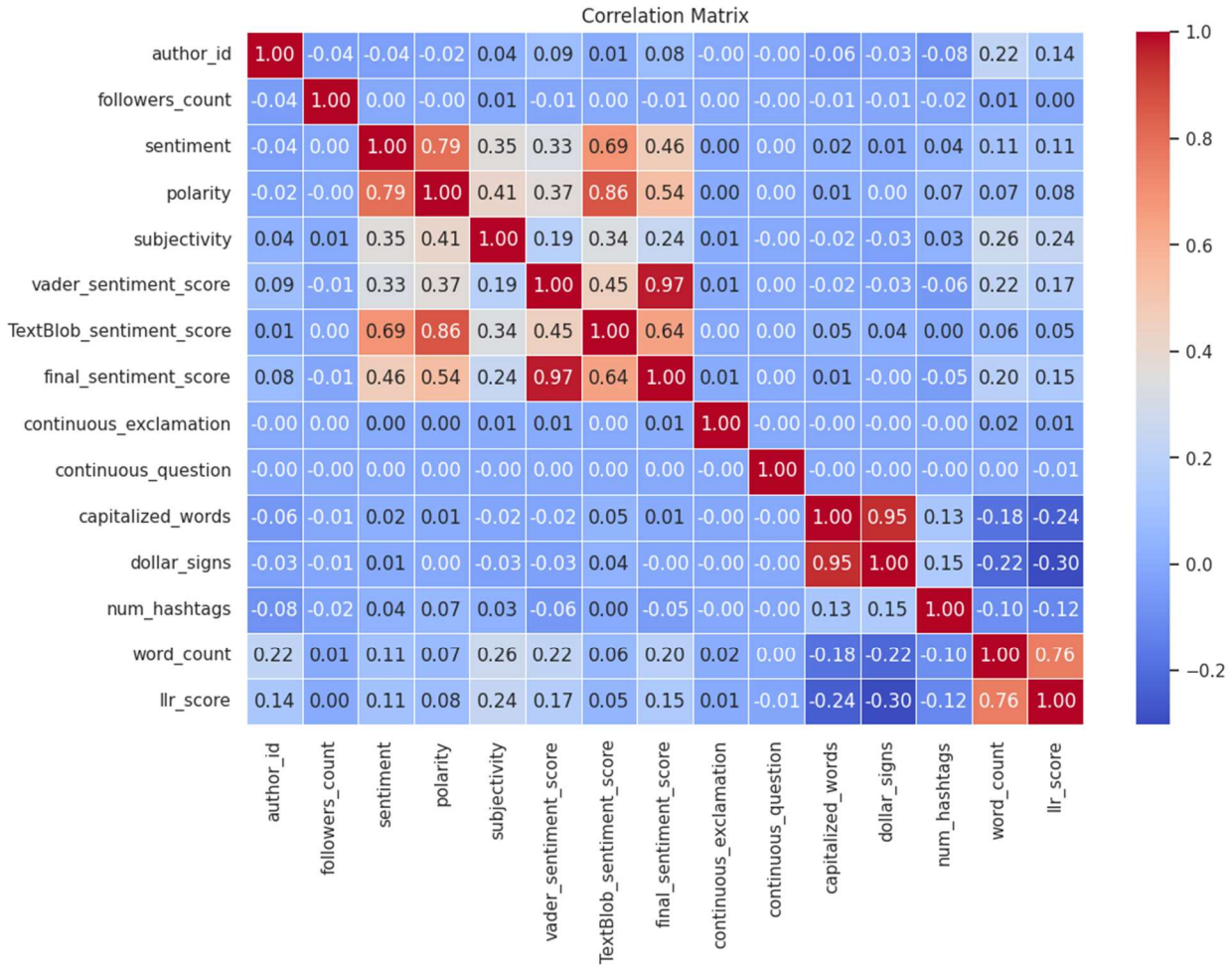
3.2.8 Correlation of Sentiment Features

The correlation demonstrates the relationship between several features in the dataset. As shown in the figure 27, there are a few high correlations between specific features.

The figure 27 depicts a strong association between some of the attributes due to their similar or close value ranges. For the time being, we will save the data and subsequently include it into the stock data before doing the feature engineering.

Figure 27

Correlation Matrix for Tweeter Dataset



3.3 Stock Data

3.3.1 Stock Data Collection

Scraping Financial Stock datasets requires less effort than scraping social media datasets. To obtain all of the essential stock data for the model, we utilised the Python package 'yfinance' to collect data from the Yahoo Finance library using date ranges and ticker values. Using the same start and finish dates as in the sentiment dataset, we extract historical stock data from web

scraping with the yfinance module. There are seven distinct web-extracted functionalities. They are as follows: Date, Open, High, Low, Close, Adjust Close, and Volume.

Figure 28

Top 5 Rows Analysis of Stock Dataset

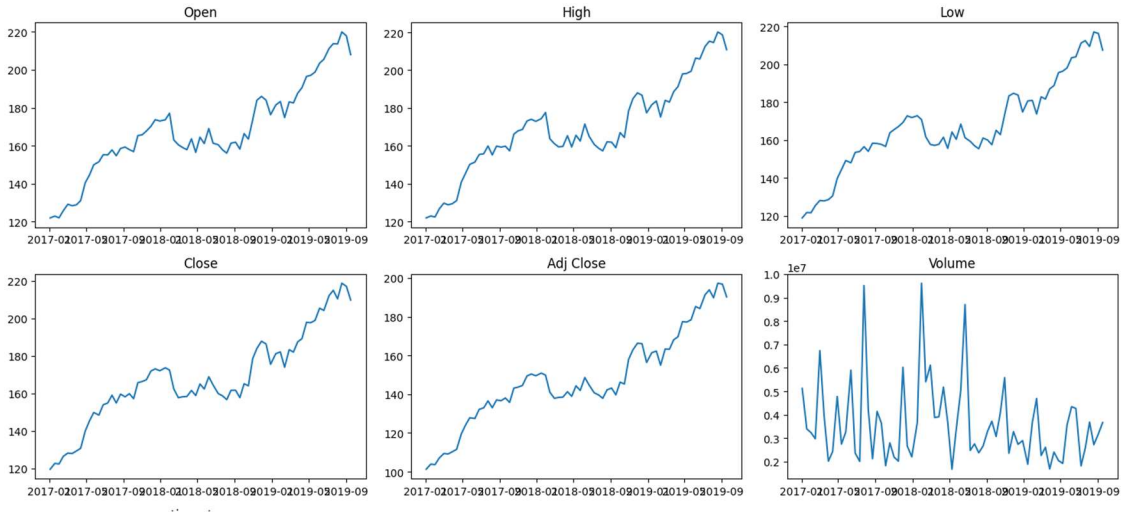
	Date	Open	High	Low	Close	Adj Close	Volume
0	2017-01-03	62.790001	62.840000	62.130001	62.580002	57.138721	20694100
1	2017-01-04	62.480000	62.750000	62.119999	62.299999	56.883072	21340000
2	2017-01-05	62.189999	62.660000	62.029999	62.299999	56.883072	24876000
3	2017-01-06	62.299999	63.150002	62.040001	62.840000	57.376122	19922900
4	2017-01-09	62.759998	63.080002	62.540001	62.639999	57.193508	20382700

Using yfinance from Python, a ticker utilised for this data extraction was MCD, IBM, PG, and NKE to perform the stock dataset scraping. The chosen start dates fall within the same range of sentiment data 'Date' values, which span from 2017-01-01 and 2019-11-01. Because of the stock market instability caused by the COVID-19 pandemic, this dataset expired on December 31, 2019. There are about 1000 records in the stock dataset.

The open, high, close, adj close, and volume features extracted by yfinance are shown in the figure 28, which aids in stock trend analysis. Their graph plot for stock price analysis is shown in the figure 29.

Figure 29

Stock Data Features Analysis



3.3.2 Stock Data Preprocessing

We used nan values to check for missing data and then dropped those values because they cannot be replaced with mean or median because they are regression values. To fix this issue, the nan values are first examined and then dropped if they are detected. The figure 30 illustrates that there are no nan values in the dataset.

Figure 30

Analysis of Null Values in Stock Dataset

```
[49] # Print rows with NaN values
     print("Rows with NaN values before dropping:")
     print(stock_data[stock_data.isnull().any(axis=1)])

     # Drop rows with NaN values
     stock_data.dropna(inplace=True)
```

```
Rows with NaN values before dropping:
Empty DataFrame
Columns: [Date, Open, High, Low, Close, Adj Close, Volume]
Index: []
```

3.3.3 Stock Data Feature Engineering

There are numerous aspects that can aid in determining the trend of stock data values. We will concentrate on close price prediction in this section. Because the stock's end date value is preserved in the close price. As a result, we will obtain the features based on the stock data's close price.

3.3.4 Adding Stock Indicators

Moving Averages and Bands

1. **Moving Averages (MA):**
    - **MA_7 (7-day Moving Average):** Short-term variations are smoothed, producing a trend indicator over a shorter time range. It can be used to determine recent price patterns and probable reversal points.
    - **MA_20 (20-day Moving Average):** Smoothers out medium-term oscillations, providing a more consistent trend indicator over a little longer time frame. It aids in identifying trends and probable turning points in the intermediate term.
2. **Exponential Moving Average (EMA):**
    - **EMA_20 (20-day Exponential Moving Average):** It gives more weight to recent data points, making it more sensitive to short-term price fluctuations. It reflects current market mood and is especially valuable for quickly capturing trends.
3. **Bollinger Bands:**
    - **Middle Band (MA_20):** Price movements are smoothed out by representing the core tendency. Serves as a starting point for calculating the price's divergence from the average.

- **Upper and Lower Bands:** Indicate probable overbought or oversold conditions. Widening bands indicate more volatility, whereas narrowing bands indicate decreased volatility. Price changes relative to the bands are frequently interpreted by traders as trend reversal or continuation signals.

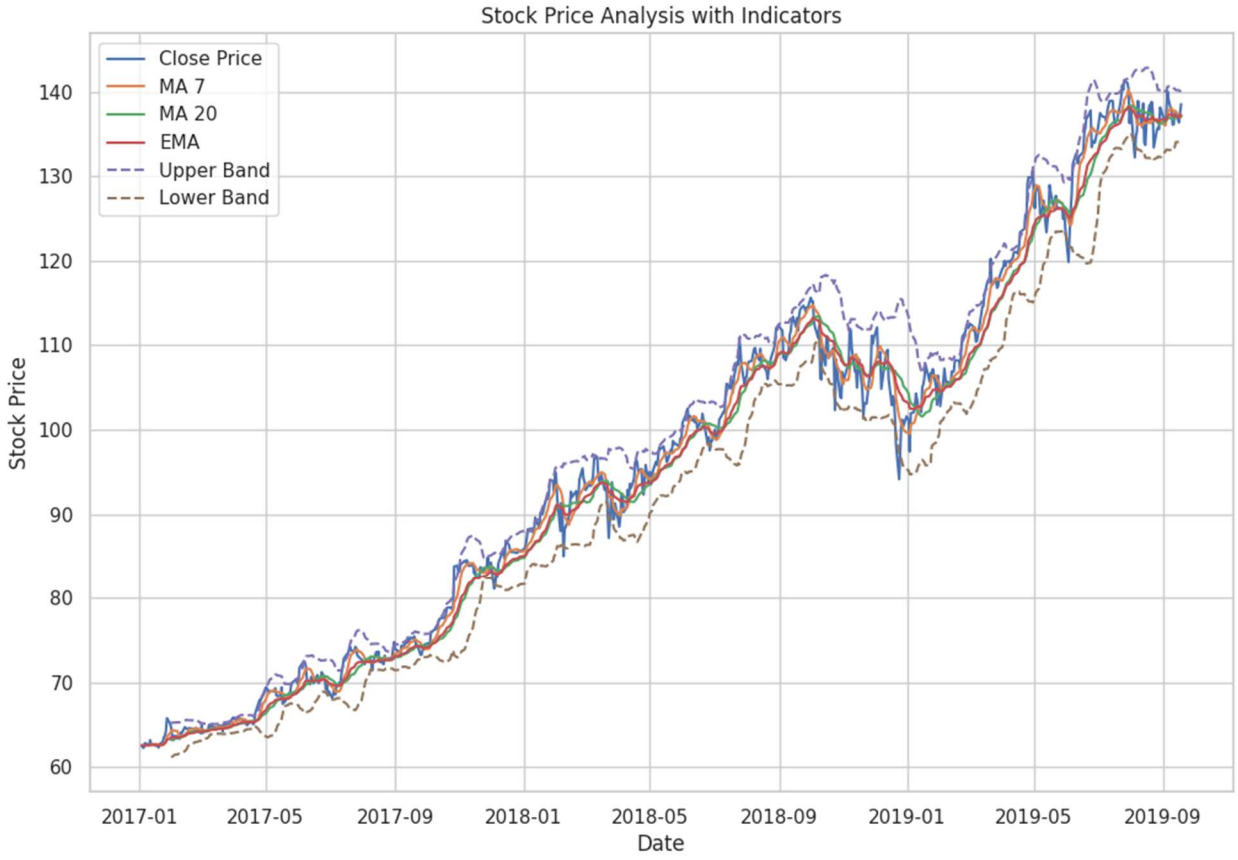
The dynamic interplay of moving averages and Bollinger Bands alongside closing prices is depicted in the figure 31, providing a comprehensive insight of the stock's price behaviour. The MA_7, indicated as a line that weave through short-term price changes, allows for a more focused observation of recent trends, assisting in the identification of prospective reversals. Similarly, the MA_20 captures medium-term trends with a smoother trajectory, offering consistency in trend analysis and detecting potential turning points over a little longer timeframe.

The EMA_20, represented as a line with increased reactivity to recent data, emphasises short-term price changes, indicating current market mood. This capability is extremely useful for quickly capturing shifting trends and adjusting to changing market conditions.

Furthermore, the Bollinger Bands give depth to the visualisation, with the Middle Band (MA_20) acting as a baseline, smoothing out price movements and serving as a reference point for evaluating deviations. The Upper and Lower Bands expand and contract dynamically, signalling probable overbought or oversold conditions. Widening bands imply increased volatility, whereas narrowing bands indicate decreased volatility. Traders frequently examine price fluctuations near these zones for clues about trend reversals or continuations. These visualisations, when combined, provide a full toolkit for analysing price dynamics and making informed stock trading decisions.

Figure 31

Stock Price Analysis with Indicators



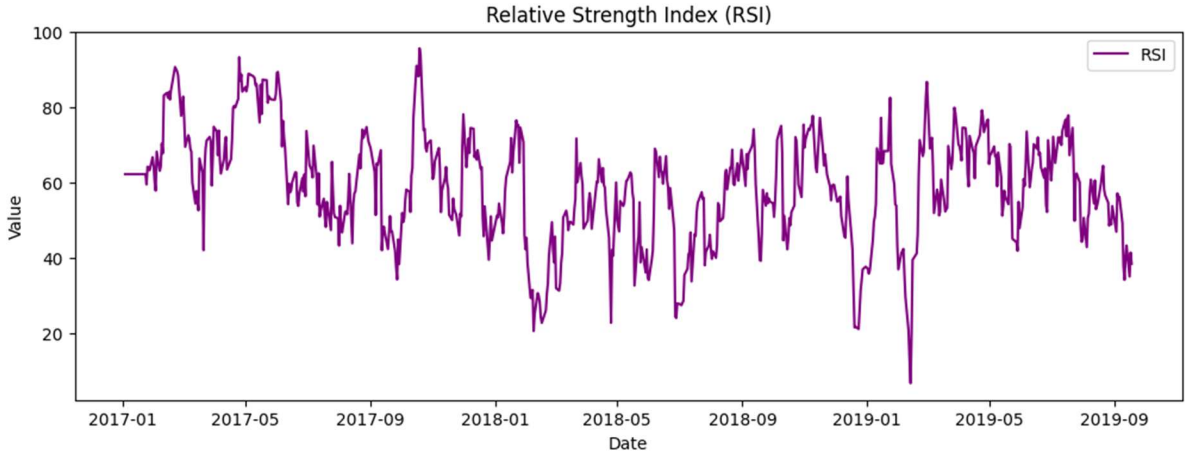
Adding Stock features

RSI (Relative Strength Index):

In figure 32, depicts the rate and amount of price fluctuations as a line chart. RSI readings greater than 70 indicate potential overbought situations, indicating a need for caution, while values less than 30 show potential oversold conditions, indicating a potential buying opportunity.

Figure 32

Relative Strength Index

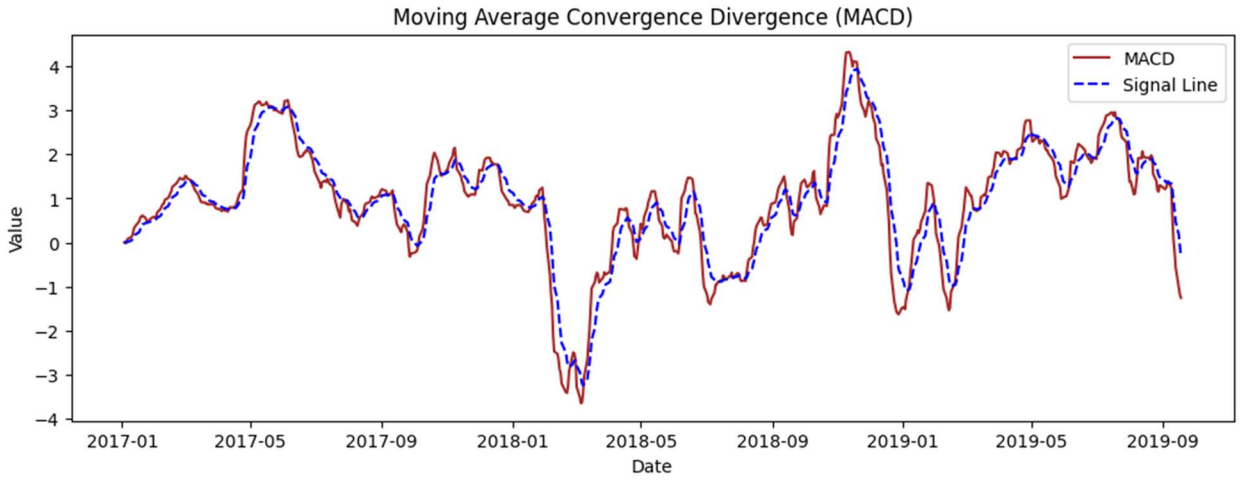


MACD (Moving Average Convergence Divergence):

The figure 33 depicts the relationship between two moving averages as a histogram or line chart. A MACD above the signal line shows a bullish trend, signalling possible upward momentum, whilst a MACD below the signal line indicates a negative trend, showing potential downward momentum.

Figure 33

Moving Average Convergence Divergence

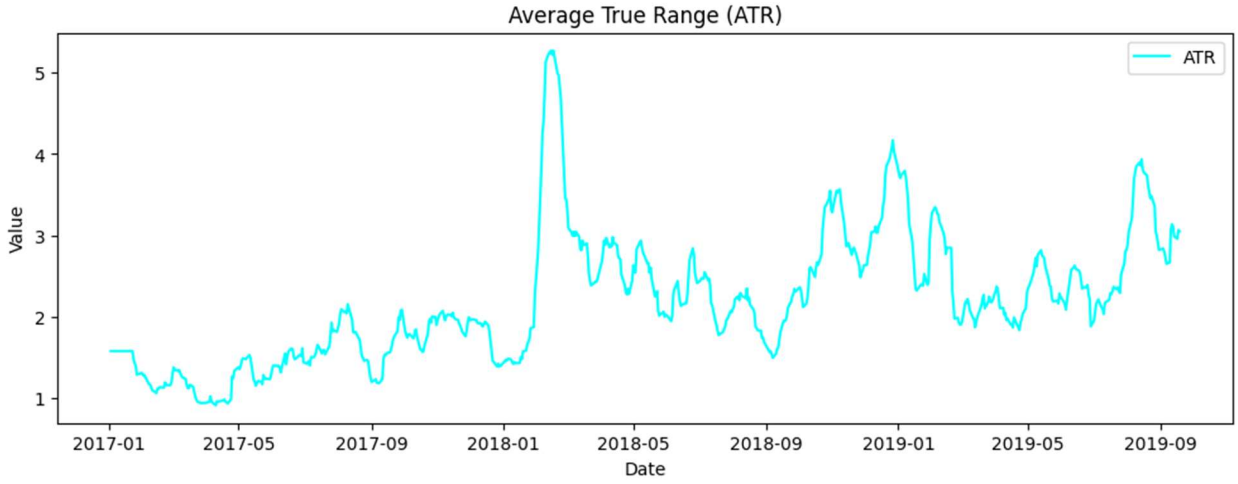


ATR (Average True Range):

Calculates the average range between high and low prices over a certain time period to determine market volatility. Figure 34 shows a high raise in price continued by small spikes.

Figure 34

Average True Range

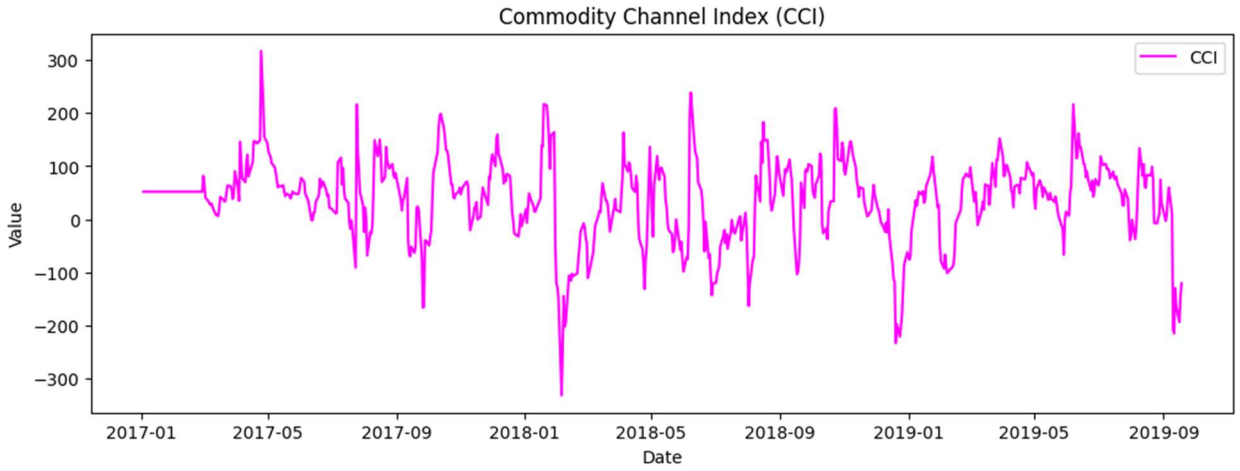


CCI (Commodity Channel Index):

Detects cyclical changes in asset values. Values greater than 100 indicate overbought conditions, while values less than -100 indicate oversold conditions. It is observed in the figure 35 that many times the stocks are over brought.

Figure 35

Commodity Channel Index

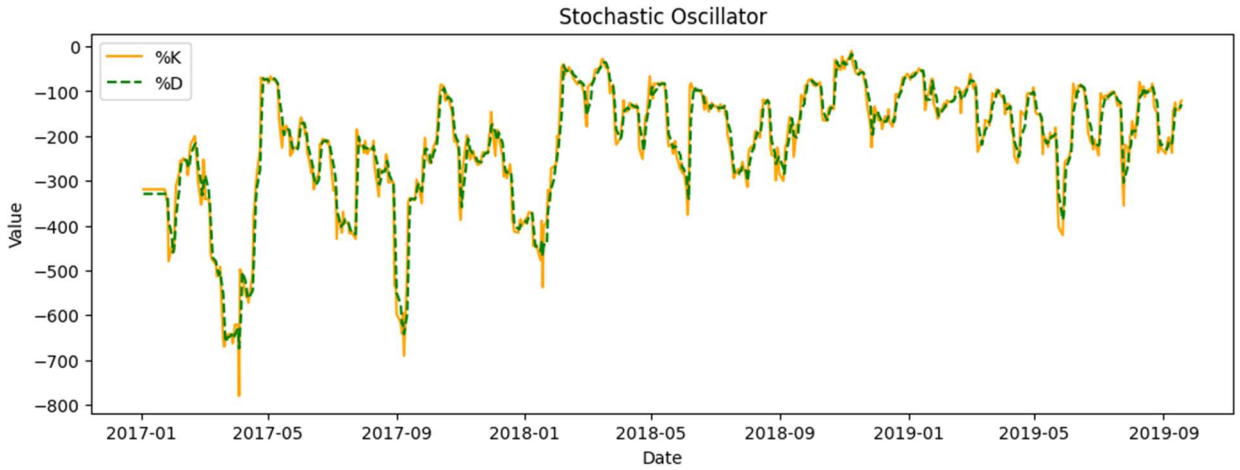


Stochastic Oscillator:

The location of a close relative to its pricing range over time. The current market rate is represented by %K, and its three-period moving average is represented by %D.

Figure 36

Stochastic Oscillator



3.3.5 Merging Sentiment and Stock Data

The 'Date' columns of sentiment data and stock data are both formatted similarly. Then, using Python's merge function, both datasets are merged, allowing for a fusion of the two datasets based on their temporal information. The 'Date' column is converted to datetime format, allowing for a more consistent sense of time across both datasets.

The integrated dataset provides a unified view, facilitating the identification of correlations or patterns between sentiment trends and stock price movements without directly referencing specific data or column names.

3.3.6 Correlation of Sentiment and Stock Dataset

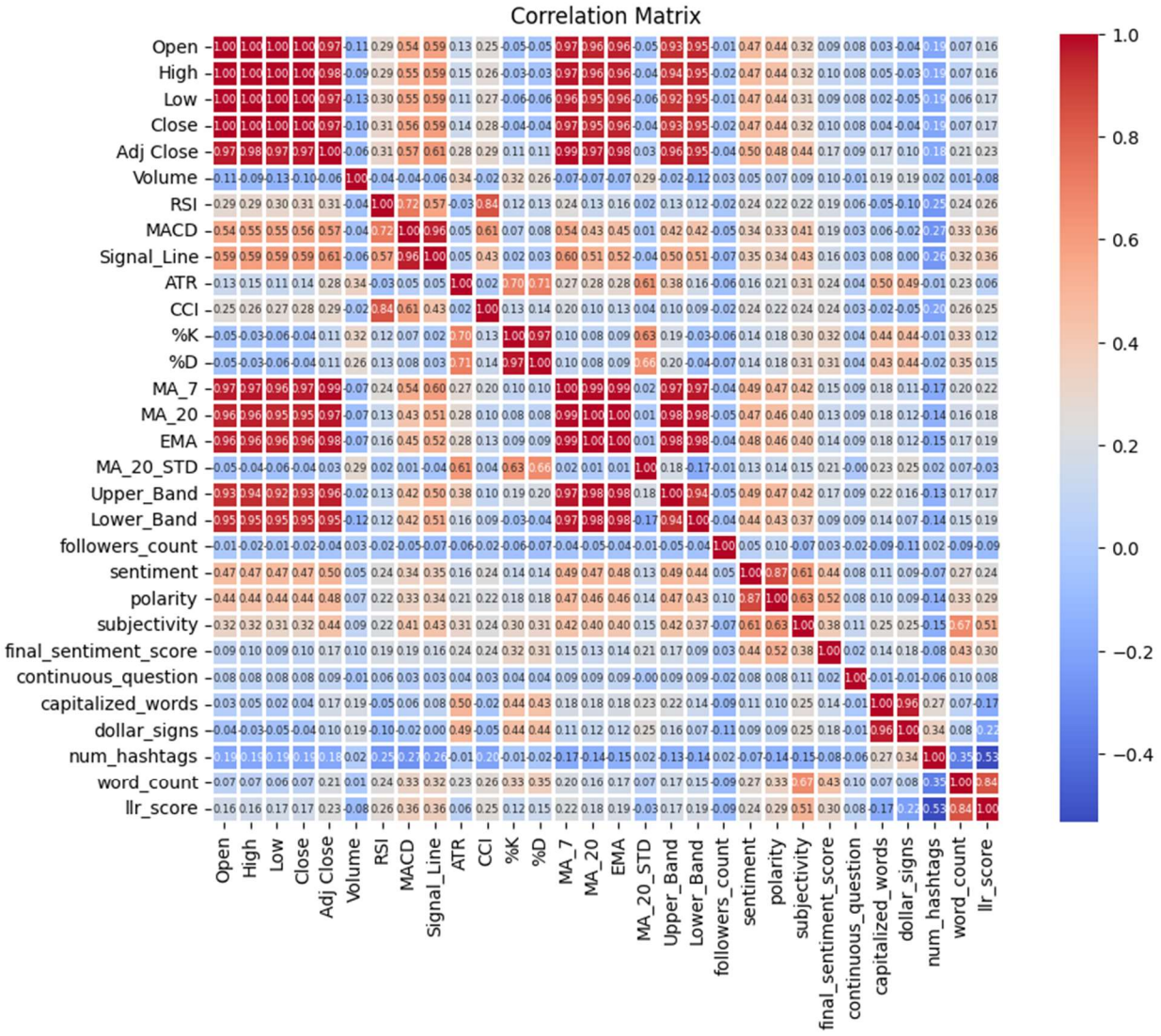
The correlation heatmap analysis correlations among several stock-related features are shown in the figure. There is a strong link between the stock data attributes open, high, low, close, adjusted close, upper band, and lower band values. This significant correlation is due to their intrinsic interdependence, as these indicators are organically linked and reflect various aspects of the stock's performance. Close price, Adjusted close price, and Bollinger Bands (upper and lower) are all intricately linked to daily price fluctuations, displaying a consistent pattern.

Moving averages (MA_7, MA_20) and the exponential moving average (EMA_20) also have a significant correlation. This link develops as a result of both indicators' similar goal of documenting trends across certain time spans. The MA_7, which represents the short-term trend, and the MA_20, which represents the intermediate-term trend, are closely related to the EMA_20, which prioritises recent data points for a more responsive trend depiction. Their interdependence stems from their shared goal of smoothing price swings in order to efficiently discover and interpret market trends.

Recognising the interdependence of these features makes feature selection for stock price prediction essential. The objective for this requirement is to methodically evaluate and delete highly correlated traits, keeping just those that give the most significant information for predicting the close price.

Figure 37

Correlation Matrix After Merging Tweeter and Stock Dataset



3.3.7 Feature Selection

The feature importance study suggests that sentiment, which has values of -1, 0, and 1, is the most influential characteristic in forecasting the close stock price. This is owing to its capacity to capture market sentiment, which can range from extremely negative to neutral to extremely positive. Understanding market sentiment is essential for forecasting stock prices since it reflects investors' general perception and sentiment towards a specific stock.

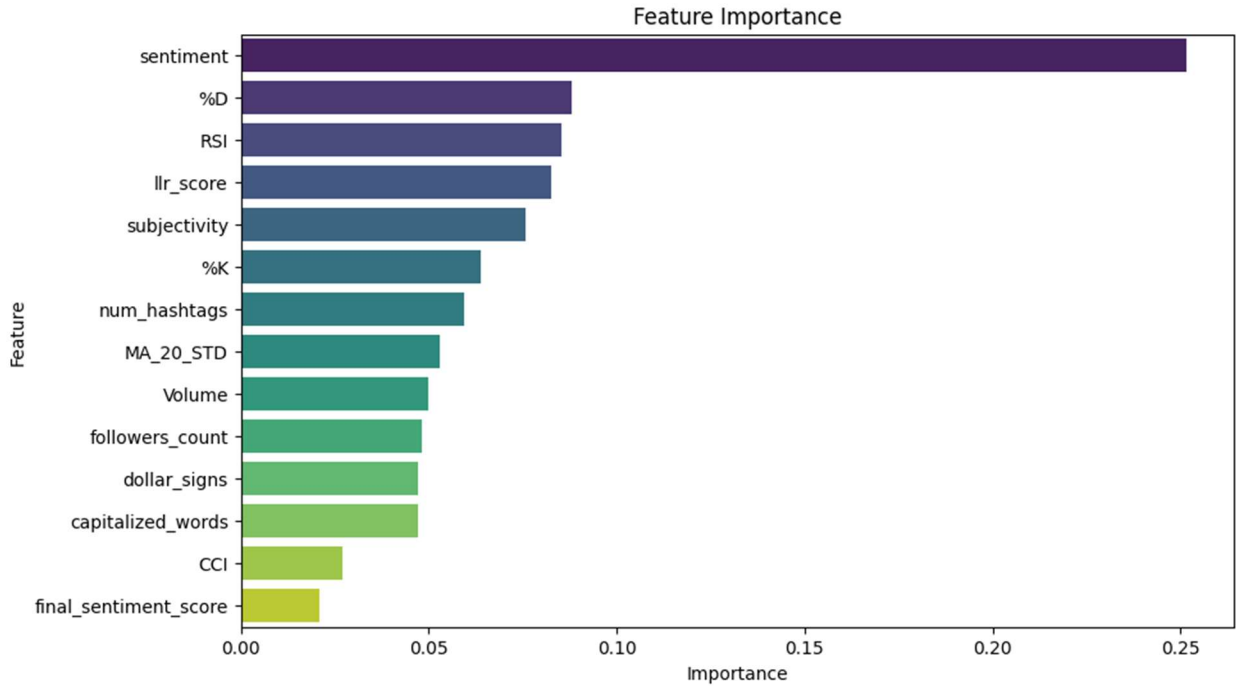
Furthermore, the indicators %D and %K, as well as the Relative Strength Index (RSI), play an important role in predicting the close price. The Stochastic Oscillator components %D and %K, as well as the RSI, provide vital insights into the momentum and strength of market moves. Investors frequently rely on these indicators to identify probable overbought or oversold levels, which aid in identifying critical market turning events.

LLR score, follower count, number of hashtags, and trading volume are all important factors. The LLR score, which is calculated using textual data, represents linguistic patterns and emotions in financial discussions. Social media measures like the number of followers and hashtags provide extra insight into the stock's social influence and trends. Meanwhile, trading volume (the total number of shares traded) is a key element determining stock prices.

Furthermore, the final_sentiment_score is highlighted as an important feature that improves the model. This metric, which is most likely obtained from aggregating sentiment scores or other sentiment-related variables, adds to the model's predictive powers by adding an additional layer of sentiment analysis for improved accuracy in predicting stock values.

Figure 38

Feature Importance of Final Dataset

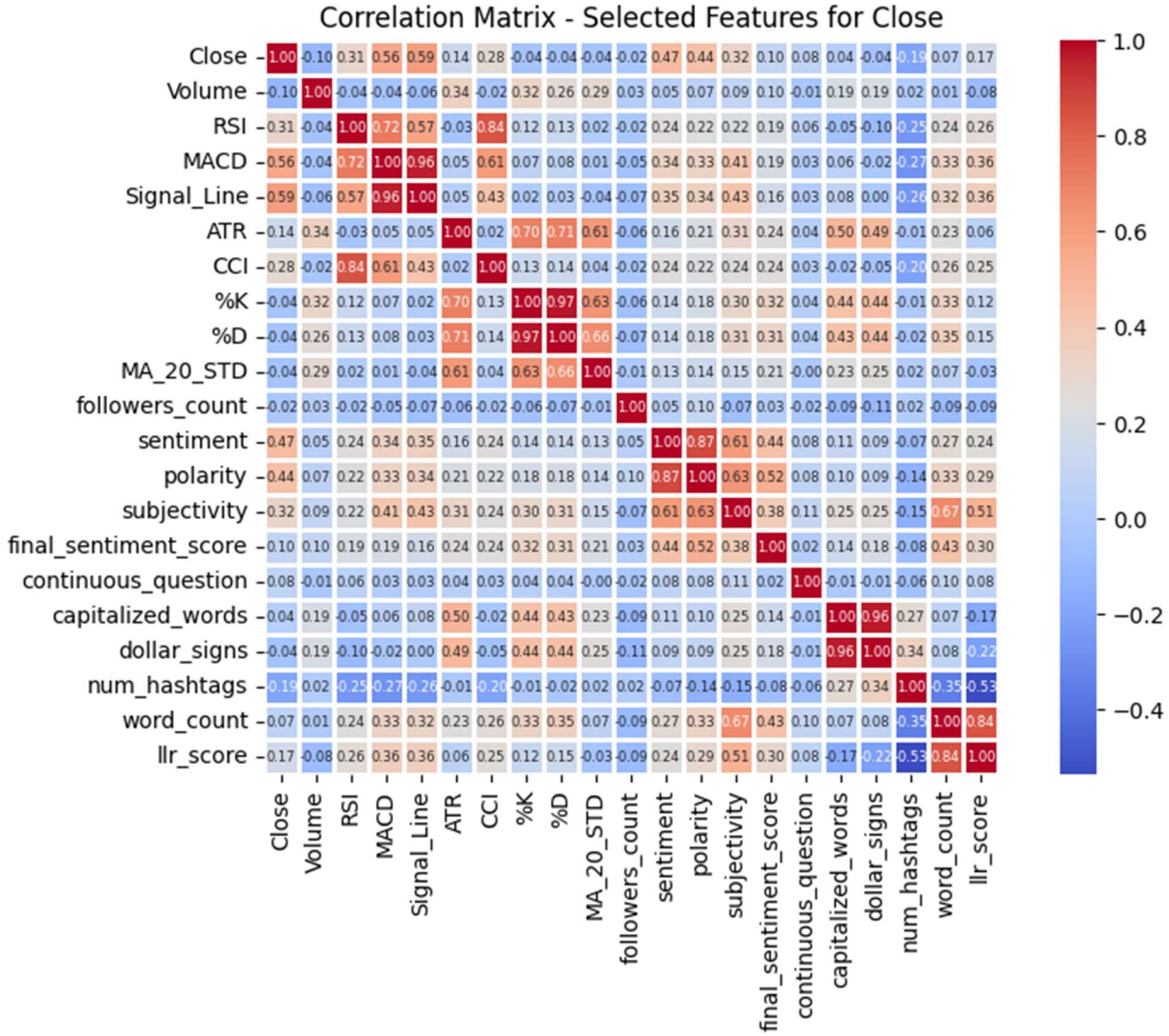


Given the substantial correlation between features in figure 7, feature selection is critical for stock price prediction. This entails systematically evaluating and removing overly associated elements to ensure that the model focuses on the most relevant and different components of the data.

This strategy method improves the predictive model's accuracy and efficiency for close stock price. As shown in Figure After feature selection, the correlation heatmap displays superior correlation values with all features, which are ideal for model training.

Figure 39

Correlation Matrix After Feature Selection and aligned with Close Price



3.4 Building Model and Training

In this research, we used a Long Short-Term Memory (LSTM) neural network model to forecast stock prices. The LSTM design, a subtype of recurrent neural network (RNN), was chosen for its ability to capture sequential dependencies in time series data. This model seeks to

predict the future adjusted closing prices of a given stock by using previous stock prices and sentiment scores as features.

The incorporation of sentiment analysis, as represented by the 'final_sentiment_score' and other elements that improve the model's ability to incorporate market sentiment into its forecasts. The LSTM model excels at managing time series data, making it ideal for forecasting stock prices, which are fundamentally time dependent.

Normalization of data

The normalisation process uses the Min-Max Scaling approach. The MinMaxScaler from the sklearn Preprocessing module is used to do this. The feature_range option is set to (0, 1), which scales the data to a certain range and ensures that all features fall between 0 and 1. Min-Max Scaling is a well-known normalisation approach that transforms data by deleting the minimum value and dividing it by the range, guaranteeing that the scaled data falls inside a defined interval.

Time Series Data Creation

To capture the temporal dependencies that arise from time series data, a function called create_time_series_data is introduced. This function generates input-output pairs for the LSTM model based on the number of time steps supplied. The variable time_steps is set to 10 in this case.

The function's primary goal is to generate input sequences (denoted as X) and their corresponding output values (denoted as y). These sequences are generated based on the amount of time steps supplied, which is set to 10 in this case.

In most cases, the input sequences (X) reflect historical data or observations from the past, whereas the corresponding output values (y) are forecasts or target values connected with those input sequences. The time steps parameter determines how far back in time the model should look when creating these sequences, and it aids in shaping the structure of the time series data.

Figure 40

Time Series Data Creation Code

```
# Function to create time series data
def create_time_series_data(data, time_steps=1):
    X, y = [], []
    for i in range(len(data) - time_steps):
        X.append(data[i:(i + time_steps), :])
        y.append(data[i + time_steps, 0])
    return np.array(X), np.array(y)

# Set the time steps
time_steps = 10

# Create time series data
X, y = create_time_series_data(df_scaled, time_steps)
```

Train-Test Split

Using scikit-learn's train_test_split method, the dataset is divided into training and testing sets. By setting shuffle to False, the temporal order is preserved, ensuring that the model is exposed to the historical sequence of data during training.

A train_test_split function normally accepts time series data (X, y) and outputs four sets: X_train, X_test, y_train, and y_test. The training sets (X_train and y_train) had 80% of the data, whereas the testing sets (X_test and y_test) held 20%. This division enabled efficient model training and evaluation in a supervised learning situation.

Hyper Parameters

The number of LSTM units is adjusted at 50 in order to strike a compromise between capturing detailed patterns and computational efficiency. Set the return sequences option to True for the first LSTM layer to improve the model's ability to capture sequential patterns. The input shape is defined depending on the dimensions of the training data, taking into consideration time steps and features. With its variable learning rate, the Adam optimizer is well suited to non-stationary time series data.

For regression problems, the loss function is Mean Squared Error. The model goes through 50 epochs to balance the hazards of convergence and overfitting, with a batch size of 32 for efficient parallelization. The dataset (X_test, y_test) incorporates validation data, which is critical for monitoring performance on unseen data.

Table 2

Hyper Parameters for LSTM Model

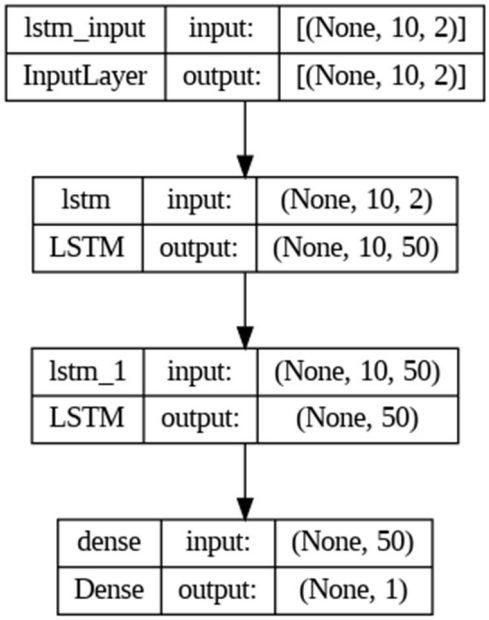
Hyperparameters	**Value**
LSTM Units	50
Return Sequences	True
Input Shape	(X_train.shape[1], X_train.shape[2])
Optimizer	Adam
Loss Function	Mean Squared Error
Epochs	50
Batch Size	32
Validation Data	(X_test, y_test)

LSTM Model Architecture

The LSTM model has a dual-layer design with an LSTM layer and a Dense layer, totaling 10,451 parameters. The LSTM layer, with 50 units and ReLU activation, processes input moulded by the time steps and features of the training dataset. It has 10,400 trainable parameters that are obtained from four times the units, input dimension, and biases. The following Dense layer, which serves as the regression output, employs a linear activation and adds two more trainable parameters. This model is parameterized for effective learning, particularly in capturing subtle sequential patterns, with a focus on adaptability. Notably, it lacks non-trainable factors, which increases its versatility during training.

Figure 41

LSTM Model Design with Hyperparameters



The LSTM model is built with TensorFlow's Sequential API. It is made up of LSTM layers that capture temporal dependencies and a Dense layer that produces the final output. The model's

architecture is designed for time series forecasting, with the goal of discovering patterns and relationships in the input data.

3.5 Summary of chapter

In this chapter, a comprehensive approach for analysing sentiment and stock data is proposed. The sentiment dataset, generated from a large Twitter dataset, is thoroughly analysed and several features are extracted. Concurrently, the stock data is analysed, and numerous features are extracted. Both datasets are subjected to data preparation and correlation analyses. Following that, the datasets are combined based on the date. Then, feature engineering is used to identify the best features.

Following the preprocessing procedures, an LSTM model with specified time steps and hyperparameters is built. To facilitate analysis and prediction, the prepared data is loaded into the LSTM model.

CHAPTER 4 : FINDINGS AND ANALYSIS

This chapter examines the model's performance metrics, which provide useful insights into the relationship between sentiment variations and stock closing prices. The first section displays the LSTM model evaluation metrics for four different equities. The model is then cross validated in section 2 to ensure that it covers all four stocks. Section 3 investigates the relationship between sentiment score and close price, providing insight into sentiment trend and stock price. In the final section, the proposed model is compared to models from relevant publications.

4.1 Evaluation Results

IBM Stock Evaluation Metrics

The forecasting model for IBM stock prices performs admirably. With a Mean Squared Error (MSE) of 3.678, the IBM stock prediction model performs well. The model's precision in predicting stock prices is demonstrated by the Root Mean Squared Error (RMSE) of 1.917. The R-squared (R2) value of 0.8080 indicates that the model effectively explains 80.80% of the variation in IBM stock prices.

The graph depicts the prediction of near values, with the forecasted values following the trend of the close price. As seen in the image, the model can detect the majority of the trends, indicating the model's strong performance. Figure shows that training and validation are reducing dramatically, indicating that the model is learning well.

Table 3

IBM Stock Evaluation Metrics

Evaluation Metric	Value
Mean Squared Error (MSE)	3.678
Root Mean Squared Error (RMSE)	1.917
R-squared (R^2)	0.8080

Figure 42

IBM Stock Actual vs Predicted Price Graph

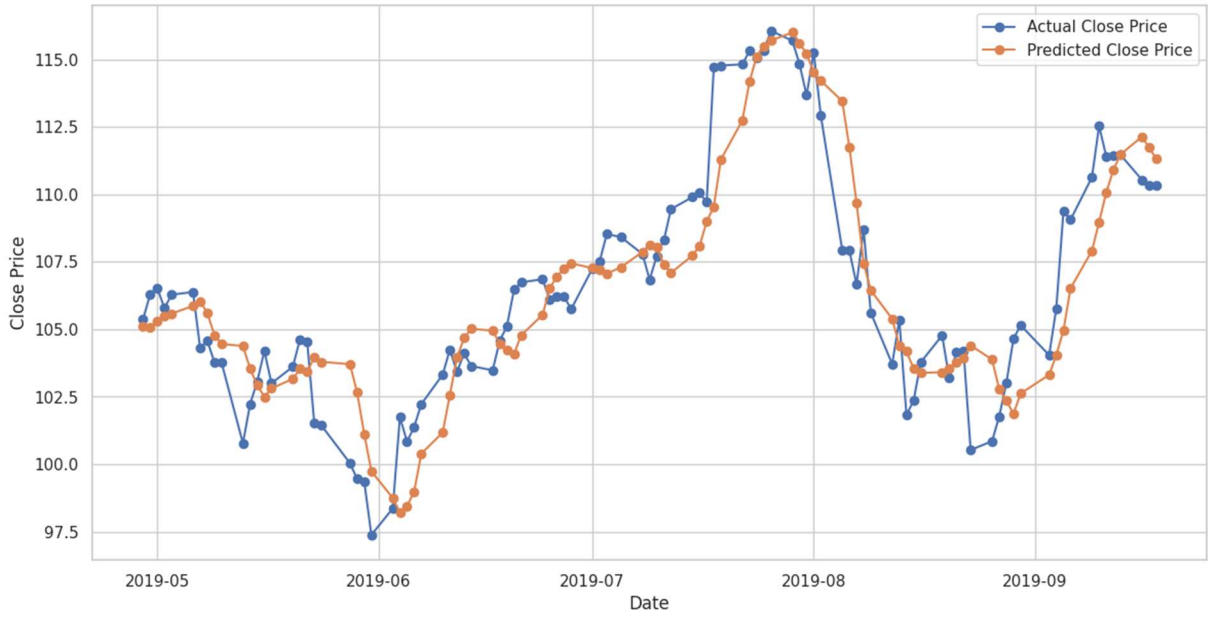
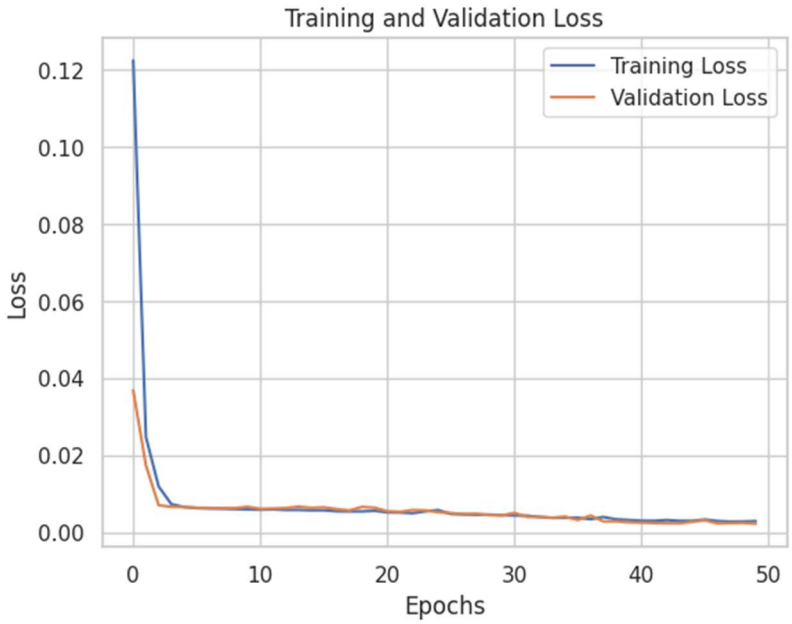


Figure 43

IBM Stock Training and Validation Loss



MCD (McDonald) Stock Evaluation Metrics

The model for McDonald's stock has a Mean Squared Error (MSE) of 10.21, which measures the average squared difference between anticipated and actual stock values. A lower MSE indicates that the model's predictions closely match the observed values. The Root Mean Squared Error (RMSE) of 3.19 is an important predictor of precision. The R-squared (R2) score of 0.821 indicates that the model explains 82.10% of the variance in forecasting accuracy.

As shown in the figure 44, this model produces the best results with slight underfitting. According to the figure, the model's learning rates are very high.

Table 4

McDonald Stock Evaluation Metrics

Evaluation Metric	Value
Mean Squared Error (MSE)	10.21
Root Mean Squared Error (RMSE)	3.19
R-squared (R^2)	0.821

Figure 44

McDonald Stock Actual vs Predicted Price Graph

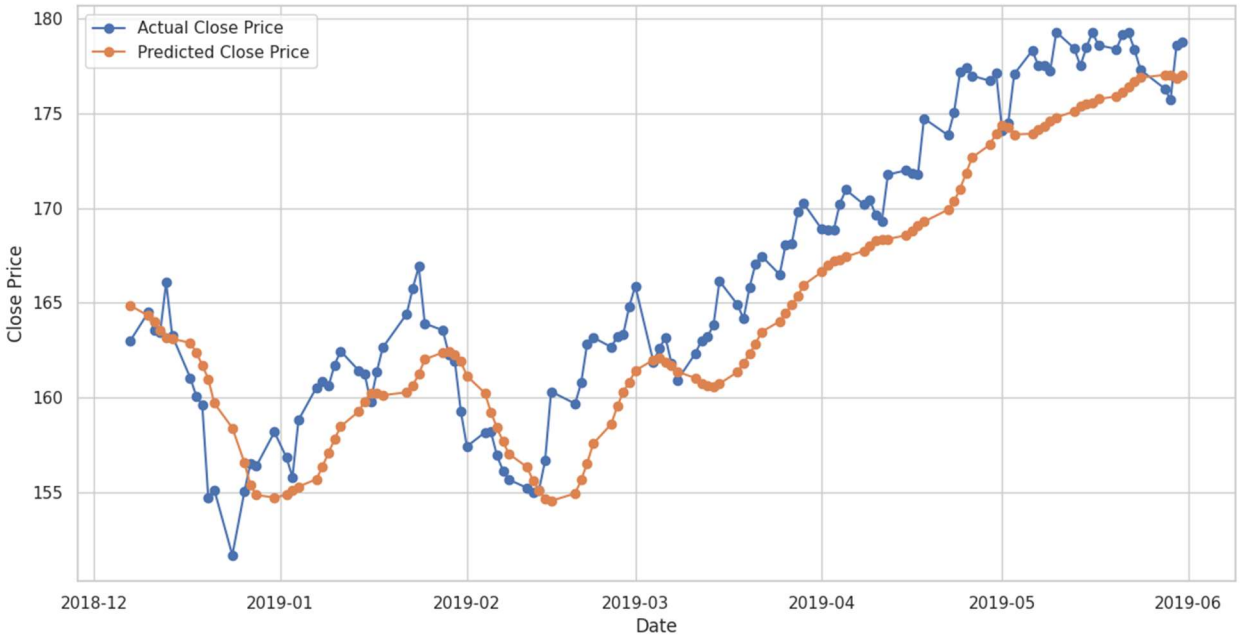
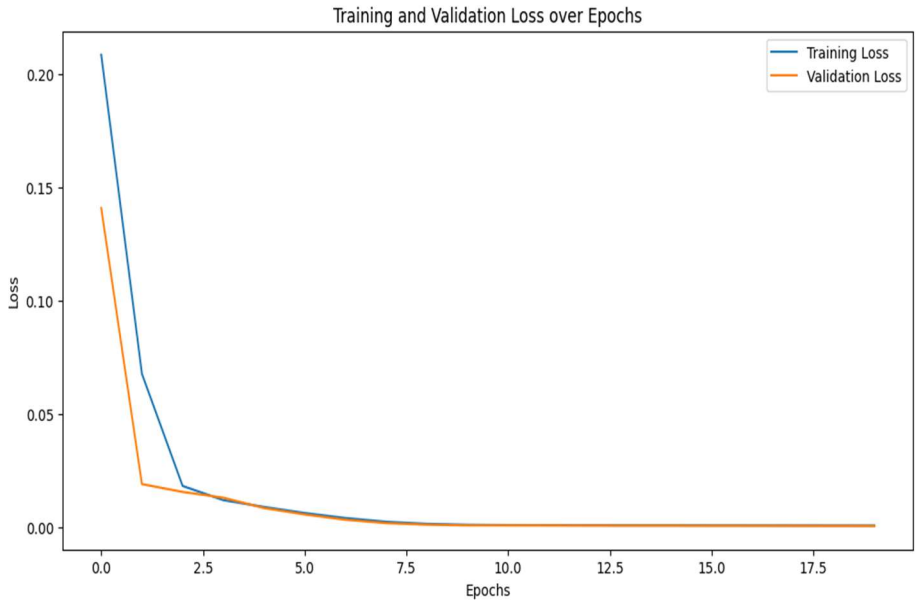


Figure 45

McDonald Training and Validation Loss



PG (Procter & Gamble Co) Stock Evaluation Metrics

The MSE of 8.47 for PG emphasises the average squared difference between projected and observed stock prices, emphasising closer alignment of predictions with observed values. The RMSE of 2.91 highlights the model's precision by providing a measurable measure of prediction error. The model explains 78% of the variability in Procter & Gamble Co's stock prices, contributing to its forecasting accuracy, with an R-squared (R2) score of 0.78.

The learning rates in the PG model fluctuate in the conclusion. However, the model predictions are good, with some underfitting at the conclusion of the projection due to variations in losses.

Table 5

PG Evaluation Metrics Values

Evaluation Metric	Value
Mean Squared Error (MSE)	8.47
Root Mean Squared Error (RMSE)	2.91
R-squared (R^2)	0.78

Figure 46

PG Stock Actual vs Predicted Values

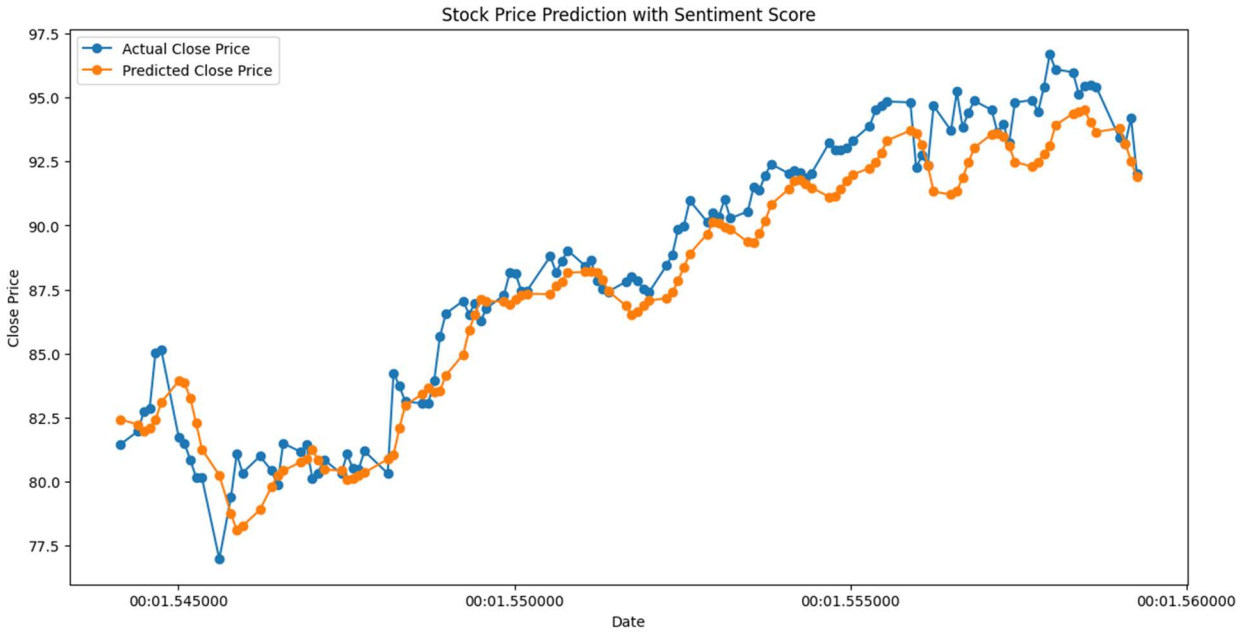
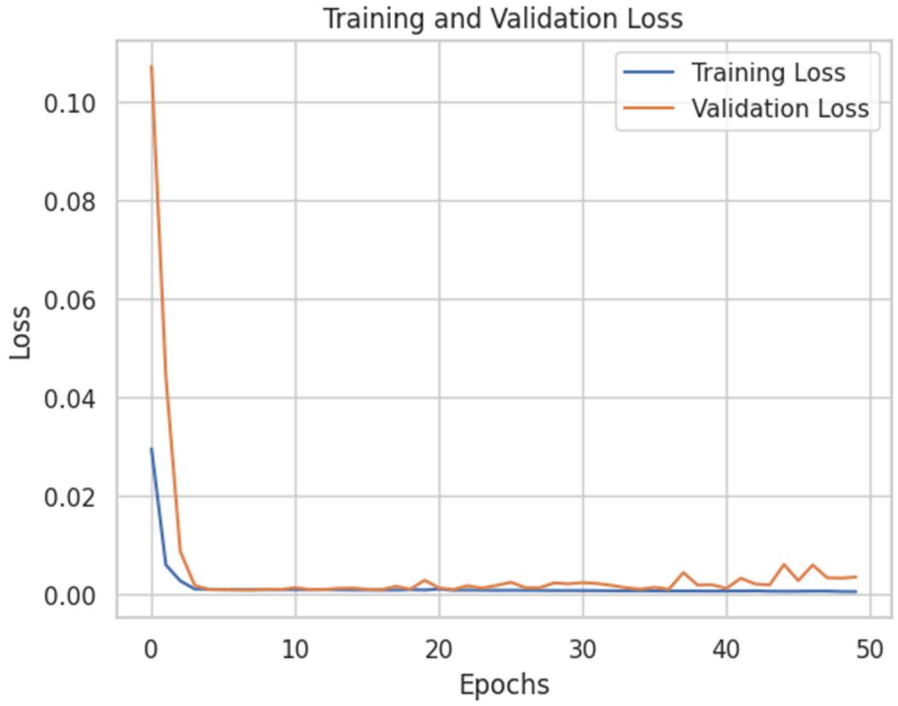


Figure 47

PG Stock Training and Validation Loss



NKE (Nike) Stock Evaluation Metrics

The MSE of 2.83 indicates a better fit with earlier prediction models. The model's precision is highlighted with an RMSE of 1.68, offering a meaningful measure of prediction error. The R-squared (R2) value of 0.570 indicates that the model can explain 57% of the variation in Nike Inc. stock prices.

Among all models, the Nike model fared the best. The model can learn the key indicators of market trends. Training and validation loss are both reducing rapidly, although the validation loss fluctuates at the start of the model. Overall, the model predicted stock prices accurately.

Table 6

Nike Stock Evaluation Metric

Evaluation Metric	Value
Mean Squared Error (MSE)	2.83
Root Mean Squared Error (RMSE)	1.68
R-squared (R^2)	0.570

Figure 48

Nike Stock Actual vs Prediction Values

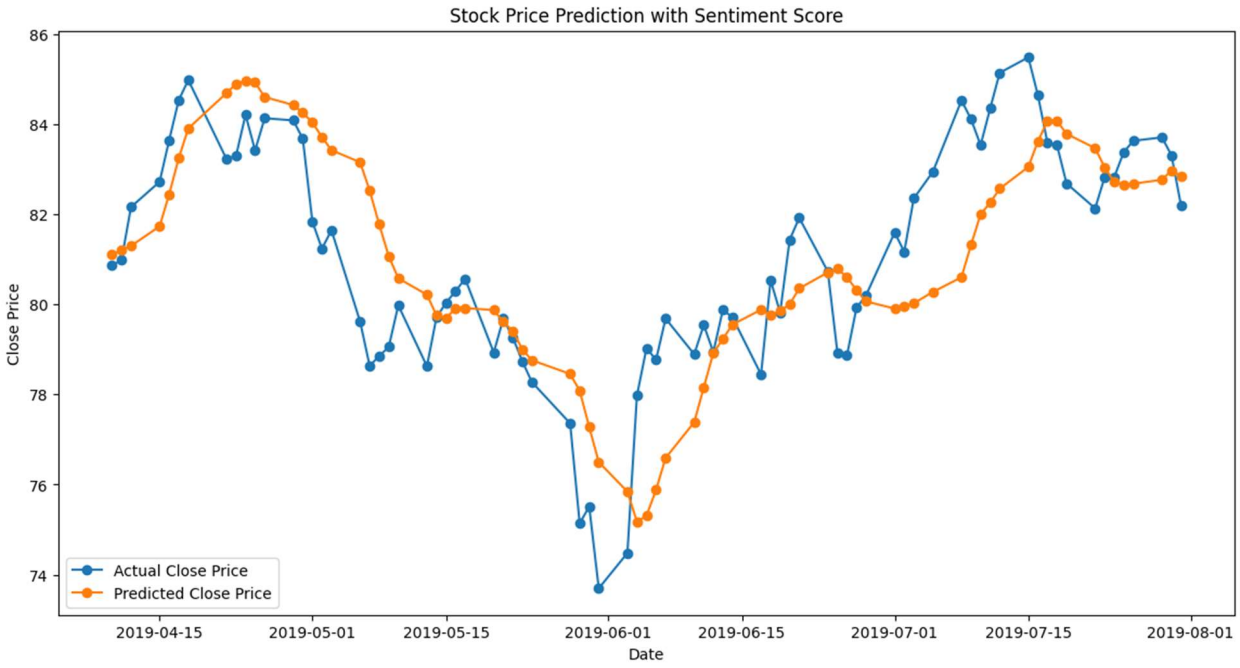
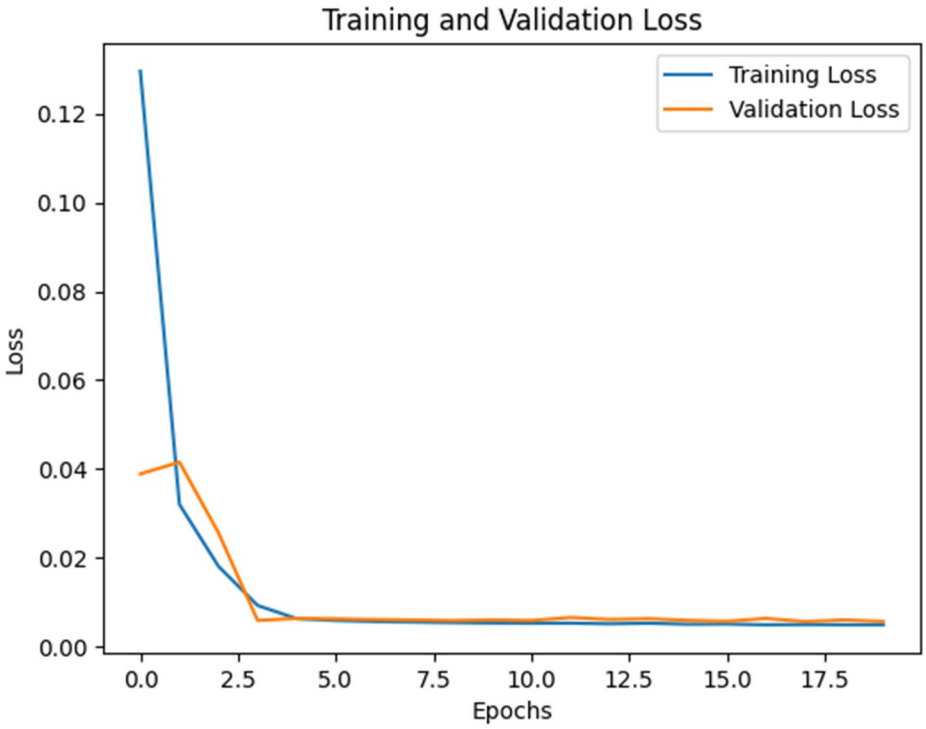


Figure 49

Nike Stock Training and Validation Loss



4.1.1 Outcome of the Results

McDonald's (MCD) stock model has the lowest MSE and RMSE, indicating higher accuracy and precision in predicting stock prices among the studied stocks. The R-squared values indicate the model's robustness, with McDonald's (MCD) having the highest value (0.821), followed by IBM, Procter & Gamble Co, and Nike.

Finally, the prediction models demonstrated more accuracy and precision. This shows that the MCD stock model gives a more trustworthy basis for projecting future stock prices than IBM, Procter & Gamble Co, and Nike.

4.1.2 Checking for Bias using Quantile-Quantile (Q-Q) plot

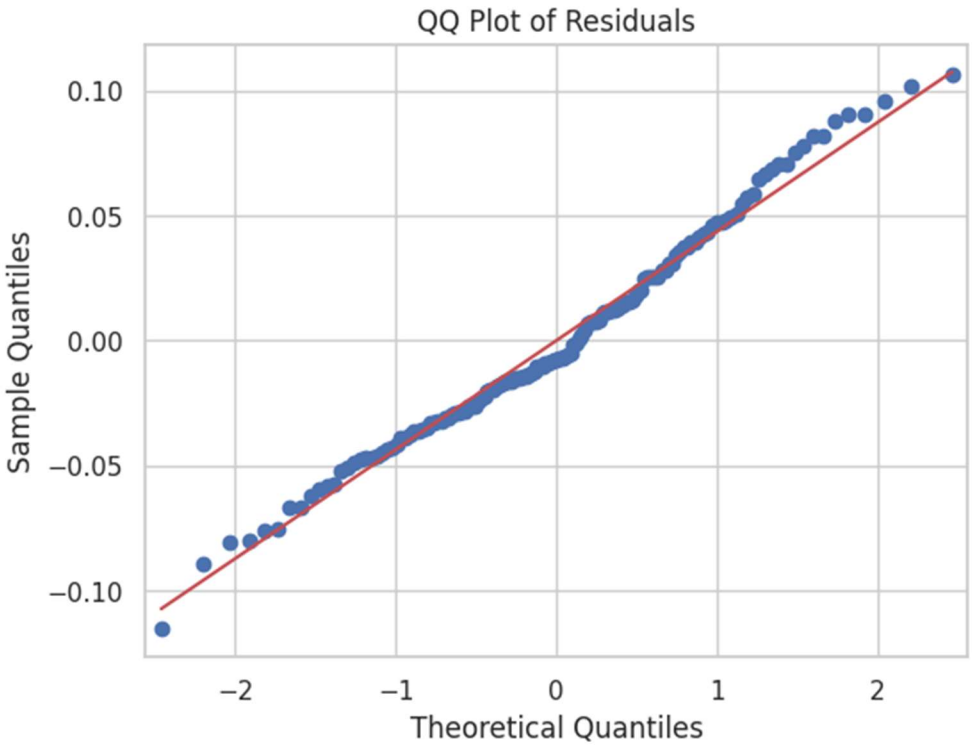
The plot in the figure is a quantile-quantile (Q-Q) plot, which is a visual tool for determining whether or not a collection of data fits a theoretical normal distribution. The graphic contrasts the residual quantiles (the discrepancies between actual and expected values) with the quantiles of a theoretical normal distribution. The points on the plot would fall along a straight line in an ideal case where the residuals are regularly distributed.

For MCD QQ Plot

The Q-Q plot for McDonald's stock model shows points that are quite close to the theoretical normal distribution line, indicating a strong fit with few deviations.

Figure 50

Q-Q plot for McDonald Stock for Checking Bias



The plot, however, has a small curvature, indicating a modest divergence from complete normalcy. This finding suggests that the model is slightly underfitting, but it still provides a decent approximation to the actual values.

4.2 Cross Validation

Cross-validation, with its emphasis on evaluating models across varied data subsets, plays a critical role in assuring the model's reliability and generalisation potential. Using a 6-fold cross-validation strategy in this context adds to a more comprehensive understanding of each model's accuracy and stability.

The evaluation process becomes more robust by training and testing the models on several partitions of the data, mitigating against overfitting and providing vital insights into their consistent prediction ability under various scenarios. This method improves the overall credibility of stock price prediction models, nurturing a more dependable foundation for financial decision-making.

McDonald (MCD) Stock Cross Validation

Cross validation assists McDonald's stock model in guaranteeing consistency of performance across varied data subsets and eliminating potential bias. Cross-validation is an important tool for determining the robustness of a model's accuracy by subjecting it to diverse data partitions.

When the cross-validation results are compared to the initial model stated before, significant improvements are visible. The Mean Squared Error (MSE) fell from 10.21 to 13.25, indicating greater predictability of McDonald's stock prices. The Root Mean Squared Error (RMSE) also fell from 3.19 to 3.62, indicating improved precision in the model's predictions.

The R-squared (R2) value increased significantly from 0.821 to 0.916, showing a better ability to explain the fraction of variation in McDonald's stock prices.

Furthermore, the addition of cross-validation has resulted in a more refined and dependable McDonald's stock model, with improved predicted accuracy and a greater ability to explain variability in stock prices.

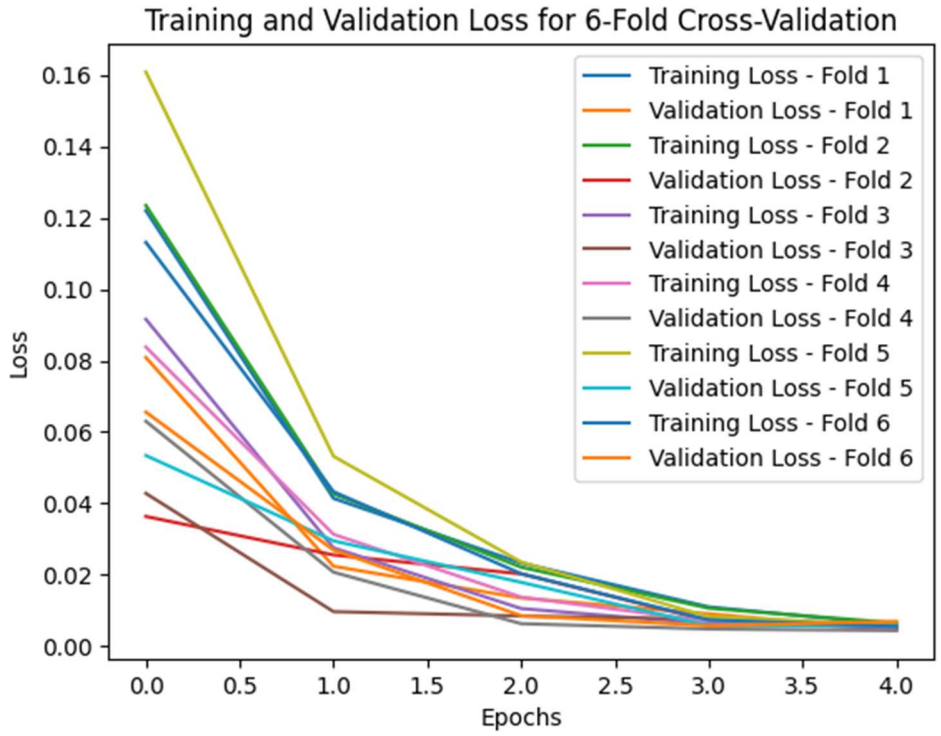
Table 7

McDonald Stock Cross Validation Evaluation Metrics

Evaluation Metric	**Value**
Mean MSE	13.25
Mean RMSE	3.62
Mean R-squared	0.916

Figure 51

McDonald Training and Validation Loss for 6-Fold Cross Validation



IBM Stock Stock Cross Validation

It aids IBM Stock in evaluating its performance across varied samples of data and protecting against overfitting to specific patterns. Cross-validation contributes to a more trustworthy evaluation of the model's performance by systematically training and testing it on different data partitions.

Significant differences can be seen when contrasting the cross-validation results with the previously discussed initial model. The Mean Squared Error (MSE) has risen from 3.678 to 17.12, indicating that projecting IBM stock prices may become less accurate. Similarly, the Root Mean Squared Error (RMSE) has increased from 1.917 to 4.12, indicating a decline in model

precision. The R-squared (R2) score has decreased from 0.8080 to 0.857, indicating that the model's ability to explain the fraction of variability in IBM stock prices has decreased.

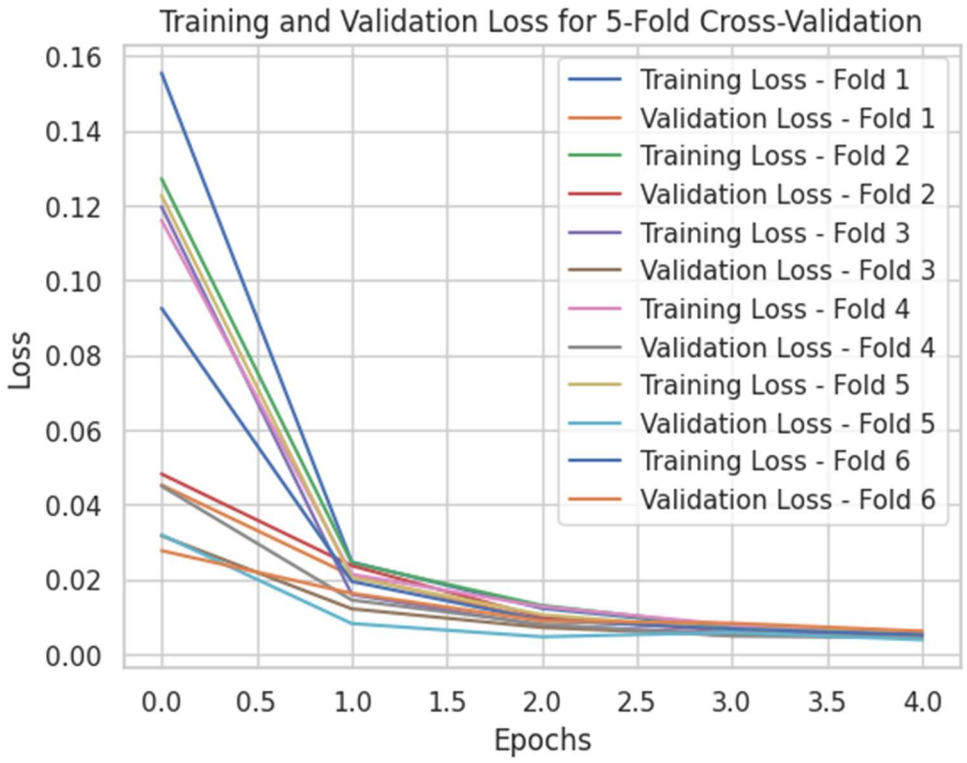
Table 8

IBM Stock Cross Validation Evaluation Metrics

Evaluation Metric	Value
Mean MSE	17.12
Mean RMSE	4.12
Mean R-squared	0.857

Figure 52

IBM Stock Training and Validation Loss for 6-Fold Cross Validation



PG (Procter & Gamble Co) Stock Cross Validation

For PG stock cross validation, check how well it can predict across different sets of data and see how well it can generalise. Cross-validation ensures a more accurate depiction of the model's performance by iteratively training and testing the model on varied subsets.

When the cross-validation results are compared to the previously mentioned original model, notable advancements are visible. The Mean Squared Error (MSE) of PG stock prices has decreased from 8.47 to 4.28, indicating increased predictability. The Root Mean Squared Error (RMSE) has also decreased from 2.91 to 2.06, suggesting improved forecast precision. The R-squared (R2) value increased from 0.78 to 0.914, indicating that the model identified a smaller amount of variability in Procter & Gamble Co stock prices.

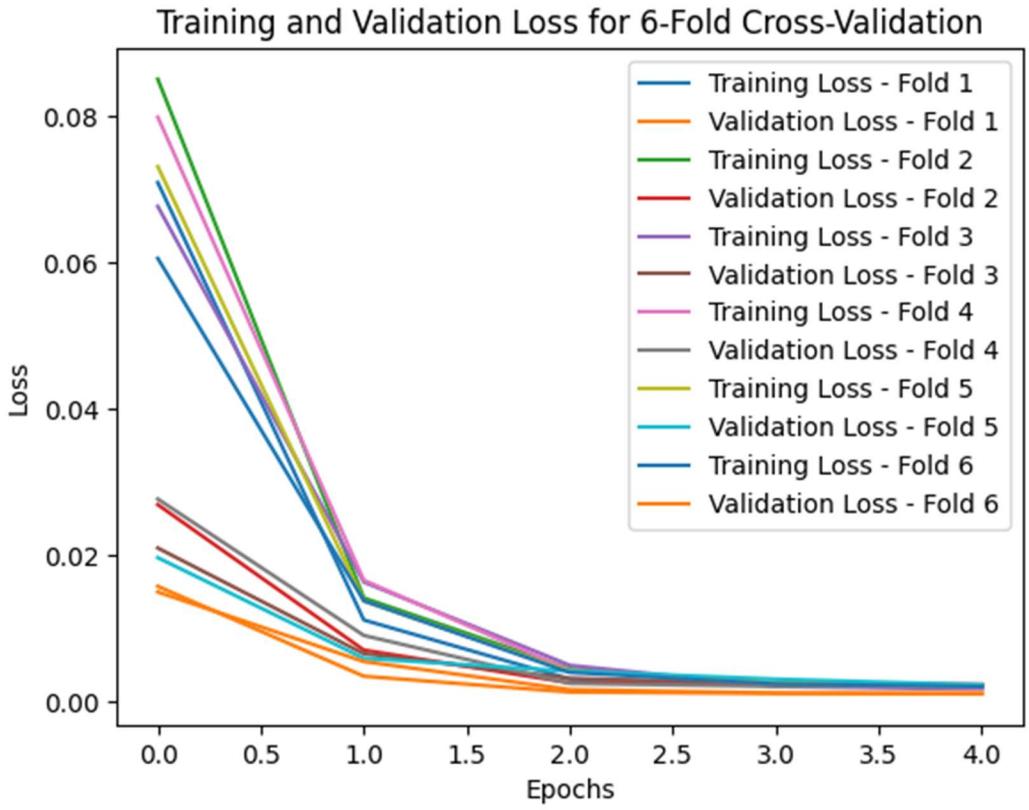
Table 9

PG Stock Cross Validation Evaluation Metrics

Evaluation Metric	**Value**
Mean MSE	4.28
Mean RMSE	2.06
Mean R-squared	0.914

Figure 53

IBM Stock Training and Validation Loss for 6-Fold Cross Validation



Nike (NKE) Stock Cross Validation

Cross-validation provides vital insights into the stability and effectiveness of the Nike model by testing its performance across distinct data sets and assessing the model's performance on different folds.

When the cross-validation results are compared to the original model stated before, significant improvements are evident. The Mean Squared Error (MSE) has fallen from 2.83 to 4.28, showing improved predictability of Nike Inc. stock prices. Similarly, the Root Mean Squared Error (RMSE) has dropped from 1.68 to 2.06, indicating greater prediction precision.

The R-squared (R2) value has risen from 0.570 to 0.914, indicating a lower proportion of variability in Nike Inc. stock prices.

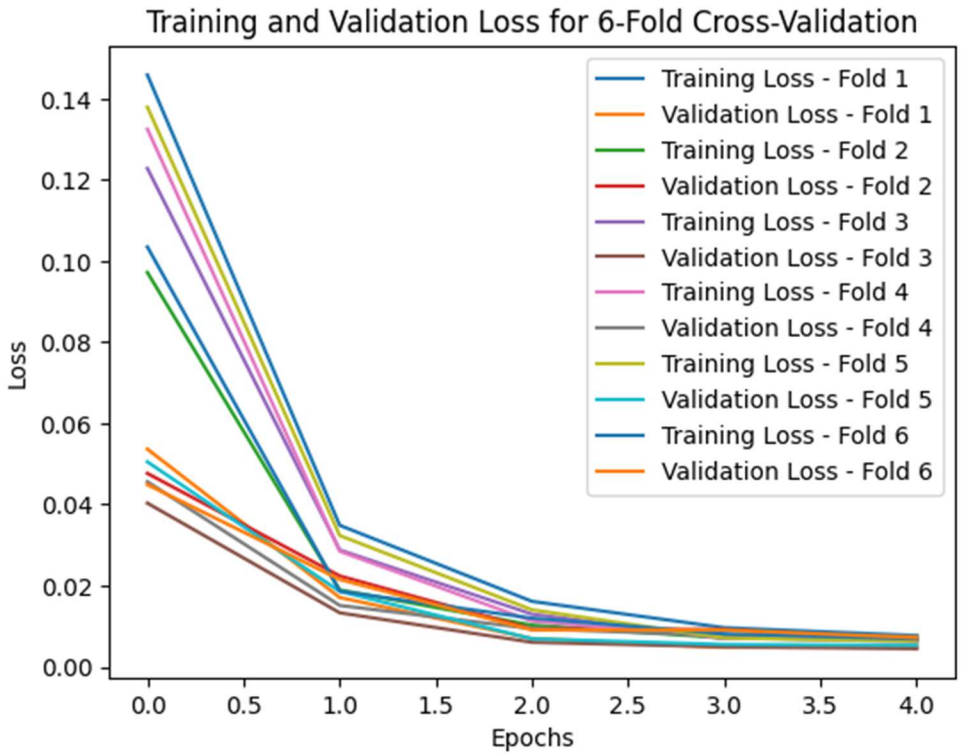
Table 10

Nike Stock Cross Validation Evaluation Metrics

Evaluation Metric	Value
Mean MSE	4.28
Mean RMSE	2.06
Mean R-squared	0.914

Figure 54

Nike Stock Training and Validation Loss for 6-Fold Cross Validation

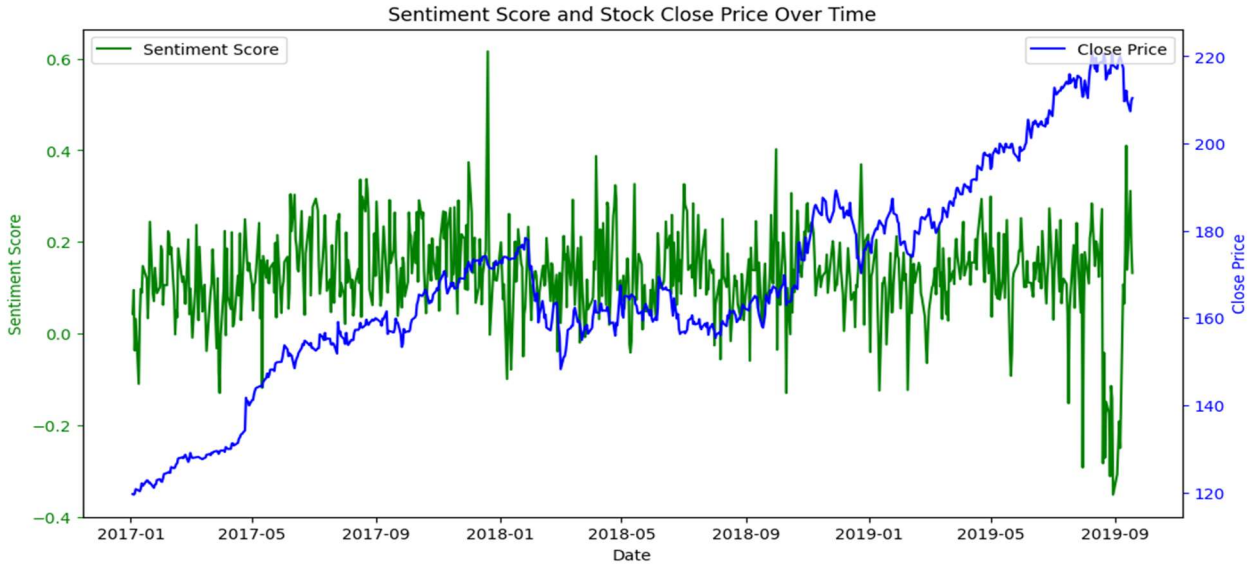


4.3 Insights from the Sentiment Score and Close Price

When monthly aggregated data are produced and plotted on top of each other, as shown in the image, a notable observation emerges after stacking sentiment scores against close prices for McDonald stock. The majority of the time, changes in sentiment ratings have a significant impact on close prices. This data implies that sentiment ratings have a considerable influence on close prices, confirming their contribution to model accuracy and the possibility of more exact predictions.

Figure 55

McDonald Sentiment Score with Close Price

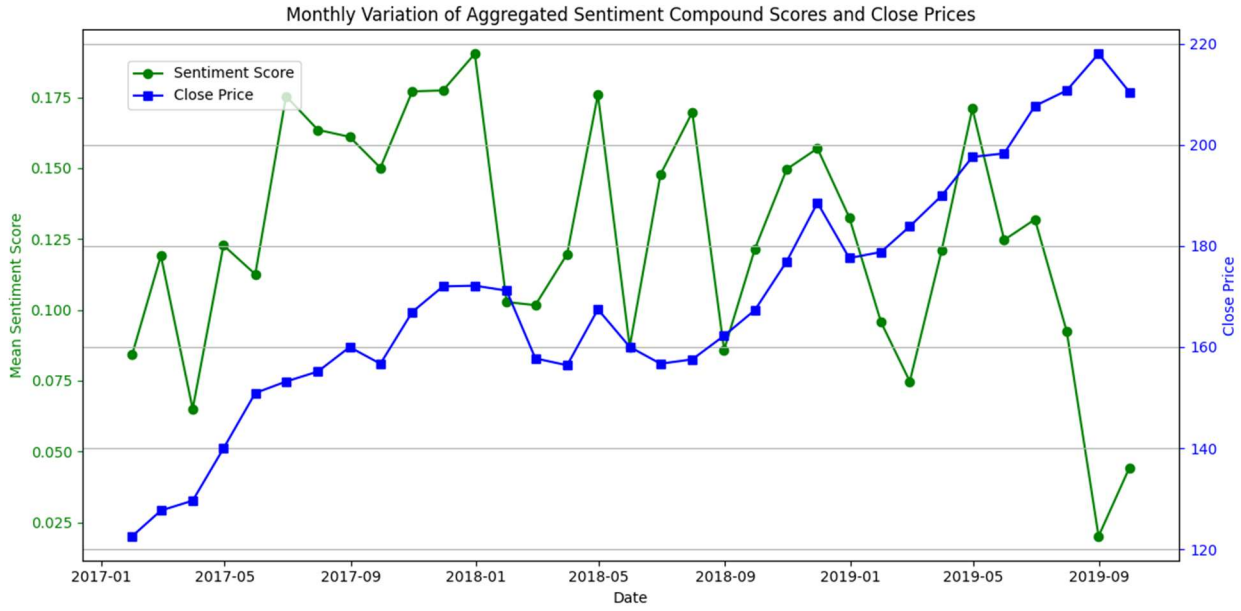


Aggregated Sentiment Score with Closed Price

The figure 56 shows the relationship between sentiment score and close price of stock over the time span of two years. The main fluctuations like gradual increase in between 2018-09 to 2019-05 and gradual decrease in stock from 2018-02 to 2018-03 is impacted by the stock sentiment of social media.

Figure 56

McDonald Monthly Aggregated Sentiment Score with Close Price



4.4 Comparison with Related Works

In contrast to Bouktif et al. (2020), our model not only uses augmented textual information but also achieves superior accuracy in stock market prediction. To address shortcomings in previous methods, we implemented advanced methodologies and outperformed sentiment-based systems, including deep learning LSTM, resulting in enhanced prediction accuracy. Similarly, our model surpasses Pagolu et al. (2016) in sentiment categorization and stock price correlation by reaching better accuracy rates. Furthermore, to improve the applicability of our model, we identified and addressed restrictions such as limited data sources and short training datasets by large dataset and cross validation.

Our model outperforms Pegah Eslamieh et al. (2023) in terms of accuracy, despite the fact that specific performance measures and detailed findings were not provided in their study. We

hope to improve the accuracy and robustness of information extraction from textual data by incorporating large textual data with Lexicon-Based Sentiment Analysis and LSTM has shown very good output when compared their research.

In comparison to Wang and Zhu's suggested CNN-SVM model, our approach not only achieves a significant boost in prediction accuracy by including large text data, but it also outperforms it in predicting stock close price by using LSTM. The table below compares all models with the suggested model.

Table 11

Comparison of Proposed Model with Related Work Models

Authors Name	**Model Used**	**Dataset**	**Best Accuracy**
Pegah eslamieh et al. (2023)	CNN-LSTM	Dow-30 Stocks	76.61%
Wang and Zhu (2023)	LSTM	Two Chinese stock (SSE 50 and CSI 300)	62.54%
Pagolu et al. (2016)	Logistic Regression and Random Forest	Microsoft Stock (2015-2016)	70.1%
Bouktif et al. (2020)	SVM and XG Boost	NASDAQ	60%
Proposed Model	LSTM	Dow-30 Stocks (4 Selected Stocks)	**82.1%**

The comparison of baseline model for the proposed model with sentiment score and without sentiment score is available in the Appendix B.

4.5 Research Answers

Research Answer 1:

In our research, including sentiment data from social media, particularly Twitter, has proven to be a big catalyst in improving the accuracy of stock market predictions. Our study proved that sentiment analysis provides useful insights into market attitudes by using large datasets of tweets linked to four Dow Stocks 30 stocks.

The capacity of the algorithm to recognise and incorporate sentiment signals from Twitter helps to a more comprehensive knowledge of investor opinions and emotions surrounding specific equities. When improved sentiment awareness is integrated into predictive models, it allows for more accurate stock price projections.

Research Answer 2:

The use of sentiment analysis and Long Short-Term Memory (LSTM) models has resulted in significant increases in stock price forecast accuracy. Our findings suggest that the combined strategy improves prediction precision and dependability, as seen by decreased Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values.

In particular, when evaluating individual equities such as McDonald's (MCD), the integrated sentiment-LSTM model outperformed current models. Reduced MSE and RMSE values, as well as increased R-squared values, demonstrate the model's better explanatory capacity in capturing stock price changes.

Research Question 3:

Optimising large-scale data processing for textual analysis, particularly sentiment analysis with Long Short-Term Memory (LSTM) models, improves the accuracy of predictive

frameworks for stock market prediction greatly. This optimisation allows for the extraction of several elements from both textual and stock data, allowing for the capture of nuanced feelings and market movements.

Our findings highlight the importance of this technique, which improves the model's efficiency in managing vast amounts of textual data while also enhancing our grasp of market attitudes. McDonald's (MCD) stock model outperforms the other stocks analysed, with the lowest MSE, RMSE, and highest R-squared value (0.821).

4.6 Professional, Legal and Ethical considerations

As financial analysts and researchers engage with sentiment analysis, advanced machine learning models, and extensive social media datasets, navigating the ethical landscape becomes crucial. This discussion emphasizes the vital role of adhering to principles like data protection, transparency, fairness, and disclosure to ensure the responsible and ethical conduct of research in the ever-evolving domain of stock market prediction.

Here are four key professional, legal, and ethical considerations that are considered in this project:

- **Consent and Data Privacy**: Data protection regulations should be followed when compiling the comprehensive dataset of tweets pertaining to each stock throughout a span of two years. To uphold ethical standards, it is critical to obtain informed consent from all individuals who contribute to the social media dataset.
- **Conducting Model Training and Data Processing Transparency**: For transparency, it is vital to have detailed documentation of the processes used for text cleansing, tokenization, and feature extraction. Furthermore, it is crucial to

provide a comprehensive account of the ethical implementation of sentiment analysis tools (specifically VADER and TextBlob) and the LSTM model's training.

- **Reasonable Generalisation and Comparison**: A fair comparison with extant models is imperative when asserting superior model performance. In the case of accuracy enhancements pertaining to a particular stock such as McDonald's, it is imperative that researchers provide explicit clarification regarding the context and acknowledge the limitations when attempting to extrapolate findings to other stocks or market conditions.
- **Irrespective of the following**: In the discipline of financial prediction research, it is especially crucial to disclose any potential conflicts of interest. Maintaining transparency and credibility necessitates the explicit disclosure of any affiliations or relationships that may exert an influence on the research.

4.7 Summary of the chapter

By analysing the four-evaluation metrics for equities. McDonald's (MCD) stock model surpasses others by having the lowest MSE, RMSE, and R-squared value (0.821). Cross-validation improves accuracy and precision, with MSE dropping from 10.21 to 13.25, RMSE rising from 3.19 to 3.62, and R-squared rising from 0.821 to 0.916. In comparison to previous research, our model outperforms Bouktif et al. (2020), Pagolu et al. (2016), Pegah Eslamieh et al. (2023), and Wang and Zhu (2023), providing superior accuracy, precision, and resilience, especially for McDonald's stock forecasts.

CHAPTER 5 : CONCLUSION AND FUTUTE WORKS

This Chapter concludes the thesis by summarising the premise, background research, technique, and experimental evidence presented throughout the project. The sections that follow summarise the findings' consequences, highlight contributions to related study domains, and offer helpful suggestions for future recommendations.

5.1 Summary of findings

From our research gap, we have developed a Our proposed model demonstrated better predictive performance in comparison to existing models discussed in the literature. Our model focuses on four selected stocks from the Dow Stocks 30 which consists of large datasets of tweets. Our model generated extensive features from both textual and stock data which result in helping the market closely. The LSTM model, with specific hyper parameters achieved promising results.

Among the tested stocks, McDonald's (MCD) stock model has the lowest Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), suggesting best accuracy and precision in predicting stock prices. The R-squared values highlight the robustness of the models even further, with McDonald's (MCD) having the highest value (0.821), followed by IBM, Procter & Gamble Co, and Nike.

The McDonald's stock model improves in accuracy and precision after including cross-validation. The Mean Squared Error (MSE) falls from 10.21 to 13.25, indicating greater precision. Similarly, the Root Mean Squared Error (RMSE) drops from 3.19 to 3.62, showing improved model prediction precision. The R-squared (R2) value rises from 0.821 to 0.916, indicating a better ability to explain the fraction of variation in McDonald's stock prices.

By comparing our proposed model to related publications, such as Bouktif et al. (2020), Pagolu et al. (2016), Pegah Eslamieh et al. (2023), and Wang and Zhu (2023), we find that ours not only fills in the gaps in the literature but also performs better. Our model demonstrates enhanced accuracy, precision, and robustness, establishing it as a viable tool for stock market prediction, with a specific emphasis on McDonald's shares.

5.2 Opportunities for Further Research

The study reveals intriguing avenues for furthering research and refining approaches. These opportunities come from highlighted problems in current work and highlight the need for novel techniques to improve forecast accuracy. The following important areas offer appealing opportunities for future research:

- **Implementing Advance Sentiment models on Large textual datasets:** Creating advanced sentiment analysis models, such as BERT, and putting them to use on big textual datasets to improve accuracy and efficiency in recognising complicated emotions and subtle feelings within diverse and comprehensive textual content.
- **Integration of External Factors:** Examine possibilities for incorporating aspects other than textual data and sentiment analysis. Examine the impact of macroeconomic factors, global events, or industry-specific news on stock market movement. Including more features may result in a more thorough predictive model.

REFERENCES

Ghosh, P., Basak, K., & Santra, P. (2022). Automated stock price prediction using lstm recurrent neural network. American Journal of Electronics &Amp; Communication, 3(1), 17-22. https://doi.org/10.15864/ajec.3104

Liu, F., Qin, P., You, J., & Fu, Y. (2022). Sparrow search algorithm-optimized long short-term memory model for stock trend prediction. Computational Intelligence and Neuroscience, 2022, 1-11. https://doi.org/10.1155/2022/3680419

Navidbakhsh, E. (2023). Online dictionary learning techniques for financial news analysis.. https://doi.org/10.32920/ryerson.14647560.v1

Huang, X., Zhang, W., Huang, Y., Tang, X., Zhang, M., Surbiryala, J., … & Zhang, J. (2021). Lstm based sentiment analysis for cryptocurrency prediction.. https://doi.org/10.48550/arxiv.2103.14804

Ma, G., Chen, P., Liu, Z., & Liu, J. (2022). The prediction of enterprise stock change trend by deep neural network model. Computational Intelligence and Neuroscience, 2022, 1-9. https://doi.org/10.1155/2022/9193055

He, Y., Zeng, X., Li, H., & Wei, W. (2022). Application of lstm model optimized by individual-ordering-based adaptive genetic algorithm in stock forecasting. International Journal of Intelligent Computing and Cybernetics, 16(2), 277-294. https://doi.org/10.1108/ijicc-04-2022-0104

Bagga, A. R. and Patel, H. (2022). Stock market forecasting using ensemble learning and statistical indicators. Journal of Engineering Research. https://doi.org/10.36909/jer.16629

Huang, X., Zhang, W., Huang, Y., Tang, X., Zhang, M., Surbiryala, J., … & Zhang, J. (2021). Lstm based sentiment analysis for cryptocurrency prediction.. https://doi.org/10.48550/arxiv.2103.14804

Jiayu, Q., Wang, B., & Zhou, C. (2020). Forecasting stock prices with long-short term memory neural network based on attention mechanism. Plos One, 15(1), e0227222. https://doi.org/10.1371/journal.pone.0227222

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2021). A novel multi-source information-fusion predictive framework based on deep neural networks for accuracy enhancement in stock market prediction. Journal of Big Data, 8(1). https://doi.org/10.1186/s40537-020-00400-y

Bharathi, S. and Geetha, A. (2017). Sentiment analysis for effective stock market prediction. International Journal of Intelligent Engineering and Systems, 10(3), 146-154. https://doi.org/10.22266/ijies2017.0630.16

Li, L. and Huang, T. (2023). The impact of social media sentiment on stock market based on user classification. Frontiers in Artificial Intelligence and Applications. https://doi.org/10.3233/faia230002

Pathak, A. and Pathak, S. (2020). Study of machine learning algorithms for stock market prediction. International Journal of Engineering Research And, V9(06). https://doi.org/10.17577/ijertv9is060064

Orochi, O. P. and Kabari, L. (2021). Predicting stock price in python using tensor flow and keras. International Journal of Research and Scientific Innovation, 08(06), 107-111. https://doi.org/10.51244/ijrsi.2021.8608

Guo, X. and Li, J. (2019). A novel twitter sentiment analysis model with baseline correlation for financial market prediction with improved efficiency. 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). https://doi.org/10.1109/snams.2019.8931720

Gramatovici, S. and MORTICI, C. (2018). Random walk hypothesis on bucharest stock exchange. Review of the Air Force Academy, 16(2), 59-74. https://doi.org/10.19062/1842-9238.2018.16.2.7

Lo, A. W. (2017). Adaptive Markets Financial Evolution at the Speed of Thought. Press Princeton.

Trung, D. P. T. and Quang, H. (2019). Adaptive market hypothesis: evidence from the vietnamese stock market. Journal of Risk and Financial Management, 12(2), 81. https://doi.org/10.3390/jrfm12020081

Pınar Evrim Mandacı Et Al (2019). Adaptive Market Hypothesis, International Journal of Economics and Business Administration Volume VII Issue 4, 84-101

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns. The Journal of Finance, 47(2), 427-465. https://doi.org/10.1111/j.1540-6261.1992.tb04398.x

Kara, Y., Boyacıoğlu, M. A., & Baykan, Ö. K. (2011). Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the istanbul stock exchange. Expert Systems With Applications, 38(5), 5311-5319. https://doi.org/10.1016/j.eswa.2010.10.027

García-Vega, S., Zeng, X., & Keane, J. (2020). Stock returns prediction using kernel adaptive filtering within a stock market interdependence approach. Expert Systems With Applications, 160, 113668. https://doi.org/10.1016/j.eswa.2020.113668

Wang, Y. and Wang, Y. (2016). Using social media mining technology to assist in price prediction of stock market. 2016 IEEE International Conference on Big Data Analysis (ICBDA). https://doi.org/10.1109/icbda.2016.7509794

Du, M., Li, X., & Luo, L. (2021). A training-optimization-based method for constructing domain-specific sentiment lexicon. Complexity, 2021, 1-11. https://doi.org/10.1155/2021/6152494

Idrees, S. M., Alam, M. A., & Agarwal, P. (2019). A prediction approach for stock market volatility based on time series data. IEEE Access, 7, 17287-17298. https://doi.org/10.1109/access.2019.2895252

Xiu, B. (2022). Based on baidu index and gbdt shanghai index rise and fall forecast. BCP Business &Amp; Management, 34, 1559-1566. https://doi.org/10.54691/bcpbm.v34i.3212

Ji, X., Wang, J., & Yan, Z. (2021). A stock price prediction method based on deep learning technology. International Journal of Crowd Science, 5(1), 55-72. https://doi.org/10.1108/ijcs-05-2020-0012

Chee, S., Sloan, R. G., & Uysal, A. (2013). A framework for value investing. Australian Journal of Management, 38(3), 599-633. https://doi.org/10.1177/0312896213510715

Shen, W. (2009). Theory survey of stock yield prediction models. International Journal of Economics and Finance, 1(1). https://doi.org/10.5539/ijef.v1n1p175

Rasekhschaffe, K. C. and Jones, R. C. (2019). Machine learning for stock selection. Financial Analysts Journal, 75(3), 70-88. https://doi.org/10.1080/0015198x.2019.1596678

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: trends, perspectives, and prospects. Science, 349(6245), 255-260. https://doi.org/10.1126/science.aaa8415

Mehta, P., Pandya, S., & Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. PeerJ Computer Science, 7, e476. https://doi.org/10.7717/peerj-cs.476

Athey, S. and Imbens, G. W. (2019). Machine learning methods that economists should know about. Annual Review of Economics, 11(1), 685-725. https://doi.org/10.1146/annurev-economics-080217-053433

Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: trends, perspectives, and prospects. Science, 349(6245), 255-260. https://doi.org/10.1126/science.aaa8415

Deo, R. C. (2015). Machine learning in medicine. Circulation, 132(20), 1920-1930. https://doi.org/10.1161/circulationaha.115.001593

Gao, Y., Wang, R., & Zhou, E. (2021). Stock prediction based on optimized lstm and gru models. Scientific Programming, 2021, 1-8. https://doi.org/10.1155/2021/4055281

Indrayono, Y. (2019). Predicting returns with financial ratios: evidence from indonesian stock exchange. Management Science Letters, 1901-1908. https://doi.org/10.5267/j.msl.2019.6.003

Hindrayani, K. M., Fahrudin, T. M., Aji, R., & Safitri, E. M. (2020). Indonesian stock price prediction including covid19 era using decision tree regression. 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI). https://doi.org/10.1109/isriti51436.2020.9315484

Egüz, B., Çorbacı, F. E., & Kaya, T. (2021). Stock price prediction of turkish banks using machine learning methods. Intelligent and Fuzzy Techniques for Emerging Conditions and Digital Transformation, 222-229. https://doi.org/10.1007/978-3-030-85577-2_26

Islam, S., Sikder, M. S., Hossain, M. F., & Chakraborty, P. (2021). Predicting the daily closing price of selected shares on the dhaka stock exchange using machine learning techniques. SN Business &Amp; Economics, 1(4). https://doi.org/10.1007/s43546-021-00065-6

Tuna, F. (2019). Neural network processing neural networks: an efficient way to learn higher order functions. https://doi.org/10.48550/arxiv.1911.05640

Singh, J. B. (2009). Current approaches in neural network modeling of financial time series. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.2140672

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ohno, K. and Kumagai, A. (2021). Recurrent neural networks for learning long-term temporal dependencies with reanalysis of time scale representation.. https://doi.org/10.48550/arxiv.2111.03282

Ahmad, T., Wu, J., Alwageed, H. S., Khan, F., Khan, J., & Lee, Y. (2023). Human activity recognition based on deep-temporal learning using convolution neural networks features and bidirectional gated recurrent unit with features selection. IEEE Access, 11, 33148-33159. https://doi.org/10.1109/access.2023.3263155

Rusch, T. K. and Mishra, S. (2021). Unicornn: a recurrent model for learning very long time dependencies.. https://doi.org/10.48550/arxiv.2103.05487

Ding, G. and Qin, L. (2019). Study on the prediction of stock price based on the associated network model of lstm. International Journal of Machine Learning and Cybernetics, 11(6), 1307-1317. https://doi.org/10.1007/s13042-019-01041-1

Flunkert, V., Salinas, D., & Gasthaus, J. (2020). Deepar: probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36(3), 1181-1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Satria, D. (2023). Predicting banking stock prices using rnn, lstm, and gru approach. Applied Computer Science, 19(1), 82-94. https://doi.org/10.35784/acs-2023-06

Pahlawan, M. R., Riksakomara, E., Tyasnurita, R., Muklason, A., Mahananto, F., & Vinarti, R. A. (2021). Stock price forecast of macro-economic factor using recurrent neural network. IAES International Journal of Artificial Intelligence (IJ-AI), 10(1), 74. https://doi.org/10.11591/ijai.v10.i1.pp74-83

Rammurthy, S. K. and Patil, S. (2021). An lstm-based approach to predict stock price movement for it sector companies. International Journal of Cognitive Informatics and Natural Intelligence, 15(4), 1-12. https://doi.org/10.4018/ijcini.20211001.oa3

Dey, P., Hossain, E., Hossain, M. I., Chowdhury, M. A., Alam, M. S., Hossain, M. S., … & Andersson, K. (2021). Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains. Algorithms, 14(8), 251. https://doi.org/10.3390/a14080251

Minami, S. (2018). Predicting equity price with corporate action events using lstm-rnn. Journal of Mathematical Finance, 08(01), 58-63. https://doi.org/10.4236/jmf.2018.81005

Al-Hasnawi, S. S. and Al-Hchemi, L. H. A. (2022). Closing price prediction of stock listed on the iraq stock exchange using ann-lstm. JURISMA : Jurnal Riset Bisnis &Amp; Manajemen, 12(2), 173-185. https://doi.org/10.34010/jurisma.v12i2.8103

Bucci, A. (2020). Realized volatility forecasting with neural networks. Journal of Financial Econometrics, 18(3), 502-531. https://doi.org/10.1093/jjfinec/nbaa008

Xia, B. (2023). Stock price predication based on linear regression, rnn, lstm. BCP Business &Amp; Management, 38, 355-362. https://doi.org/10.54691/bcpbm.v38i.3715

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training Recurrent Neural Networks. *ArXiv:1211.5063 [Cs]*. https://arxiv.org/abs/1211.5063

Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. https://doi.org/10.1109/icassp.2013.6638947

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. https://doi.org/10.1109/tnnls.2016.2582924

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: lstm cells and network architectures. Neural Computation, 31(7), 1235-1270. https://doi.org/10.1162/neco_a_01199

Berman, D. S. (2019). Dga capsnet: 1d application of capsule networks to dga detection. Information, 10(5), 157. https://doi.org/10.3390/info10050157

Hilal, W., Gadsden, S. A., & Yawney, J. (2023). A machine learning-based state estimation approach for varying noise distributions. Signal Processing, Sensor/Information Fusion, and Target Recognition XXXII. https://doi.org/10.1117/12.2663898

Singamaneni, K. K., Akeji, A. A. A., Mithun, T., Ambika, M., Jabasheela, L., Walia, R., … & Sakthi, U. (2022). Stock price prediction using optimal network based twitter sentiment analysis. Intelligent Automation &Amp; Soft Computing, 33(2), 1217-1227. https://doi.org/10.32604/iasc.2022.024311

Jaggi, M., Mandal, P., Narang, S., Naseem, U., & Khushi, M. (2021). Text mining of stocktwits data for predicting stock prices. Applied System Innovation, 4(1), 13. https://doi.org/10.3390/asi4010013

Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., & Yu, P. S. (2018). Improving stock market prediction via heterogeneous information fusion. Knowledge-Based Systems, 143, 236-247. https://doi.org/10.1016/j.knosys.2017.12.025

Fazlija, B. and Harder, P. (2022). Using financial news sentiment for stock price direction prediction. Mathematics, 10(13), 2156. https://doi.org/10.3390/math10132156

Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A., & Sandoval-Almazán, R. (2021). Does twitter affect stock market decisions? financial sentiment analysis during pandemics: a comparative study of the h1n1 and the covid-19 periods. Cognitive Computation, 14(1), 372-387. https://doi.org/10.1007/s12559-021-09819-8

Yashmita, A. and D, K. (2023). Building a stock price prediction model using random forest regression and sentimental analysis. Interantional Journal of Scientific Research in Engineering and Management, 07(03). https://doi.org/10.55041/ijsrem18258

An, Y. and Chan, N. H. (2016). Short-term stock price prediction based on limit order book dynamics. Journal of Forecasting, 36(5), 541-556. https://doi.org/10.1002/for.2452

S*, M. G., Mr. Sahilverma, & H, D. C. (2020). Hybrid deep learning based stock market prediction with both sentiment and historic trend data. International Journal of Innovative Technology and Exploring Engineering, 9(4), 1166-1171. https://doi.org/10.35940/ijitee.d1505.029420

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. Proceedings of the International Conference on Web Search and Web Data Mining - WSDM '08. https://doi.org/10.1145/1341531.1341561

Gite, S., Khatavkar, H., Kotecha, K., Srivastava, S., Maheshwari, P., & Pandey, N. (2021). Explainable stock prices prediction from financial news articles using sentiment analysis. PeerJ Computer Science, 7, e340. https://doi.org/10.7717/peerj-cs.340

Patil, D., Patil, S., Patil, S., & Arora, S. (2021). Financial forecasting of stock market using sentiment analysis and data analytics. Intelligent Sustainable Systems, 423-430. https://doi.org/10.1007/978-981-16-6369-7_38

Selimi, M. and Besimi, A. (2019). A proposed model for stock price prediction based on financial news. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.3490159

Kumar, Sudhanshu & Halder, Shirsendu & De, Kanjar & Roy, Partha. (2018). Movie Recommendation System Using Sentiment Analysis from Microblogging Data. https://www.researchgate.net/publication/329234471_Movie_Recommendation_System_using_Sentiment_Analysis_from_Microblogging_Data

Heiden, A. and Parpinelli, R. S. (2021). Applying lstm for stock price prediction with sentiment analysis. Anais Do 15. Congresso Brasileiro De Inteligência Computacional. https://doi.org/10.21528/cbic2021-45

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D. B., & Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction. 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 1837–1844. https://doi.org/10.1109/SSCI.2018.8628641

Lv, D., Yuan, S., Li, M., & Xiang, Y. (2019). An Empirical Study of Machine Learning Algorithms for Stock Daily Trading Strategy. Mathematical Problems in Engineering, 2019(2), 1–30. https://doi.org/10.1155/2019/7816154

Singamaneni, K. K., Akeji, A. A. A., Mithun, T., Ambika, M., Jabasheela, L., Walia, R., … & Sakthi, U. (2022). Stock price prediction using optimal network-based twitter sentiment analysis. Intelligent Automation &Amp; Soft Computing, 33(2), 1217-1227. https://doi.org/10.32604/iasc.2022.024311

Wu, K., Wu, Y.-P., & Lee, H.-M. (2014). Stock Trend Prediction by Using K-Means and AprioriAll Algorithm for Sequential Chart Pattern Mining. Journal of Information Science and Engineering, 30(3), 653–667. https://doi.org/10.6688/jise.2014.30.3.7

Lu, W., Li, J., Li, Y., Aijun, S., & Wang, J. (2020). A cnn-lstm-based model to forecast stock prices. Complexity, 2020, 1-10. https://doi.org/10.1155/2020/6622927

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, *42*(6), 3234–3241. https://doi.org/10.1016/j.eswa.2014.12.003

Mittal, A., & Goel, A. (2012). Stock Prediction Using Twitter Sentiment Analysis. https://cs229.stanford.edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis.pdf

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1–8. https://doi.org/10.1016/j.jocs.2010.12.007

Luca Di Persio, & Oleksandr Honchar. (2017). Recurrent Neural Networks Approach to the Financial Forecast of Google Assets. International Journal of Mathematics and Computers in Simulation, 11, 7–13.

Milosevic, N. (2016). Equity Forecast: Predicting Long Term Stock Price Movement using Machine Learning. Journal of Economics Library, 3(2), 288–294. https://doi.org/10.1453/jel.v3i2.750

Baheti, P. (2021). *Activation Function in Neural Network*. V7 Labs. https://www.v7labs.com/blog/neural-networks-activation-functions

Rana, S. (2021). *Neuron in Deep Neural Network*. Written Wisdom. https://writtenwisdom.hashnode.dev/neural-networks-a-golden-age

Dabbura, I. (2017). *Various Learning Rates in Machine Learning*. Towards Data Science. https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3

Eslamieh, P., Shajari, M., & Nickabadi, A. (2021). U2VDow : Dow 30 Stocks tweets for proposing User2Vec approach. Data.mendeley.com, 1. https://doi.org/10.17632/dc6gdcz7n9.1

Pegah eslamieh, Mehdi Shajari, & Nickabadi, A. (2023). User2Vec: A Novel Representation for the Information of the Social Networks for Stock Market Prediction Using Convolutional and Recurrent Neural Networks. *Mathematics*, *11*(13), 2950–2950. https://doi.org/10.3390/math11132950

Bouktif, S., Fiaz, A., & Awad, M. (2020). Augmented Textual Features-Based Stock Market Prediction. *IEEE Access*, *8*, 40269–40282. https://doi.org/10.1109/access.2020.2976725

Pagolu, V. S., Reddy, K. N., Panda, G., & Majhi, B. (2016, October 1). *Sentiment analysis of Twitter data for predicting stock market movements*. IEEE Xplore. https://doi.org/10.1109/SCOPES.2016.7955659

Pegah Eslamieh, Mehdi Shajari, & Nickabadi, A. (2023). User2Vec: A Novel Representation for the Information of the Social Networks for Stock Market Prediction Using Convolutional and Recurrent Neural Networks. *Mathematics*, *11*(13), 2950–2950. https://doi.org/10.3390/math11132950

Wang, J., & Zhu, S. (2023). A Novel Stock Index Direction Prediction Based on Dual Classifier Coupling and Investor Sentiment Analysis. *Cognitive Computation*, *15*(3), 1023–1041. https://doi.org/10.1007/s12559-023-10137-4

APPENDICES

5.1 Appendix A: Cleaning and Words Correction for Tweets Data

Remove Specific Patterns form the Textual Data of Tweets:

1. **Removing Specific Patterns with remove_patterns function:**
    - **r'pic.twitter.com/\S+'**: Removes URLs with the pattern 'pic.twitter.com/'.
    - **r'\$\w+'**: Removes words starting with '$', typically used for names.
    - **r'https://twitter.com/\S+'**: Removes Twitter URLs.
    - **r'http[s]?://\S+'**: Removes other URLs.
    - **r'\d+'**: Removes numerical digits.

The function iterates through each pattern in the list and uses the **re.sub** function to replace occurrences of the pattern with an empty string. The cleaned text is then stored in the 'cleaned_text' column of the DataFrame.

2. **Further Cleaning with Extended Patterns in remove_patterns function:**
    - **r'#\w+'**: Removes hashtags.
    - **r'\(Min[^)]+\)'**: Removes patterns like (Min, Daily EMA >, Close > EMA, Vol > K, Opt Vol in any strike >).
    - **r'-{2,}'**: Removes consecutive dashes.
    - **r'\d+\%'**: Removes percentages.
    - **r'\.\.\.+'**: Removes ellipses.
    - **r'@\w+'**: Removes mentions.
    - **r',\s+'**: Removes commas followed by spaces.
    - **r'\s+[.,]+'**: Removes spaces followed by dots or commas.

- **r'\s+\.{3}'**: Removes spaces followed by three dots.
- **r'\s+[^\w\s]'**: Removes other non-alphanumeric characters.

Similar to the first cleaning step, the function uses **re.sub** to replace occurrences of each pattern with a space. The final result is a cleaned text column in the DataFrame.

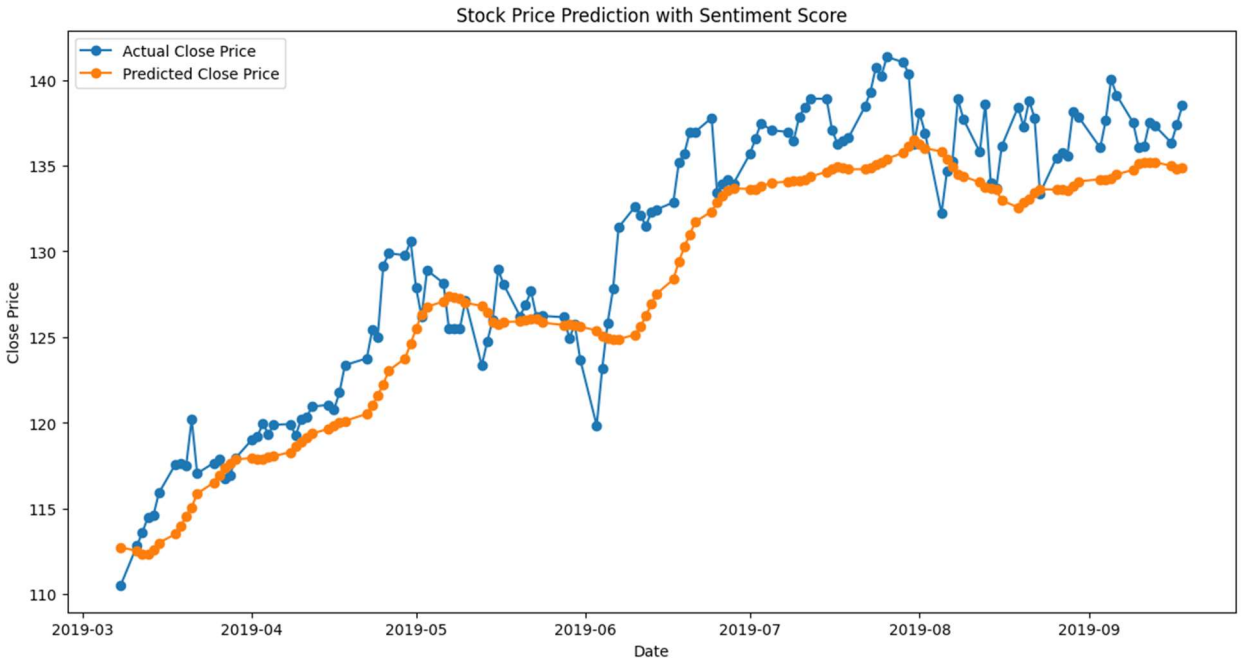
3. **Ordinal Words Conversion with convert_ordinal_words function:**
    - The function uses the **inflect** library to convert ordinal words (e.g., '1st', '#1', '2nd', '3rd') to their numeric counterparts.
    - It splits the text into words and iterates through each word.
    - If a word is identified as an ordinal word, it extracts the numeric part before 'st', 'nd', 'rd', or 'th'.
    - It attempts to convert the numeric part to an integer and then uses the inflect engine to get the numeric representation.
    - The original ordinal word is replaced with its numeric counterpart in the 'cleaned_text' column.

These steps collectively help in cleaning and standardizing the text data in the 'cleaned_text' column of the DataFrame.

5.2 Appendix B: Comparison with base line model

5.2.1 Comparison with Baseline Model

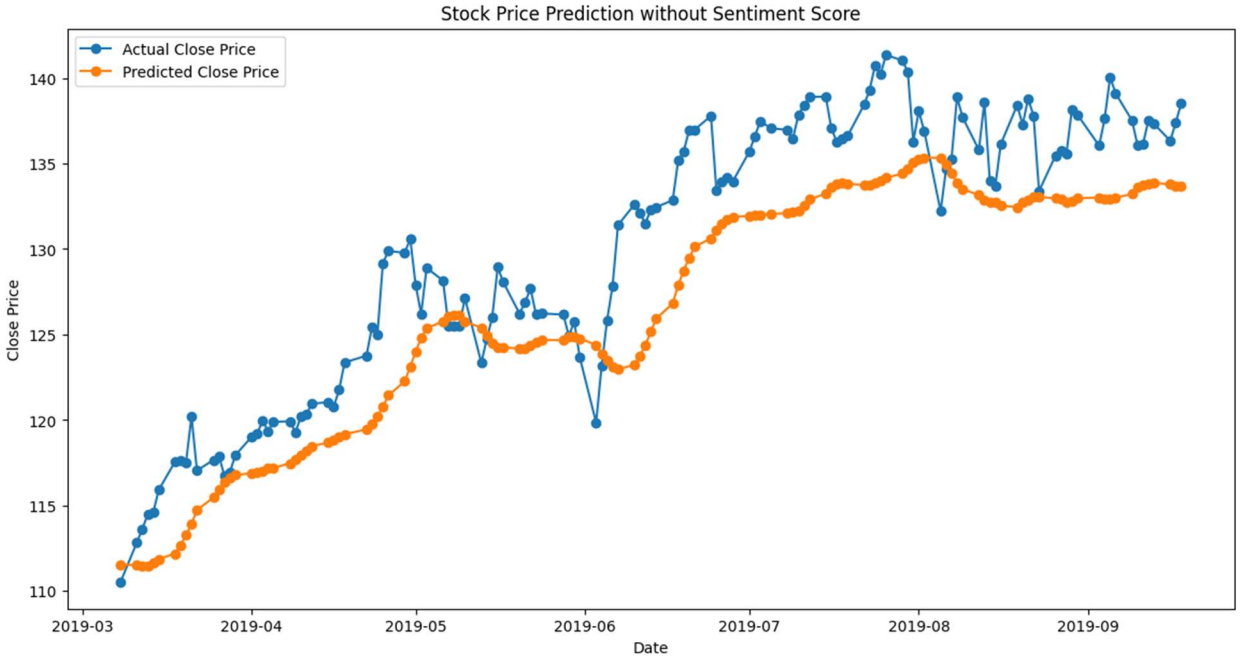
For MCD with base hyper tune parameters



Evaluation Metric	Value
Mean Squared Error (MSE)	11.0413
Root Mean Squared Error (RMSE)	3.3228
R-squared (R^2)	0.78

The above metrics collectively provide insights of closeness in predicting stock prices based on the features, including the sentimental score. A good model would ideally have low MSE and RMSE, indicating accurate predictions, and a high R-squared, suggesting that a significant portion of the variability in stock prices is captured by the model.

Prediction without Sentiment Score



Metric	Value
Mean Squared Error (MSE)	18.71296
Root Mean Squared Error (RMSE)	4.32585
R-squared (R^2)	0.704

5.3 Appendix C: Code Listing

Here is the link for the Code for the Proposed Stock Price Prediction Model

https://github.com/Shaik-36/stock-price-prediction-dissertation

University of Huddersfield

School of Computing and Engineering

PROJECT ETHICAL REVIEW FORM

Applicable for all research, masters and undergraduate projects

Project Title:	Optimizing Stock Price Prediction Using Long Short-Term Memory (Lstm) And Sentiment Analysis with Large Social Media Data
Student:	Imamuddin Shaik
Course/Programme:	MSc Computing
Department:	School of Computing and Engineering
Supervisor:	Professor Zhijie Xu
Project Start Date:	02/05/2023

ETHICAL REVIEW CHECKLIST

	Yes	No
1. Are there problems with any participant's right to remain anonymous?	☐	☒
2. Could a conflict of interest arise between a collaborating partner or funding source and the potential outcomes of the research, e.g. due to the need for confidentiality?	☐	☒
3. Will financial inducements be offered?	☐	☒
4. Will deception of participants be necessary during the research?	☐	☒
5. Does the research involve experimentation on any of the following?		

(i) animals?	☐	☒
(ii) animal tissues?	☐	☒
(iii)human tissues (including blood, fluid, skin, cell lines)?	☐	☒
6. Does the research involve participants who may be particularly vulnerable, e.g. children or adults with severe learning disabilities?	☐	☒
7. Could the research induce psychological stress or anxiety for the participants beyond that encountered in normal life?	☐	☒
8. Is it likely that the research will put any of the following at risk:		
(i) living creatures?	☐	☒
(ii) stakeholders (disregarding health and safety, which is covered by Q9)?	☐	☒
(iii)the environment?	☐	☒
(iv)the economy?	☐	☒
9. Having completed a health and safety risk assessment form and taken all reasonable practicable steps to minimise risk from the hazards identified, are the residual risks acceptable (Please attach a risk assessment form – available at the end of this document)	☒	☐

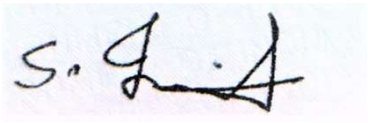
STATEMENT OF ETHICAL ISSUES AND ACTIONS

If the answer to any of the questions above is yes, or there are any other ethical issues that arise that are not covered by the checklist, then please give a summary of the ethical issues and the action that will be taken to address these in the box below. If you believe there to be no ethical issues, please enter "NONE".

NONE

STATEMENT BY THE STUDENT

I believe that the information I have given in this form on ethical issues is correct.

Signature:  Date: 05/01/2024

AFFIRMATION BY THE SUPERVISOR

I have read this Ethical Review Checklist, and I can confirm that, to the best of my understanding, the information presented by the student is correct and appropriate to allow an informed judgement on whether further ethical approval is required.

Signature: Zhijie Xu Date: 05/01/2024

SUPERVISOR RECOMMENDATION ON THE PROJECT'S ETHICAL STATUS

Having satisfied myself of the accuracy of the project ethical statement, I believe that the appropriate action is:

The project proceeds in its present form	X
The project proposal needs further assessment by an Ethical Review Panel. The Supervisor will pass the form to the Ethical Review Panel Leader for consideration.	

RETENTION OF THIS FORM

- The Supervisor must retain a copy of this form until the project report/dissertation is produced.
- The student must include a copy of the form as an appendix in the report/dissertation.

OUTCOME OF THE ETHICAL REVIEW PANEL PROCESS, WHERE REQUIRED

Tick One

1. Approved. The ethical issues have been adequately addressed and the project may commence. ☒

2. Approved subject to minor amendments. The required amendments are stated in the box below. The project may proceed once the form has been amended in line with the requirements and signed by the Supervisor in the box imediately below to confirm this. ☐

I confirm, as Supervisor, that the amendments required have been made:

Signature: Zhijie Xu Date: 05/01/2024

3. Resubmit. The areas requiring further action are stated in the box below. The project may not proceed until the form has been resubmitted and approved. ☐

4. Reject. The reasons why it will not be possible to address the ethical issues adequately are stated in the box below. ☐

For any of the outcomes 2, 3 or 4 above, please provide a statement in the box below.

AFFIRMATION BY THE REVIEW PANEL LEADER

I approve the decision reached above by the review panel members:

Signature: Zhijie Xu Date: 05/01/2024

THE UNIVERSITY OF HUDDERSFIELD: RISK ANALYSIS & MANAGEMENT

ACTIVITY:			Name:	
LOCATION:			Date:	Review Date:
Hazard(s) Identified	Details of Risk(s)	People at Risk	Risk management	Other comments