Predictive model for stock price movements using historical data and sentiment analysis

Phu Pham
pham84@purdue.edu
Purdue University
Indiana, USA

Ashwanth Kumar Gundeti
agundeti@purdue.edu
Purdue University
Indiana, USA

Raj Venkat Reddy Mavuram
rmavuram@purdue.edu
Purdue University
Indiana, USA

Felicia Grace Ravichandran
fravicha@purdue.edu
Purdue University
Indiana, USA

ABSTRACT

Stock price movements are influenced by a wide range of factors, including company performance, market trends, and global events. The ability to predict these movements is crucial for investors, traders, and financial analysts. In this project, we developed various machine learning models (Linear Regression, Random Forest, recurrent neural network) to evaluate their performances in predicting stock prices. We used two different sentiment analysis techniques, finBERT, and TextBlob, to extract sentiment scores from tweets related to the stock tickers in the same period. The proposed models leverage machine learning and deep learning techniques to identify patterns in historical and sentiment data, and make predictions about future stock prices. The results show that the Random Forest model outperformed the other two models for both sentiment analysis techniques. The findings of this work provide insights into the effectiveness of different machine learning models and sentiment analysis techniques for predicting stock prices, which can be useful for investors and financial analysts.

CCS CONCEPTS

• **Applied computing** → **Business** → **Finance and economics**;

KEYWORDS

Stock price prediction, Financial data, Sentiment analysis, Machine learning, Deep learning, Time-series analysis

1 INTRODUCTION

The stock market is one of the most important financial markets in the world. It plays a crucial role in the economic growth of the country and is a major source of investment for investors. Predicting the stock market is a challenging task due to its high volatility and complexity. It is a complex and dynamic system that is influenced by a wide range of factors, including macroeconomic trends, industry developments, and investor sentiment. In recent years, social media has emerged as a potential source of information that can be used to predict changes in investor sentiment and stock prices. Twitter, in particular, has become a popular platform for sharing opinions and news related to the stock market.

One can question why the sentiments of people have anything to do with predicting the pricing of the stock. Stock price prediction is crucial for investors as it helps them make informed investment decisions. By incorporating sentiment analysis of tweets into stock price prediction, we can gain valuable insights into investor sentiment, which can help us make more accurate predictions. Accurate predictions of stock prices can help investors make profits, while inaccurate predictions can result in significant losses.

Recently, the advent of machine learning algorithms has made it possible to develop predictive models that can identify patterns in historical data and use them to make predictions about future events. The goal of this project is to investigate the relationship between social media sentiment and stock prices by integrating two datasets: one containing historical stock prices for the Dow Jones Industrial Average and the other containing Twitter tweets about the top companies. We will use sentiment analysis to assign sentiment scores to each tweet and integrate this with the stock price dataset. Then, we will apply various machine learning algorithms to predict stock prices and evaluate their performance.

The motivation behind this project is to explore whether social media sentiment can be a useful tool for predicting changes in the stock market. If successful, this approach could provide investors with a new source of information that can be used to make more informed investment decisions. It could also help researchers better understand the relationship between social media sentiment and financial markets.

There have been several studies on the relationship between social media sentiment and stock prices. For example, some studies have found that Twitter sentiment can predict changes in stock prices with a high degree of accuracy. Other studies have found that social media sentiment can be a useful tool for predicting changes in financial markets and investor sentiment. By providing a detailed analysis of the relationship between social media sentiment and stock prices, our project will shed light on the potential benefits and limitations of this approach.

2 RELATED WORK

There has been a growing interest in using social media sentiment as a predictor of stock prices. Several studies have investigated the relationship between social media sentiment and stock prices, with varying degrees of success.

One study by Bollen et al.[2] found that Twitter sentiment can predict changes in the Dow Jones Industrial Average with an accuracy of up to 87.6%. The study used a set of sentiment analysis algorithms to analyze tweets about the stock market and found that changes in sentiment were strongly correlated with changes in the stock market. The methodology adopted can be seen in Figure 1.

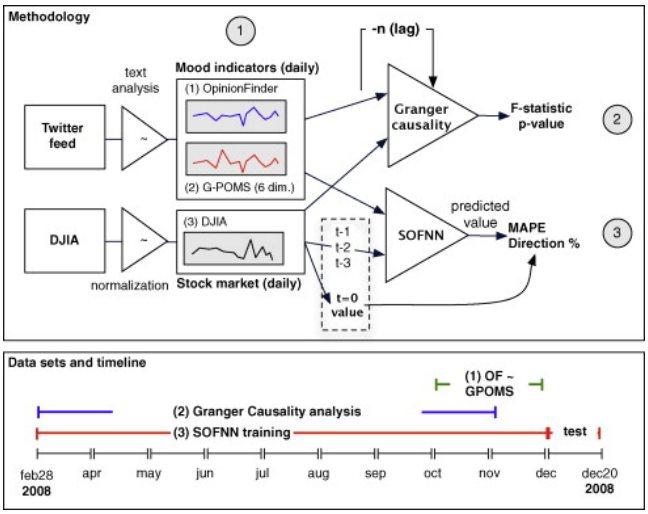


Figure 1: Procedure and Datasets timeline followed in Bollen et al.[2]

Another study by Tsyplakov and Alexandrov[10] used sentiment analysis to analyze tweets about specific companies and found that Twitter sentiment can be a useful predictor of changes in stock prices. The study used a machine learning algorithm to predict stock prices based on sentiment scores and found that the algorithm was able to outperform a random walk model.

However, there are also studies that have questioned the effectiveness of social media sentiment as a predictor of stock prices. For example, a study by Zhang et al.[12] found that sentiment analysis tools may be influenced by noise and that sentiment scores may not accurately reflect the true sentiment of the market.

Despite these limitations, there is still significant interest in using social media sentiment as a predictor of stock prices. The ability to quickly and accurately predict changes in the stock market could provide investors with a significant advantage and could help researchers better understand the complex relationship between social media sentiment and financial markets.

In this project, we aim to build on previous research by applying multiple machine learning algorithms to predict stock prices based on social media sentiment. By evaluating the performance of different algorithms, we hope to provide a more comprehensive analysis of the relationship between social media sentiment and stock prices.

3 DATA COLLECTION

3.1 Stock Price Dataset

The stock price data was collected from Yahoo Finance[1], a popular financial news and data platform. The dataset contains daily stock prices for the Dow Jones Industrial Average, a widely recognized index of 30 large, publicly traded companies in the United States. The data spans from January 2015 to June 2020 and includes the Open, Close, High, Low, Adjusted Close prices, and Volume for each trading day.

The data was collected using Python's pandas-datareader library, which provides a simple and convenient way to retrieve financial data from various sources. We used the library to download the historical stock price data for the Dow Jones Industrial Average from Yahoo Finance and save it in a pandas dataframe.

Date	Open	High	Low	Close	Adj Close	Volume
2015-01-02	27.847500	27.860001	26.837500	27.332500	24.565695	212818400
2015-01-05	27.072500	27.162500	26.352501	26.562500	23.873638	257142000
2015-01-06	26.635000	26.857500	26.157499	26.565001	23.875889	263188400
2015-01-07	26.799999	27.049999	26.674999	26.937500	24.210684	160423600
2015-01-08	27.307501	28.037500	27.174999	27.972500	25.140911	237458000

Figure 2: Stock price dataset

3.2 Twitter Dataset

The Twitter tweets dataset was collected from Kaggle[2], a well-known platform for data science competitions and datasets. The dataset contains over 3 million tweets related to the top companies in the United States, including Apple, Amazon, Google, Microsoft, and Tesla. The data spans from January 2015 to June 2020 and information such as tweet id, author of the tweet, post date, the text body of the tweet, and the number of comments, retweets, and likes.

The dataset was downloaded from Kaggle in CSV format and imported into a pandas dataframe using Python's pandas library. The dataframe was then cleaned and preprocessed to remove duplicate tweets, irrelevant columns, and convert the date to a standard datetime format.

4 DATA PREPROCESSING

The raw data obtained from the stock price and Twitter tweets datasets required preprocessing to prepare it for analysis. The preprocessing steps included cleaning the data, removing irrelevant columns, and converting the date columns to a standard datetime format.

The first step in preprocessing the data was to clean it by removing any missing or invalid values. We used Python's pandas library to check for missing values in the datasets and dropped any rows or columns containing missing data.

[1] https://finance.yahoo.com/quote/%5EDJI/history?period1

[2] https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020?select=Tweet.csv

tweet_id	writer	post_date	body	comment_num	retweet_num	like_num
550441509175443456	VisualStockRSRC	1420070457	lx21 made $10,008 on $AAPL -Check it out! http://profit.ly/1MnD8s?aff=202 Learn #howtotrade http://...	0	0	1
550441672312512512	KeralaGuy77	1420070496	Insanity of today weirdo massive selling. $aapl bid up 45 cents after hours after non stop selling i...	0	0	0
550441732014223360	DozenStocks	1420070510	S&P100 #Stocks Performance $HD $LOW $SBUX $TGT $DVN $IBM $AMZN $F $APA $GM $MS $HAL $DIS $MCD $BMY $...	0	0	0
550442977802207232	ShowDreamCar	1420070807	$GM $TSLA: Volkswagen Pushes 2014 Record Recall Tally Higher https://pic.twitter.com/WIIc1l	0	0	1

Figure 3: Twitter tweets dataset

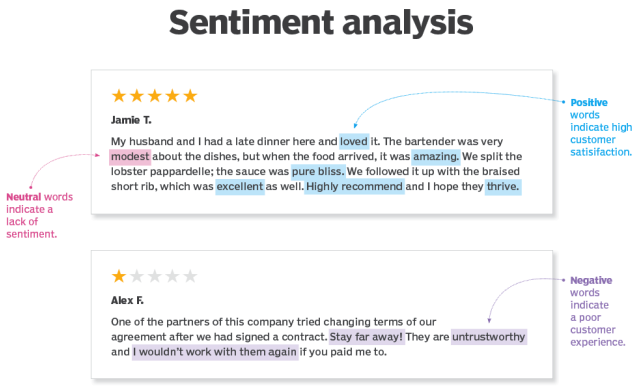


Figure 4: Example of Sentiment Analysis

We also removed columns that would not contribute to the stock price prediction from both the datasets. In the stock price dataset, we dropped the column "Adjusted Close" as we were only interested in predicting the "Close" price of the Dow Jones Industrial Average. In the tweets dataset, we dropped columns such as "Username" and "Tweet URL" as they were not relevant to the analysis.

Both datasets contained date columns that were not in a standard datetime format. We converted these date columns to a standard datetime format to ensure consistency across the datasets and to facilitate merging the datasets. We used Python's pandas library to convert the date columns to the datetime format.

After the data preprocessing steps were completed, we had a clean and consistent dataset ready for analysis. The integrated dataset contained the Dow Jones Industrial Average stock prices and sentiment scores for each trading day from January 2015 to June 2020. The dataset was then used to train and test various machine learning models for stock price prediction.

5 SENTIMENT ANALYSIS

Sentiment analysis, also known as opinion mining, is the process of using natural language processing (NLP) and machine learning techniques to identify and extract subjective information from text, such as opinions, attitudes, emotions, and sentiments. Sentiment analysis can be applied to various types of text data, such as product reviews, social media posts, news articles, customer feedback, and survey responses. The goal of sentiment analysis is to automatically classify the text as positive, negative, or neutral, or to quantify the intensity of the sentiment on a continuous scale. Sentiment analysis is a complex task that involves several subtasks, such as text preprocessing, feature extraction, model selection, and evaluation.

Figure 4 shows the example of sentiment analysis of sample reviews.[3]

Text Preprocessing: The first step in sentiment analysis is to preprocess the text data to remove noise and irrelevant information. This includes tasks such as tokenization, stemming, stopword removal, and spelling correction. Tokenization involves splitting the text into individual words or phrases, known as tokens. Stemming involves reducing the tokens to their base form or stem, such as converting "running" to "run". Stopword removal involves removing common words that do not contribute to the sentiment, such as "the", "and", "a", etc. Spelling correction involves identifying and correcting misspelled words.

Feature Extraction: The next step in sentiment analysis is to extract features from the preprocessed text data that can be used as input to a machine learning model. This includes tasks such as bag-of-words representation, n-grams, and word embeddings. Bag-of-words representation involves counting the frequency of each word in the text and representing it as a vector of numbers. N-grams involve considering sequences of n words instead of individual words. Word embeddings involve representing each word as a vector in a high-dimensional space based on its context and meaning.

Model Selection: The third step in sentiment analysis is to select a machine learning model that can learn to classify the text as positive, negative, or neutral. This includes tasks such as choosing the algorithm, tuning the hyperparameters, and training the model. Common machine learning algorithms used for sentiment analysis include Naive Bayes, Support Vector Machines (SVM), Decision Trees, and Neural Networks. Hyperparameters are parameters that need to be set before training the model, such as the learning rate, the regularization strength, and the number of hidden layers. Training the model involves using a labeled dataset of text data to learn the patterns and relationships between the features and the sentiment labels.

For analysing the tweets, used two different pretrained models for sentiment analysis of the tweets.

- FinBERT
- TextBlob

5.1 FinBERT

FinBERT is a pre-trained language model designed specifically for financial sentiment analysis, using the BERT (Bidirectional Encoder Representations from Transformers) architecture. Its main advantage over generic language models like BERT is its ability to capture the nuances and complexities of financial languages, such as technical jargon, abbreviations, and financial metrics. FinBERT is

[3]https://www.techtarget.com/searchbusinessanalytics/definition/opinion-mining-sentiment-mining

trained on a large corpus of financial news articles and other financial documents to learn the language patterns and domain-specific knowledge required for financial sentiment analysis. This makes it particularly useful for applications such as stock price prediction, financial news analysis, and sentiment analysis of social media data related to finance.

FinBERT is made up of several layers of deep neural networks that are trained to predict the next word in a sequence of words based on the context. The model is trained on a large dataset of financial news articles and other financial documents to learn the patterns and relationships between words and their meanings in the financial domain. This allows FinBERT to generate embeddings, which are vector representations of words that capture their semantic and syntactic properties, specifically tailored to the financial domain.

To use FinBERT for sentiment analysis, the model is fine-tuned on a labeled dataset of financial sentiment data, where each text sample is labeled as positive, negative, or neutral. The fine-tuning process involves adjusting the weights of the model to optimize the performance on the labeled dataset by minimizing a loss function that measures the difference between the predicted and actual sentiment labels. After fine-tuning, the FinBERT model can be used to predict the sentiment of new text samples related to finance, such as news articles, social media posts, and stock market reports. This allows investors, traders, and risk managers to make informed decisions and develop effective strategies.

Below are the scores produced by FinBERT model for the tweet "$AAPL Apple goes global with 'Start Something New' ad campaign http://bit.ly/1tof86M @MaximumPenny".

Sentence Positive Score	0.22936
Sentence Negative Score	0.01361
Sentence Neutral Score	0.75703

Table 1: Sentiment scores using finBERT

5.2 TextBlob

TextBlob is a pre-trained natural language processing (NLP) model that is built on top of Python and uses the NLTK library for text processing. It is designed to perform various language processing tasks, including sentiment analysis and text classification. The sentiment analysis algorithm used by TextBlob is simple and based on the polarity of words in the text. It uses a list of pre-defined positive and negative words and calculates the overall sentiment of the text by counting the number of positive and negative words and subtracting them. This approach makes TextBlob lightweight and easy to use, particularly useful for applications where speed and simplicity are important.

However, TextBlob also incorporates machine learning techniques to improve the accuracy of its sentiment analysis. It includes a Naive Bayes classifier that is trained on a large corpus of labeled data. The classifier learns the patterns and relationships between words and their sentiment labels and can be used to predict the sentiment of new text samples. This combination of a simple algorithm and a trained classifier makes TextBlob an effective tool for sentiment analysis of social media data, customer feedback, and other forms of text data. To use TextBlob, you simply pass in a text sample and call the sentiment method, which returns two values: the polarity and subjectivity of the text. The polarity value ranges from -1 (very negative) to 1 (very positive), with 0 indicating a neutral sentiment. The subjectivity value ranges from 0 (very objective) to 1 (very subjective), with 0.5 indicating a neutral subjectivity.

Below are the scores produced by Textblob model for the tweet "$AAPL Apple goes global with 'Start Something New' ad campaign http://bit.ly/1tof86M @MaximumPenny".

Sentence sentiment Score	0.06818

Table 2: Sentiment scores using TextBlob

6 STOCK PRICE PREDICTION

Machine learning algorithms can be used for stock price prediction by analyzing large amounts of historical data and identifying patterns and trends that can be used to make predictions about future stock prices. We will explore three such algorithms in this paper.

6.1 Regression Models

Regression models are a class of statistical models that are used to predict a continuous target variable based on one or more input variables. Linear regression is one type of regression model that assumes a linear relationship between the input variables and the target variable. In linear regression, the coefficients of the input variables are estimated to minimize the sum of squared differences between the predicted and actual values.

In the context of stock price prediction, the dependent variable is the stock price, while the independent variables are various economic and financial indicators that are believed to influence the stock price. The model assumes a linear relationship between the dependent and independent variables, and the goal is to find the best fit line that represents this relationship. The line is determined by minimizing the sum of the squared differences between the actual stock price and the predicted stock price.

The mathematical equation linear regression can be expressed as follows:

$$y' = \sum_{i=0}^{D} x_i . \beta_i \tag{1}$$

where each x_i is the i^{th} feature in input x, D is the dimension, or number of features in our dataset, and y' is the predicted future stock price. $\{\beta_i\}_{i=0}^{D}$ is the weight vector that is to be determined. The line (weight vector) is determined by minimizing the sum of the squared differences between the actual stock price and the predicted stock price. This is known as the cost function, and the process of finding the weight vector that minimizes this function is known as optimization. The most common optimization algorithm used in linear regression is gradient descent. Gradient descent iteratively updates the weight vector to find the minimum cost.

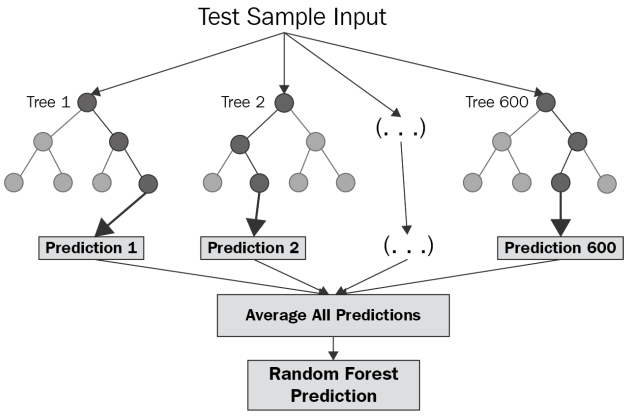


Figure 5: Random Forest: An ensemble of decision trees.[4]

One important consideration in linear regression is overfitting. Overfitting occurs when the model is too complex and fits the noise in the data instead of the underlying trend. This can be addressed by adding regularization terms to the cost function. The two most common types of regularization used in linear regression are L_1 regularization (Lasso) and L_2 regularization (Ridge).

6.2 Decision Trees

Decision trees are a type of machine learning model that is used for both classification and regression problems. The algorithm works by recursively partitioning the data into smaller and smaller subsets, based on the values of the input variables. At each node of the decision tree, the algorithm selects the input variable that best separates the data into different regions based on a chosen criterion. One commonly used criterion is the Gini index, which is calculated as:

$$Gini(X) = 1 - \sum_x p(x)^2 \tag{2}$$

Another commonly used criterion is entropy, which is calculated as:

$$H(X) = -\sum_x p(x) \log_2 p(x) \tag{3}$$

Random forests are an extension of decision trees. It is an ensemble model that combines the predictions of multiple decision trees. The prediction for a new input is the average (for regression) or the mode (for classification) of the predictions of the individual trees. Random forests are more powerful models than linear regression as they are capable of identifying and capturing non-linear relationship among the features.

6.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural networks that are designed to handle sequential data, such as time-series data. RNNs have a hidden state that is updated at each time step based on the input and the previous hidden state. This allows the network to capture temporal dependencies in the data. In simple words, output from the previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in case of time-series inputs, there is a requirement to keep track of previous inputs. A simple architecture of RNN is shown in Figure 2. Based on the Figure

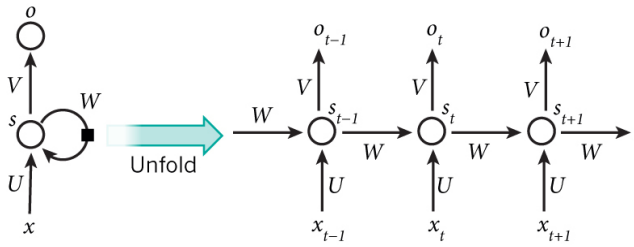


Figure 6: Architecture of RNN.[5]

2, the equations for calculating hidden state and outputs can be formulated as:

$$s_t = f(Ws_{t-1} + Ux_t) \tag{4}$$

$$o_t = g(Vs_t) \tag{5}$$

where f, g are activation functions. However, traditional RNNs suffer from the vanishing gradient problem, where the gradients used for updating the model weights become very small over time, making it difficult for the model to learn long-term dependencies. RNNs also suffer from exploding gradients problem, i.e, if the gradients are large, the multiplication of these gradients will become huge over time. This results in the model being unable to learn and its behavior becomes unstable.

Long Short-Term Memory (LSTM) networks are a type of RNN that address the vanishing gradient problem by introducing a memory cell and three gates: the input gate, output gate, and forget gate. LSTMs can be a real good choice if the data is influenced by long-term dependencies, such as trends or seasonal effects. They are widely used for various applications, including natural language processing, speech recognition, and stock price prediction. We can take a look at the architecture of one memory cell in LSTM in Figure 3.

Based on the Figure 3, the formulations for Input gate, Forget gate, Output gate, input node and memory cell internal state are given by:

$$\mathbf{I}_t = \sigma(X_t W_{XI} + H_{t-1} W_{HI} + b_I) \tag{6}$$

$$\mathbf{F}_t = \sigma(X_t W_{XF} + H_{t-1} W_{HF} + b_F) \tag{7}$$

$$\mathbf{O}_t = \sigma(X_t W_{XO} + H_{t-1} W_{HO} + b_O) \tag{8}$$

$$\tilde{\mathbf{C}}_t = \tanh(X_t W_{XC} + H_{t-1} W_{HC} + b_C) \tag{9}$$

$$\mathbf{C}_t = \mathbf{F}_t . \mathbf{C}_{t-1} + \mathbf{I}_t . \tilde{\mathbf{C}}_t \tag{10}$$

where $W_{X.}$ are weight vectors corresponding to input, $W_{H.}$ are weight vectors corresponding to hidden state, and $b_.$ are the bias parameters. The input gate $\mathbf{I}_t$ and forget gate $\mathbf{F}_t$ govern how much

[4]https://www.oreilly.com/api/v2/epubs/9781789132212/files/assets/1a024738-4913-4f17-9a5c-0fe116328393.png

[5]https://www.oreilly.com/api/v2/epubs/9781789132212/files/assets/1a024738-4913-4f17-9a5c-0fe116328393.png

of the current data $\tilde{\mathbf{C}}_t$ and past data $\mathbf{C}_{t-1}$ should be considered respectively. The hidden state $\mathbf{H}_t$ is computed as follows:

$$\mathbf{H}_t = \mathbf{O}_t . \tanh(\mathbf{C}_t) \tag{11}$$

The value of output gate, which lies in the range (0,1), decides how much the memory cell internal state impacts the subsequent layers. The LSTM network is trained using backpropagation through time (BPTT) algorithm, which is an extension of the backpropagation algorithm that is used for feedforward neural networks. In our experiment, we used Keras, a deep learning library in Python, to implement the LSTM network for stock price prediction.

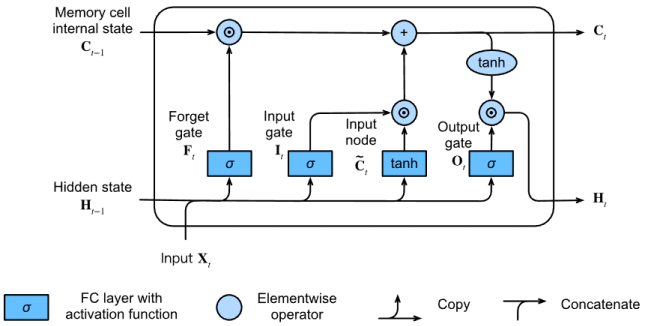


Figure 7: A memory cell unit in LSTM.[6]

7 INTEGRATION OF SENTIMENT ANALYSIS AND STOCK PRICE PREDICTION

We aim to accurately predict stock prices based on historical values and market sentiment. Thus, we need to incorporate this information to train our models. To this end, we collected all the tweets related to a stock ticker on a specific date. We computed the average sentiment scores of all the tweets using the finBERT [1] or TextBlob [6] model. Depending on the sentiment model, the output could be a vector of size three or just a number. The finBERT model returns a vector, indicating the probability of a tweet is positive, negative, or neutral. On the other hand, TextBlob output a number in the range (-1, 1) to indicate the polarity of the tweet. These aggregated sentiment scores of a stock ticker are later used as additional features in our training data. Figure 8 illustrates some examples of the sentiment scores output from the finBERT model.

Date	Tweet	Stock Name	Company Name	positive	negative	neutral
2022-09-29	Mainstream media has done an amazing job at br...	TSLA	Tesla, Inc.	0.176084	0.054749	0.769167
2022-09-29	Tesla delivery estimates are at around 364k fr...	TSLA	Tesla, Inc.	0.037382	0.022118	0.940500
2022-09-29	3/ Even if I include 63.0M unvested RSUs as of...	TSLA	Tesla, Inc.	0.048793	0.020222	0.930985
2022-09-29	@RealDanODowd @WholeMarsBlog @Tesla Hahaha why...	TSLA	Tesla, Inc.	0.051176	0.114965	0.833859
2022-09-29	@RealDanODowd @Tesla Stop trying to kill kids,...	TSLA	Tesla, Inc.	0.027116	0.689721	0.283164

Figure 8: Sentiment scores from finBERT model

For the historical data, we also applied some feature engineering methods. We first removed invalid or missing data. We also dropped the unnecessary data columns and retained only four columns: "Open", "Close", "Low", "High", and "Volume". Since these columns are not on the same scale, we standardized the data using Scikit-learn's MinMaxScaler. These features are scaled to the (0, 1) range. Figure 9 illustrates the standardized historical data.

Date	Open	High	Low	Close	Adj Close	Volume	Stock Name
2021-09-30	0.365894	0.365092	0.366737	0.363536	0.363744	0.172033	TSLA
2021-10-01	0.364622	0.361057	0.361106	0.363408	0.363616	0.163123	TSLA
2021-10-04	0.373477	0.373713	0.367289	0.366499	0.366706	0.292744	TSLA
2021-10-05	0.367753	0.369045	0.366342	0.366038	0.366245	0.176625	TSLA
2021-10-06	0.363545	0.363899	0.365858	0.367096	0.367303	0.140011	TSLA

Figure 9: Standardized historical data

Finally, we concatenated these features and sentiment features to create new training data that takes into account both the history and sentiment of the market.

After constructing the dataset, we split the dataset into training and testing sets. To predict the price of a stock on a specific date, we use the data of the same stock for the previous 20 days. We reserve one year of the latest data of a stock ticker as the test set and the rest as the training set.

In this project, we experimented with three different models: linear regression, random forest regressor, and recurrent neural network (LSTM). For the first two models, we use `LinearRegression` and `RandomForestRegressor` from the scikit-learn library. For the recurrent neural network, we construct an LSTM [4] model from `Keras` framework. The model comprises three LSTM layers of 64, 128, and 64 units, followed by a fully connected layer. The following code snippet illustrates how we construct the LSTM model in `Keras` [3].

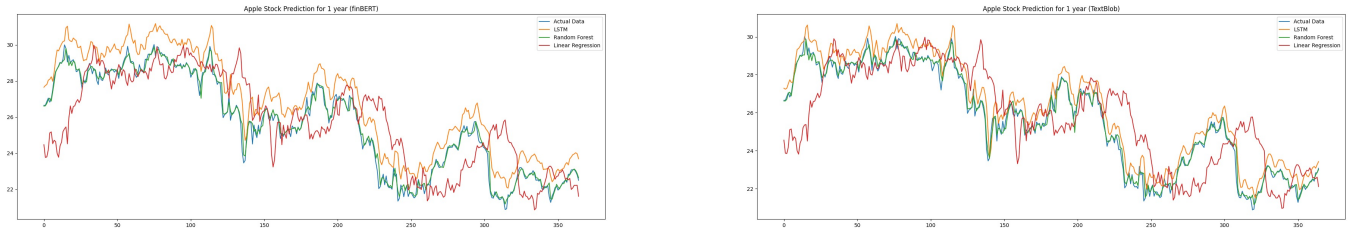
```
model = Sequential()
model.add(LSTM(units=64, return_sequences=True,
               input_shape=(20, 8)))
model.add(Dropout(0.2))
model.add(LSTM(units=128,
               return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=64))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

8 RESULTS

To compare the performance of our models, we use Root Mean Square Error (RMSE) as our evaluation metric.

RMSE (Root Mean Square Error) is a commonly used performance metric in regression tasks. It measures the difference between the actual and predicted values of the target variable. It is calculated by taking the square root of the mean of the squared differences between the actual and predicted values. RMSE is preferred over other error metrics like Mean Absolute Error (MAE) because it penalizes larger errors more heavily, making it more sensitive to outliers. The lower the RMSE value, the better the performance of the regression model. RMSE is often used to compare the performance of different regression models and to tune the hyperparameters of the models to achieve better results.

[6] https://www.oreilly.com/api/v2/epubs/9781789132212/files/assets/1a024738-4913-4f17-9a5c-0fe116328393.png



(a) finBERT **(b) TextBlob**

Figure 10: Apple's stock price predictions using different sentiment analysis models

We report the results for our experimental models in Table 3. The table presents the quantitative results of three machine learning models, Linear Regression, Random Forest, and LSTM, using two different sentiment analysis techniques, finBERT and TextBlob.

	finBERT	TextBlob
Linear Regression	0.0321	0.0265
Random Forest	0.0177	0.0178
LSTM	0.0237	0.0186

Table 3: Quantitative results

Based on Table 3, the Random Forest model is the most effective model in predicting stock prices in both sentiment analysis techniques. In all cases, the Random Forest model had the lowest RMSE values (0.0177 and 0.0178), indicating better performance. The LSTM model is the second best among the three, with RMSE values are 0.0237 and 0.0186, respectively. Linear regression performs worst, regardless of the sentiment analysis technique used. It is also worth noting that finBERT sentiment analysis generally yielded higher RMSE values compared to TextBlob, which suggests that TextBlob may be more effective in extracting sentiment from the tweets related to the stock tickers used in this experiment. Further analysis may be necessary to investigate the reason for this difference in performance between the two sentiment analysis techniques. Overall, these results provide valuable insights into the effectiveness of different machine learning models and sentiment analysis techniques for predicting stock prices.

To visualize the performance of our models, we pick Apple's stock (APPL as ticker). We predict Apple's stock prices for one year using 20 days of previous data. The prediction results using finBERT and TextBlob are plotted in Figures 10a and 10b.

Similar to the quantitative results, the Random Forest performs the best, followed by LSTM and Linear Regression models. The Random Forest model is very close to the actual data, indicating its ability to precisely capture the pattern of the test data. The LSTM model often predicts the stock price to be slightly higher than the actual price. On the contrary, the Linear Regression model cannot capture the complexity of the data, its prediction is pretty far off from the ground truth.

9 CONCLUSION

This project involved understanding the relationship between social media sentiment and stock prices by integrating two datasets: one containing historical stock prices for the Dow Jones Industrial Average and the other containing Twitter tweets about the top companies from 2015 to 2020. We used sentiment analysis to assign sentiment scores to each tweet and integrated this with the stock price dataset. Then, we applied three machine learning algorithms namely Long Short Term Memory (LSTM), Random Forest Regressor and Linear Regression with two different sentiment analysis models finBERT and TextBlob to predict stock prices and evaluate their performance.

Our results suggest that there is a correlation between social media sentiment and stock prices. Using finBERT sentiment analyzer we got the respective Root Mean Square Error values of 0.0237, 0.0177, 0.0321 for LSTM, Random Forest Regressor and Linear Regressor. On the other hand, we got the values 0.0186, 0.0178, 0.0265 for LSTM, Random Forest Regressor and Linear Regressor by using TextBlob for sentiment analysis. We observed that Random Forest Regressor had the best performance of the three models we used while Linear Regression had the worst performance which is expected as it cannot be used directly for time series prediction because it assumes that the observations are independent of each other, which is not the case in a time series modeling such as stock prices.

These findings are in line with previous research on the relationship between social media sentiment and stock prices. For example, some studies have found that Twitter sentiment can predict changes in stock prices with a high degree of accuracy. Other studies have found that social media sentiment can be a useful tool for predicting changes in financial markets and investor sentiment.

However, it is important to note that our analysis only covers a specific time period and may not generalize to other time periods or other datasets. In addition, our analysis is limited by the fact that we only used two sentiment analysis models (finBERT and TextBlob) and three models to predict (LSTM, Random Forest Regressor, and Linear Regression). Further research is needed to confirm these findings and explore other tools and algorithms that may be useful for predicting stock prices based on social media sentiment.

We have also predicted the stock prices using models such as Gated Recurrent Units (GRU), and Auto Regressive Integrated Moving Average (ARIMA). However, the results of the predictions using

these models was not quite accurate as intened and hence we decided not include them.

Overall, our project provides a useful starting point for investors and researchers who are interested in using social media sentiment as a predictor of stock prices. By providing a detailed analysis of the relationship between social media sentiment and stock prices, our project sheds light on the potential benefits and limitations of this approach.

10 CONTRIBUTIONS

Phu: Researched the datasets and implemented the overall pipleline, including data processing, feature engineering, model selection. Phu implemented two machine learning models: recurrent neural network (RNN) and deep neural network (DNN), only the RNN model was used in our evaluation since DNN model had similar performance.

Raj: Researched the datasets across various websites such as Kaggle and Yahoo Finance. Used Afinn for the sentiment analyis on the tweets dataset but decided to move forward with TextBlob and finBERT in favor of better results. Worked on the implementation of two models using the integrated dataset : Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU). Dropped the GRU model from our evaluation due to less accurate predictions compared to the other models.

Ashwanth: Pre-processed the data to compute sentiment analysis scores using FinBERT and TextBlob models on a dataset of over 42,00,000 tweets for five companies over five years. Used these two datasets to implement, train and evaluate Linear Regression, Random Forests, and Long Short Term Memory (LSTM) models for stock price prediction, and visualized the results.

Felicia: Researched for dataset in various datasources. Analysed different sentiment models. chose FinBERT and Textblob. Worked on pre-processing, LSTM and ARIMA models but ARIMA did not performed well so dropped as part of evaluation.

REFERENCES

[1] Dogu Araci. 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. *CoRR* abs/1908.10063 (2019). arXiv:1908.10063 http://arxiv.org/abs/1908.10063

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

[3] François Chollet et al. 2015. Keras. https://keras.io.

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[5] M. Omair Shafq Jingyi Shen. 2020. *Short-term stock market price trend prediction using a comprehensive deep learning system.*

[6] Steven Loria. 2018. textblob Documentation. *Release 0.15* 2 (2018).

[7] Himank Sharma Milind Manjrekar Nutan Hindlekar Pranali Bhagat Usha Aiyer Yogesh Agarwal Narayana Darapaneni, Anwesh Reddy Paduri. 2022. *Stock Price Prediction using Sentiment Analysis and Deep Learning for Indian Markets.*

[8] Bhavya K Padmanayana, Varsha. 2021. *Stock Market Prediction Using Twitter Sentiment Analysis.*

[9] Jajati Keshari Sahoo Pushpendu Ghosh, Ariel Neufeld. 2021. *Forecasting directional movements of stock prices for intraday trading using LSTM and random forests.*

[10] Alexandrov D Tsyplakov A. 2017. *Can Twitter help predict firm-level stock returns?. Financial Analysts Journal, 73(4), 108-123.*

[11] Jingyang Wang Lele Qin Wenjie Lu, Jiazheng Li. 2020. *A CNN-BiLSTM-AM method for stock price prediction.*

[12] Mao Y. Li Y Zhang, Y. 2018. *Examining the quality of online news and its relationship with market sentiment. Information Processing Management, 54(1), 1-12.*