Stock Price Prediction with Sentiment Analysis

Amey Shankar Basangoudar, Nidhi Vasantbhai Bodar, Saachi Chandrashekhar, Soham Shinde (Team 7)
GitHub: https://github.com/sohamthirty/Stock-Price-Prediction-with-Sentiment-Analysis

1. SUMMARY

The volatility of the stock market presents a substantial challenge for investors, particularly when it comes to timing the sale of shares amidst market fluctuations. Although accurate predictions are hard to come by, analyzing historical trends and employing machine learning can help mitigate losses. This project seeks to explore the relationship between the sentiments expressed on social media and stock price variations by integrating sentiment analysis of online discussions with historical data from Yahoo Finance. The goal is to assess the influence of public sentiment on stock prices, acknowledging the significant impact that news, events, and the digital conversation can have on market behavior.

Hence, the core objective of our project is to conduct a comprehensive analysis and scrutiny of stock prices. We place a particular emphasis on discerning underlying patterns and implementing time series evaluations on the stock prices data. This involves using time-series models such as ARIMA and LSTM. Our aim is to build a robust model that integrates sentiment scores of social media posts using VADER scores and news headlines using FINBERT with stock prices. The focus is on predicting future stock price movements. Additionally, we aim to conduct a comparative study between the movement of stocks with and without social media sentiments.

2. METHODS

2.1 Data Sources

Table 1. gives a brief description of all the datasets used in this project.

Dataset	**Source**	**Size (Rows)**	**Features**	**Purpose / Note**
Stock Data	Yahoo Finance API [5]	1500	Close, Open, High, Low, Adjusted Close	Primary Data for Forecasting. Scrapped for 2015-2019.
Twitter Data	Kaggle [3]	4.3M	Stock Ticker, Date, Tweet, Author	For obtaining sentiments.
News Data	Kaggle [9]	1.4M	Stock Ticker, Date, News Title	For obtaining sentiments.
Reddit Data	PRAW API [8]	1581	Stock Ticker, Date, Title, Content, Author, Score	For obtaining sentiments. Scrapped subreddits like 'stocks,' 'investing,' 'wallstreetbets,' etc.

Table 1: Data sources

2.2 Data Preprocessing

In the Data Preprocessing phase, a comprehensive check was conducted on all datasets to identify any null or missing values. Fortunately, no instances of missing data were found. Subsequently, the datasets were transformed into a structured and tidy format, setting the stage for in-depth Exploratory Data Analysis (EDA).

To enhance the text data for analysis, a custom preprocessing function was applied. This function incorporated essential techniques such as lemmatization, stemming, removal of punctuation, hashtags, stop words, and hyperlinks. Additionally, all text was converted to lowercase while dealing with the FinBERT model to maintain uniformity across the tokens.

Furthermore, the date information in the datasets was processed by converting it into a standardized datetime format. In this project, our analysis is centered around Google (GOOGL) stock due to the consistent and reliable data available for this particular stock across all datasets.

2.3 Model Description

2.3.1 VADER Scores

VADER (Valence Aware Dictionary for sEntiment Reasoning) is a model used for text sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It operates as a lexicon and rule-based sentiment analysis tool designed to be especially attuned to sentiments expressed in web-based content. It relies on a dictionary to map lexical features to emotion intensities, yielding sentiment scores for each word within a sentence. These scores, ranging from -4 to 4 for individual words, are normalized to fall within the range of -1 (most negative) to 1 (most positive) for the sentence as a whole.

Our analysis centered on the 'compound_score' derived from VADER polarity scores across all three datasets. While we adjusted threshold values as per the dataset's text characteristics, we maintained a consistent sentiment range across them: positive sentiment (0 to 1), neutral sentiment (0), and negative sentiment (-1 to 0).

2.3.2 FinBERT

We have leveraged the FinBERT, a specialized BERT model pre-trained for financial applications as our second choice. FinBERT has been trained on a vast corpus of financial communication text, totaling 4.9 billion tokens. This corpus comprises three key sources:

1. Corporate Reports 10-K & 10-Q: 2.5 billion tokens
2. Earnings Call Transcripts: 1.3 billion tokens
3. Analyst Reports: 1.1 billion tokens

For our specific sentiment analysis task, we have employed the finbert-tone model, which is a fine-tuned version of FinBERT. This fine-tuning process involves training FinBERT on 10,000 manually annotated sentences sourced from analyst reports. These annotations cover a spectrum of sentiments, including positive, negative, and neutral tones.

2.3.3 Prophet Model

Prophet is an open-source library developed by Facebook designed for making forecasts for univariate time series datasets. We implemented this as it is easy to use and designed to automatically find a good set of hyperparameters for the model in an effort to make skillful forecasts for data with trends and seasonal structure by default. Prophet is robust to outliers, missing data, and dramatic changes in time series and uses a combination of regression models and Bayesian inference to model time series data.

2.3.4 ARIMA and VARIMA

ARIMA is widely utilized in stock price forecasting for its robust ability to model both trends and seasonality. It excels when stock price patterns are pronounced, delivering forecasts that leverage historical data and market tendencies. ARIMA's three elements—Auto Regression (AR), Moving Average (MA), and Integration (I)—form a comprehensive approach to address the non-stationarity often present in stock data. AR uses past stock prices for predictions, MA uses past forecast errors, and I transforms non-stationary data to a stationary form, crucial for consistent forecasting.

To confirm data stationarity, the Augmented Dickey-Fuller (ADF) test is employed. It identifies if a series is stationary, ensuring the ARIMA model's prerequisites are met. If the data is non-stationary, differencing is applied to stabilize it. The ARIMA model is then constructed with specific lags for AR and MA, and the degree of differencing, denoted by the parameters p, d, and q respectively.

However, ARIMA's univariate nature can lead to less accurate predictions since it overlooks other influential variables. This limitation is addressed by the VARIMA model, a multivariate extension that considers various related data points like open, high, and low stock prices. VARIMA enhances the predictive prowess by applying ARIMA's principles to a broader dataset.

2.3.5 LSTM

LSTM networks are also explored due to their capacity to adapt to market dynamics. As a type of Recurrent Neural Network (RNN), LSTM employs memory cells to capture long-range dependencies, allowing controlled information flow via gates. It leverages univariate and multivariate lag values to make predictions based on historical and related data. The LSTM model takes historical stock data, including features like past prices and trading volume, as input and processes it through the network architecture to predict future stock prices, specifically the close price. It remembers past data and understands temporal patterns, making it well-suited for time series analysis. Hyperparameter tuning is conducted by experimenting with different numbers of hidden layers in the LSTM architecture, leveraging the network's adaptive capabilities.

2.4 Evaluation Metrics

1. RMSE (Root Mean Square Error) measures the square root of the average of the squared differences between predicted and actual values. It tends to penalize larger errors more than smaller ones due to the squaring operation.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

2. MAPE (Mean Absolute Percentage Error) calculates the mean of the absolute percentage differences between predicted and actual values. It expresses the error as a percentage of the actual values, making it easier to interpret in terms of accuracy.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|$$

In the context of stock price forecasting, both RMSE and MAPE can be useful. We consider both metrics to gain a comprehensive understanding of a model's performance. RMSE can help identify the overall magnitude of forecasting errors, while MAPE can provide insights into the relative accuracy, especially when comparing the forecasting accuracy of different stocks or when considering the impact on investment decisions.

3. RESULTS

3.1 Exploratory Data Analysis

After data preprocessing, we initiated the exploratory data analysis to enhance our comprehension of the data. We conducted a review to determine the quantity of data points present per dataset, per stock, and per year. This was crucial in order to make well-informed decisions regarding the appropriate data subset to be utilized for subsequent modeling efforts.

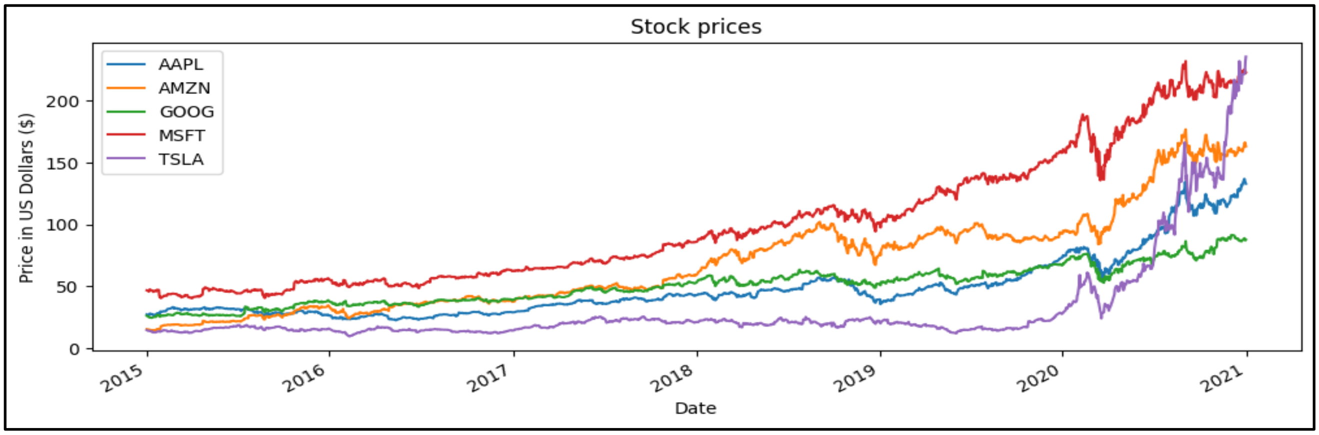
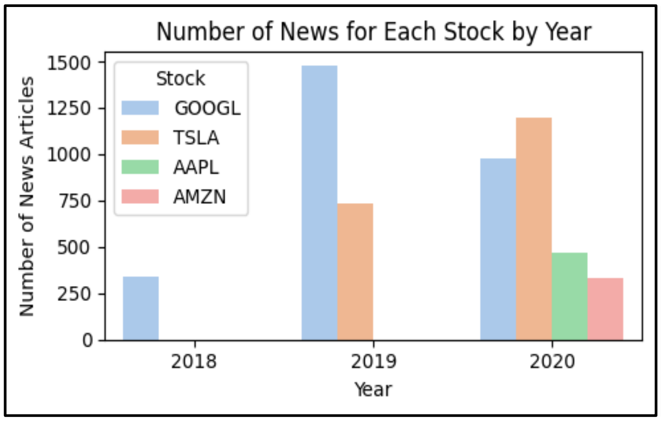


Fig.1 : Stock price trend lines

The stock close price trend was plotted for different stocks through 2015 to 2021 shown in Fig.1.



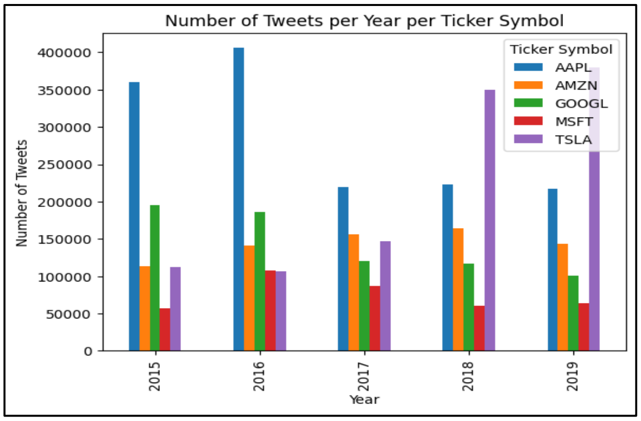


Fig.2 : Number of News Articles

Fig.3 : Number of Tweets

Observing Fig. 2 and Fig. 3, it becomes evident that a majority of news articles are associated with an abundance of data related to Google, while the Tweets predominantly pertain to Apple and Tesla stocks.

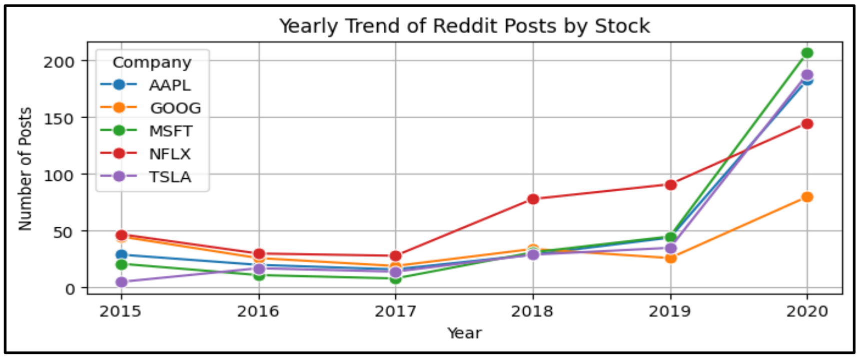


Fig.4: Yearly Trend of Reddit Posts

Fig. 4 shows us the yearly trend for the Reddit data by representing the stocks by different colors. We can observe a significant increase in no. of records in the later years for all the stocks.

3.2 Sentiment Analysis:

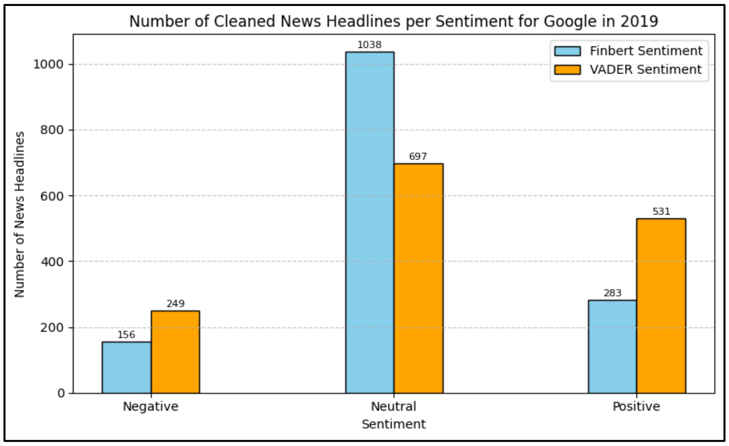


Fig.5: VADER and FinBert scores on News Data

Fig. 5 shows the comparison between VADER and FinBert for news data on Google stock. It seems that the news dataset consists of more neutral news as both models predict a higher number of sentiments for the neutral category. Although, FinBert classifies more news as neutral than VADER.

Similarly, we have used the VADER model on twitter and reddit datasets to obtain sentiments.

3.3 ARIMA and VARIMA for Close Prices:

To identify the best hyperparameters for the ARIMA model, the ACF (Autocorrelation function) and PACF (Partial Autocorrelation Function) plots are used. An ACF measures and plots the average correlation between data points in time series and previous values of the series measured for different lag lengths. A PACF is similar to an ACF except that each partial correlation controls for any correlation between observations of a shorter lag length. The best hyperparameters are chosen by looking at the plots in Fig. 6 and counting the number of times differencing is done. The lags with the highest correlation with the Close price are chosen as hyperparameters. The final hyperparameters are chosen as (p, d, q) = (1, 1, 1). The same hyperparameters are used for the VARIMA model as well.

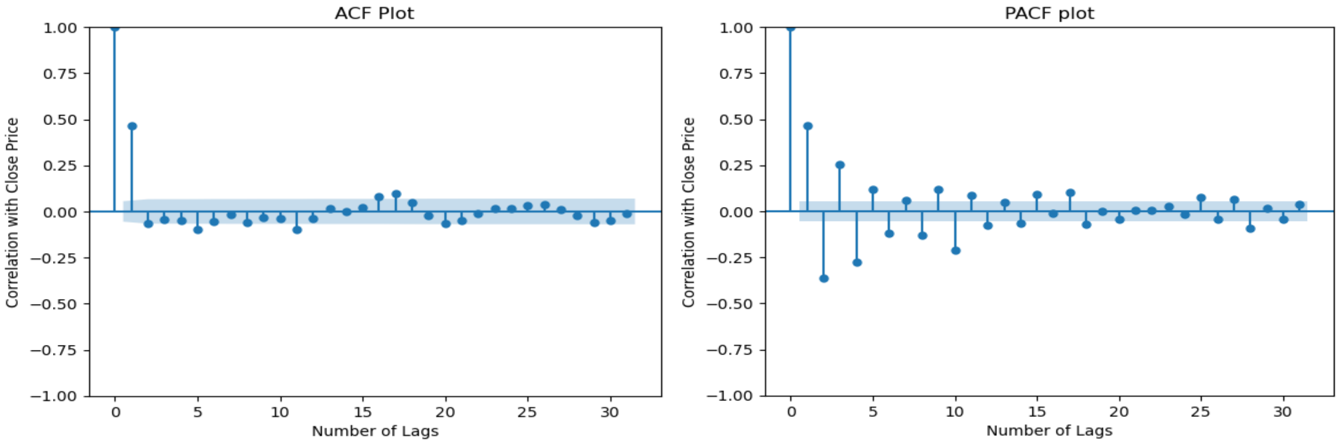


Fig.6: ACF and PACF plots to identify best hyperparameters

3.4 Stock Price Prediction:

The data was splitted into train, validation & test sets for predicting the Google stock prices across different time frames. This has enabled us to evaluate performance along with validation data to prevent overfitting.

Fig. 7 and Fig. 8 show the comparison between Actual and Predicted points from the different models.

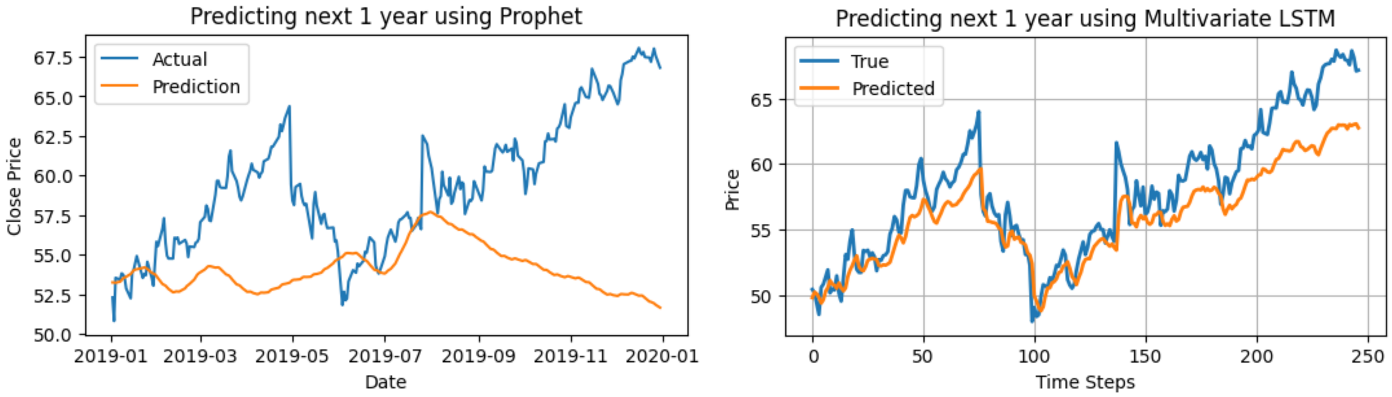


Fig.7: Forecasted stock prices using Prophet and Multivariate LSTM.

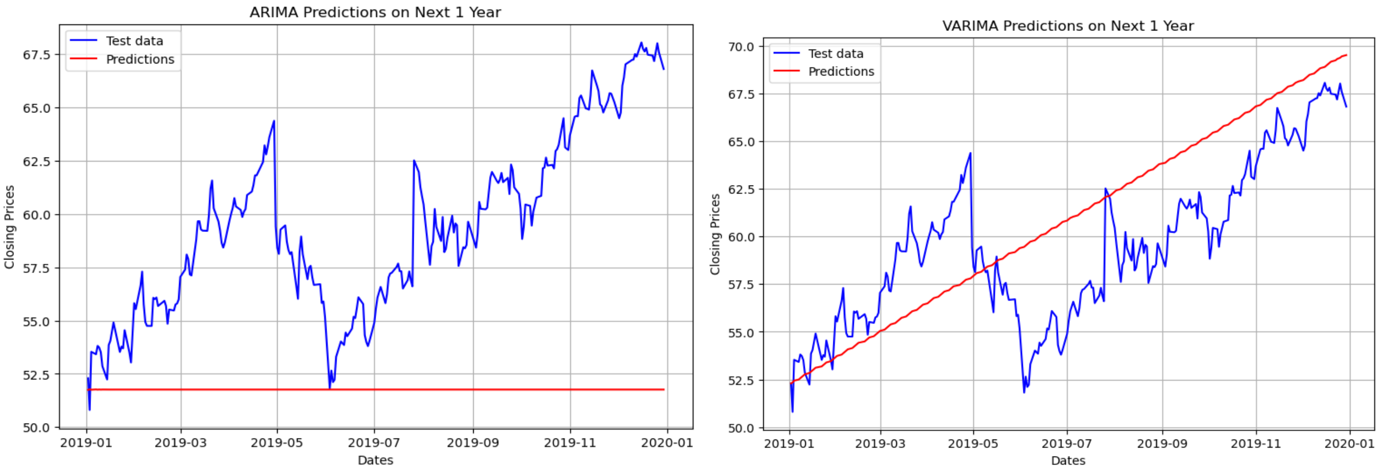


Fig.8: Forecasted stock prices using ARIMA and VARIMA.

From Fig. 9 we see how each model performed using the error metrics RMSE and MAPE for the cases of predicting next 1 year, 1 month and 10 days. All models exhibit improved performance when predicting shorter time frames. However, it's noteworthy that the multivariate LSTM model stands out as it demonstrates strong predictive capabilities even when forecasting one year ahead.

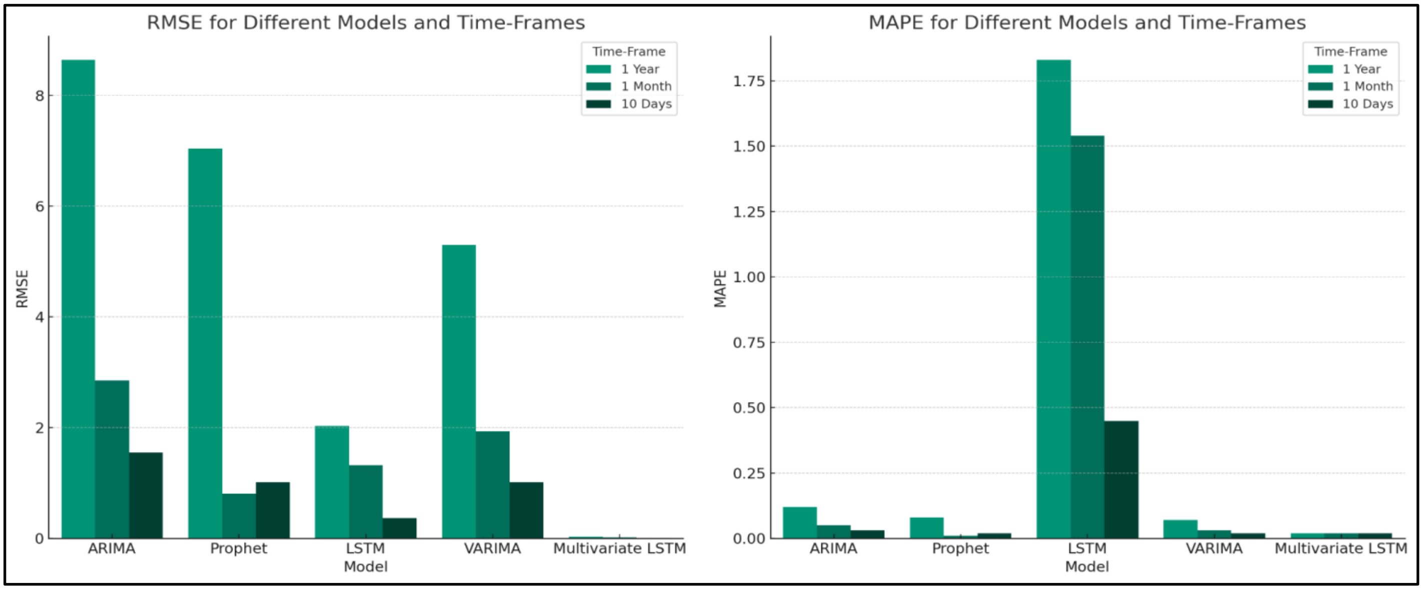


Fig.9: *Model Performance on Google Stocks*.

4. DISCUSSION

We have observed that multivariate analysis outperforms univariate analysis when analyzing the modeling results. Our models have also demonstrated a higher accuracy in predicting smaller and more recent time frames which makes them particularly suitable for short-term investments due to their sensitivity to daily decisions. Just as real-world stock prices are impacted by current events, our analysis shows that our prices are influenced by these short-term happenings, making them a trustworthy indicator of upward and downward trends in the market. Hence all models perform well for predictions made in a short time frame. However, the multivariate LSTM model, that takes into account long term and short term data, works well for predictions made over long periods of time.

Moreover, we encountered challenges while attempting to scrape Twitter data, as Elon Musk's Twitter handle changed to 'X,' making it difficult to access the Twitter API. Consequently, we opted for a Kaggle Twitter dataset for sentiment analysis tasks.

In Phase 2 of our project, we will extend the insights gained during Phase 1 by incorporating sentiment analysis into our stock price prediction system. This will enable us to determine if the sentiments have any influence on the stock prices, offering a more holistic view of market dynamics. Additionally, we will conduct a thorough feature importance analysis using packages such as SHAP to identify the most influential factors in both stock price and sentiment data. To make our predictions and insights accessible, we will deploy our models and develop a user-friendly interface featuring interactive dashboards, providing real-time stock price forecasts, sentiment analysis, and historical data visualization.

5. STATEMENT OF CONTRIBUTIONS

Amey Shankar Basangoudar	Stock Data Scraping + EDA (Time series), ARIMA,VARIMA predictions
Nidhi Vasantbhai Bodar	News Data EDA, Sentiment Analysis, FinBERT Scores
Saachi Chandrashekhar	Twitter Data EDA, Sentiment Analysis, VADER scores, Prophet predictions
Soham Shinde	Reddit Data Scraping, EDA, Sentiment Analysis, LSTM prediction

6. REFERENCES

[1] Bollen, J., &amp; Mao, H. (2011). Twitter mood as a stock market predictor. Computer, 44(10), 91–94. https://doi.org/10.1109/mc.2011.323

[2] Sharma, V., Khemnar, R., Kumari, R., & Mohan, B. R. (2019). Time series with sentiment analysis for stock price prediction. 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). https://doi.org/10.1109/icct46177.2019.8969060

[3] Metin, & Dogan. (2020, November 26). Tweets about the top companies from 2015 to 2020. Kaggle. https://www.kaggle.com/datasets/omermetinn/tweets-about-the-top-companies-from-2015-to-2020/data

[4] Briggs, J. (2021, September 2). Sentiment analysis for stock price prediction in Python. Medium. https://towardsdatascience.com/sentiment-analysis-for-stock-price-prediction-in-python-bed40c65d178

[5] Yahoo! (n.d.). Yahoo Finance - Stock Market Live, quotes, Business & Finance News. Yahoo! Finance. https://finance.yahoo.com/

[6] Yahoo Finance. pandas. (n.d.). https://pandas-datareader.readthedocs.io/en/latest/readers/yahoo.html

[7] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014. https://github.com/cjhutto/vaderSentiment

[8] PRAW: The Python Reddit API Wrapper#. https://praw.readthedocs.io/en/stable/

[9] Daily Financial News for 6000+ Stocks https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests

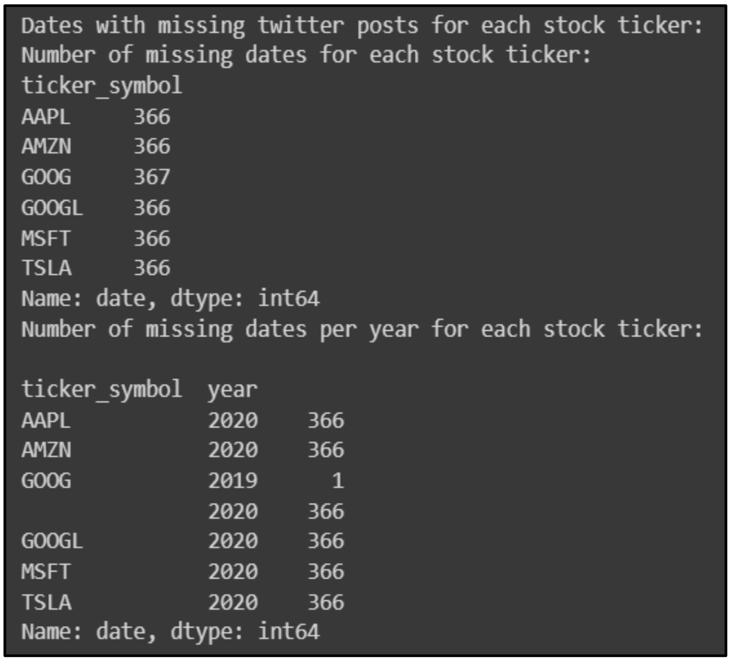
[10] Huang, Allen H., Hui Wang, and Yi Yang. "FinBERT: A Large Language Model for Extracting Information from Financial Text." Contemporary Accounting Research (2022).

7. APPENDIX

- Script for Scraping Yahoo Finance Stocks: Stock_Price_Scraping_EDA.ipynb

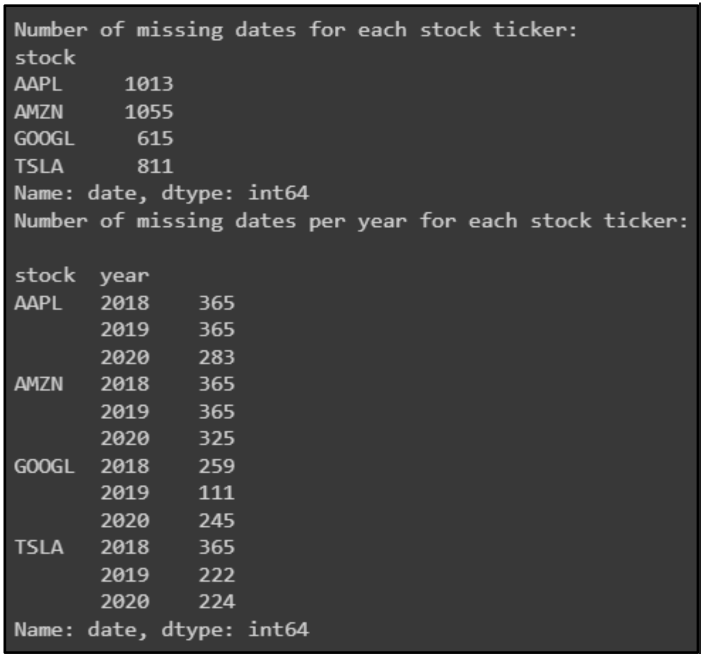
- Script for Scraping Reddit Posts: Scrape_RedditPosts.ipynb

- Number of missing Twitter posts, News Headlines and Reddit posts



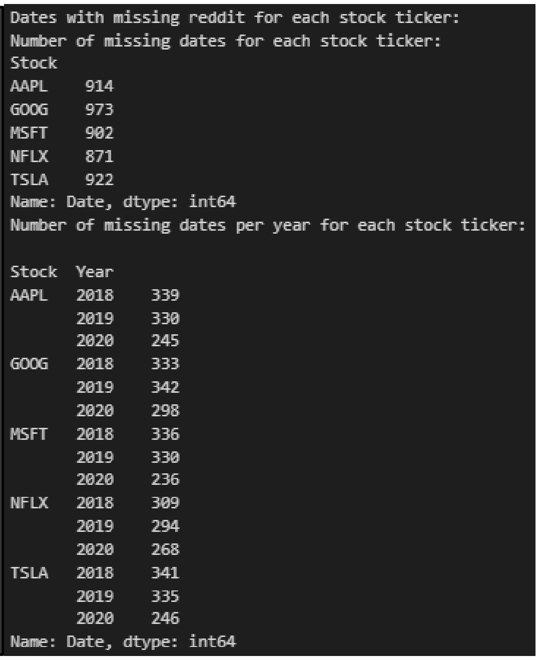
```
Dates with missing twitter posts for each stock ticker:
Number of missing dates for each stock ticker:
ticker_symbol
AAPL     366
AMZN     366
GOOG     367
GOOGL    366
MSFT     366
TSLA     366
Name: date, dtype: int64
Number of missing dates per year for each stock ticker:

ticker_symbol  year
AAPL           2020    366
AMZN           2020    366
GOOG           2019      1
               2020    366
GOOGL          2020    366
MSFT           2020    366
TSLA           2020    366
Name: date, dtype: int64
```



```
Number of missing dates for each stock ticker:
stock
AAPL     1013
AMZN     1055
GOOGL     615
TSLA      811
Name: date, dtype: int64
Number of missing dates per year for each stock ticker:

stock  year
AAPL   2018    365
       2019    365
       2020    283
AMZN   2018    365
       2019    365
       2020    325
GOOGL  2018    259
       2019    111
       2020    245
TSLA   2018    365
       2019    222
       2020    224
Name: date, dtype: int64
```



```
Dates with missing reddit for each stock ticker:
Number of missing dates for each stock ticker:
Stock
AAPL    914
GOOG    973
MSFT    902
NFLX    871
TSLA    922
Name: Date, dtype: int64
Number of missing dates per year for each stock ticker:

Stock  Year
AAPL   2018    339
       2019    330
       2020    245
GOOG   2018    333
       2019    342
       2020    298
MSFT   2018    336
       2019    330
       2020    236
NFLX   2018    309
       2019    294
       2020    268
TSLA   2018    341
       2019    335
       2020    246
Name: Date, dtype: int64
```

Fig 1. Number of missing Twitter posts, News Headlines and Reddit posts

- Model Diagnostics for the ARIMA model:

To check if the assumptions of the model are not violated, the diagnostic plots are used. We clearly see that error terms are normally distributed and have constant variance (homoscedastic). Hence, no assumptions are violated and the results generated are not by chance.

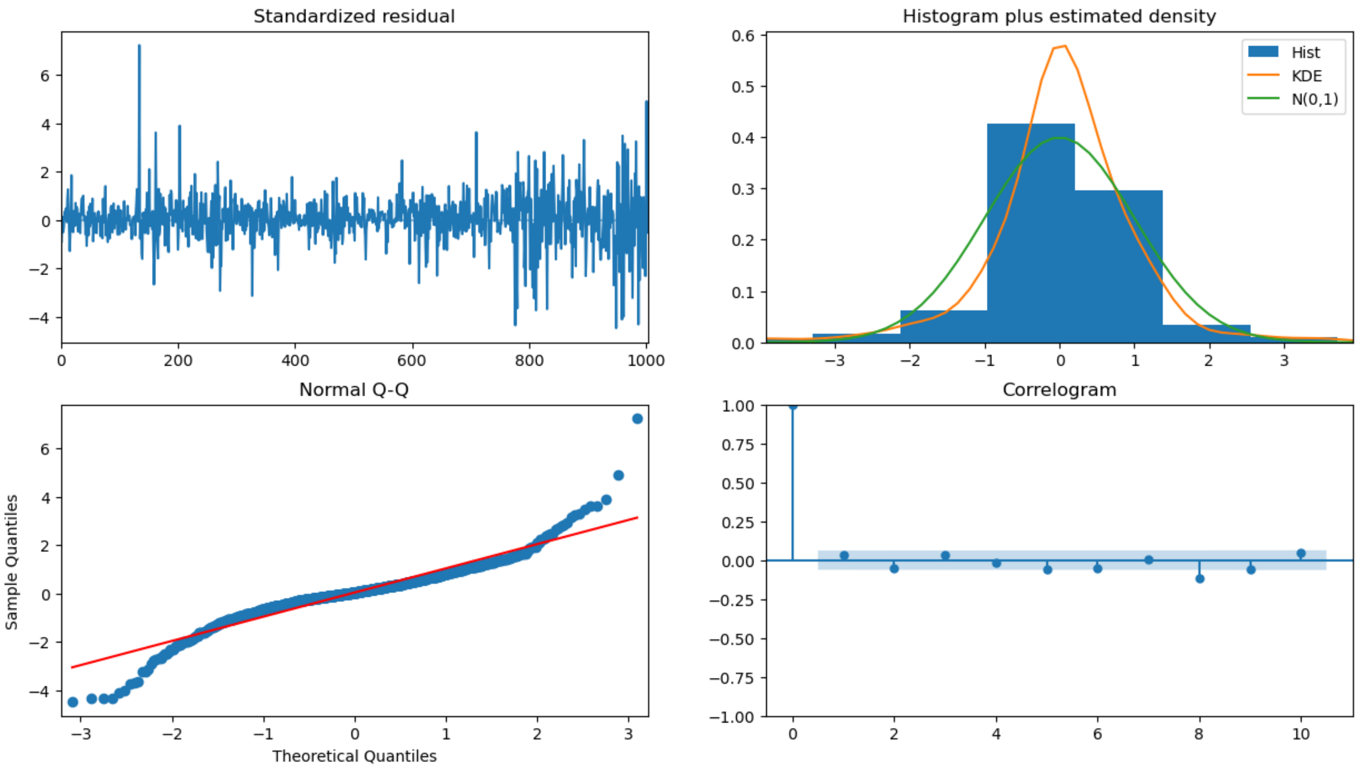


Fig 2. ARIMA model diagnostic plots

- Analysis of social media posts:

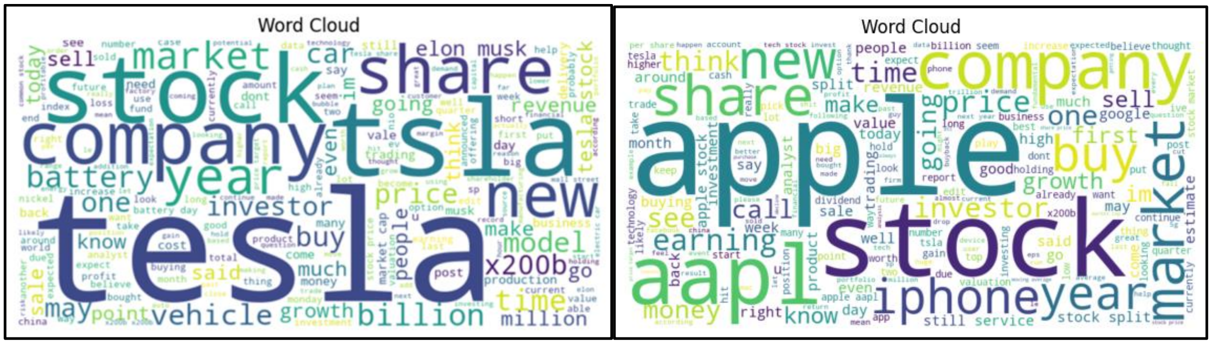


Fig 3: ‘TSLA’ and ‘AAPL’ Word Cloud from Reddit data

The analysis of Reddit data yielded word clouds that visually represent the most frequently occurring words shown in Fig.3. These word clouds help us identify prevalent topics like ‘battery’, ‘vehicle’, ‘elon musk’ for the ‘TSLA’ word cloud. This serves as a foundation for sentiment analysis.

- Twitter Data: Using the VADER sentiment analysis scores, we quantified the sentiments expressed in the tweets.

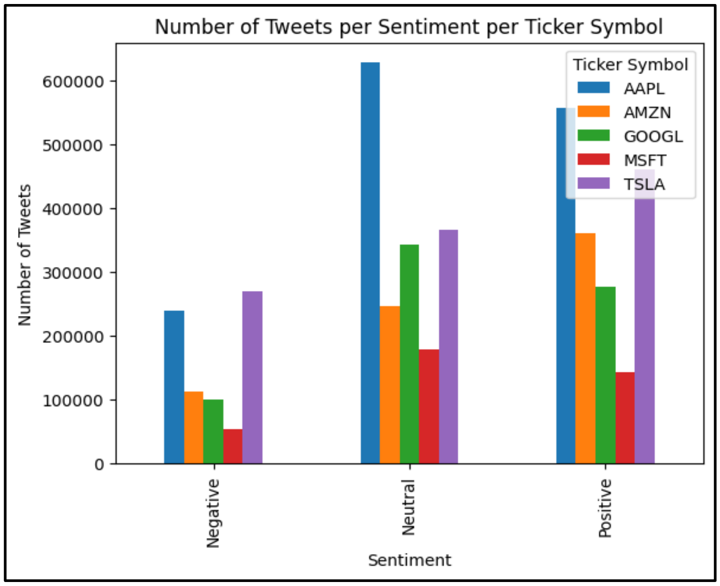


Fig.4: Tweets Sentiment distribution

The bar chart in Fig. 4 illustrates the collective sentiment towards various stocks, serving as a valuable reference for gauging public sentiment. Notably, the majority of tweets related to Amazon and Tesla stocks express positivity, whereas for the remaining stocks, sentiment appears to be predominantly neutral. The distribution of sentiment in the chart can serve as input data for sentiment analysis models.

- Model Evaluation Results: Shows the RMSE and MAPE scores after running the different models.

Model	**Time-Frame**	**RMSE**	**MAPE**
ARIMA	Next 1 Year	*8.64*	*0.12*
	Next 1 Month	*2.85*	*0.05*
	Next 10 Days	*1.55*	*0.03*
Prophet	Next 1 Year	*7.04*	*0.08*
	Next 1 Month	*0.81*	*0.01*
	Next 10 Days	*1.01*	*0.02*
LSTM	Next 1 Year	*2.03*	*1.83*
	Next 1 Month	*1.32*	*1.54*
	Next 10 Days	*0.36*	*0.45*

VARIMA	Next 1 Year	*5.30*	*0.07*
	Next 1 Month	*1.93*	*0.03*
	Next 10 Days	*1.01*	*0.02*
Multivariate LSTM	Next 1 Year	*0.03*	*0.02*
	Next 1 Month	*0.02*	*0.02*
	Next 10 Days	*0.01*	*0.02*

Table 1. Model Performance on Google Stocks