Stocks Recommendation USING Stock Price Prediction And Sentiment Analysis

Srikar Devulapalli, Shivam Mishra, Sanjith Reddy P V, Anirudh Sharma[1]*

Abstract
Investment in stocks can be intimidating for beginners and even experienced investors may struggle with where to invest next. Many failures in the stock market occur due to limited research and impulsive decision-making driven by greed or speed. To succeed in the stock market, it's important to conduct thorough research on companies and industries, diversify investments, invest for the long-term, and stay disciplined and patient. In our Solution we are using historical data and statistical models to predict future trends and outcomes. Help businesses and investors make informed decisions and plan for the future. We are also using natural language processing (NLP) to analyze text and determine the overall sentiment of stocks news. Combining forecasting and sentiment analysis can provide a more comprehensive understanding of the stock market and help investors make better decisions.

Keywords
NASDAQ, Sentiment Data, VADER, NLP, AFIN Score

[1] *Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*

Contents

1. Problem and Data Description

The main idea for our project is to predict the stock trend of a company. Most of the existing models get trained on the past stock details or on the sentiment analysis done on the news related to the company. Our model takes both the stock data and does sentiment analysis and predicts the stock trend based on their correlation on the prediction.

We have two datasets that we need to work on. One based on the stock and another related to the sentiment analysis. The first dataset is stock details of NASDAQ companies. The dataset contains the attributes like company name, date, opening price, closing price, low and high value and the amount of stock.

For sentiment analysis, we have dataset of tweets about companies from 2015 to 2020. The sentiment data is comprised of three .csv files. THe first one named Company.csv has information about the company and its code name for the further use. There are 6 total companies present in the data that are Apple, Google Inc, Google, Tesla, Amazon and Microsoft. The second file named Company_Tweet.csv have ID of the tweet and company code that the tweet corresponds to. The last file named Tweet.csv has the information of tweets like ID, tweet,data and others. All these corresponds to a company that we get to know across the tables(.csv files).

After this milestone we are planing to work on dynamic data from reddit as well that gives even more practical output with increased accuracy hopefully.

2. Data Preprocessing & Exploratory Data Analysis

Initially the labels of the companies are matched with the tweet's ID. Then for each tweet total interactions which is sum of number comments, retweets and likes. The date column of the tweets are converrted into readable data format and new

attribute of the day of the tweet is created. The tweets are sorted by date. The unwanted data that will not be correlated to the sentiment will be removed. These attributes are tweet ID, date, and number of comments, retweets and likes. We are considering the comments, likes and retweets as reach of the tweet and each of them affecting only the reach of that particular tweet, As for date we are keeping the readable format and discarding the other one.

We are merging this data with the stock data where we are converting the date into the same format as the tweet data. Only the days in which the stock price increased from opening to close time i.e., when the stock has profit for that day. We are sorting the stock value according to the date.

For the text values in the tweets, we are lower casing the text in the tweets so that the data will be uniformly present in the data. Then the stop words and punctuation are removed form the text using random expression library. All the special characters are removed as they will not suggest the sentiment of the tweet.

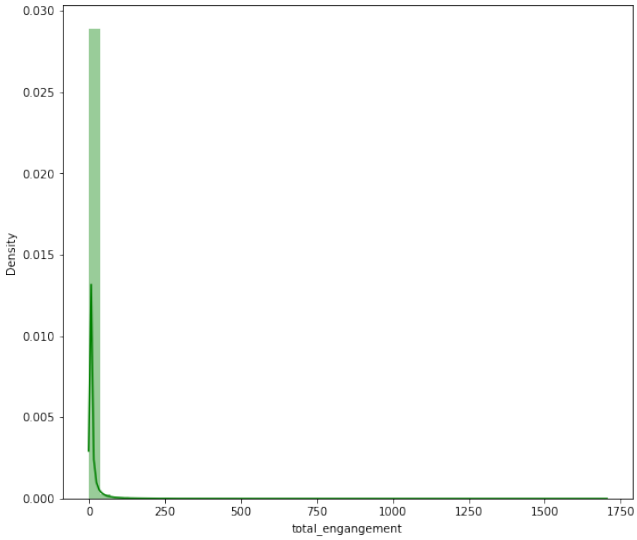

Figure 2.1: Dendrogram before PCA

2.1 Handling Missing Values

There were no missing values initially in the dataset. Missing values were encountered in the place where we have done derived calculations, which were replaced with 0s to make sense

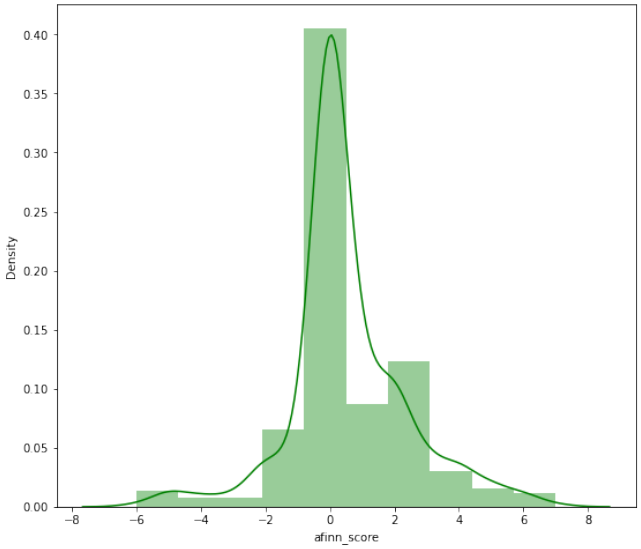

Figure 2.2: Affin sentiment score of apple company

2.2 Exploratory Data Analysis

Our end goal in this project is to provide stock recommendations using stock price prediction and sentiment data So, here we are making use of 2 datasets:

- Sentiment Data
- Financials Data

2.2.1 Sentiment Data

Affin and Vander models are used to perform sentiment analysis. The following diagrams tell us the positivity score of the tweets for corresponding company.

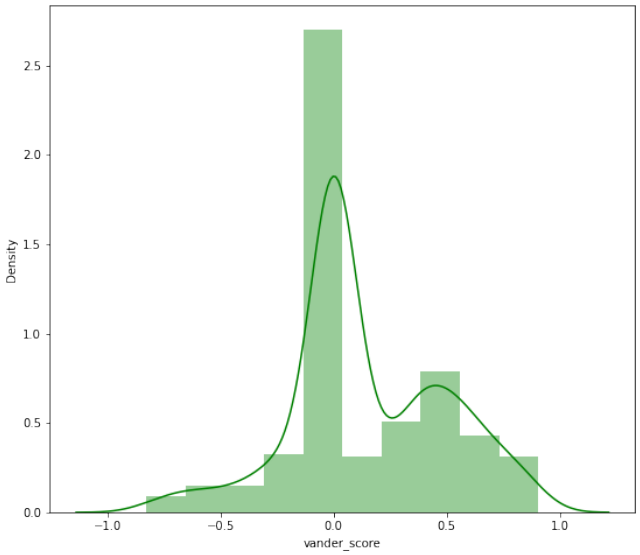

Figure 2.3: Vander sentiment score of apple company

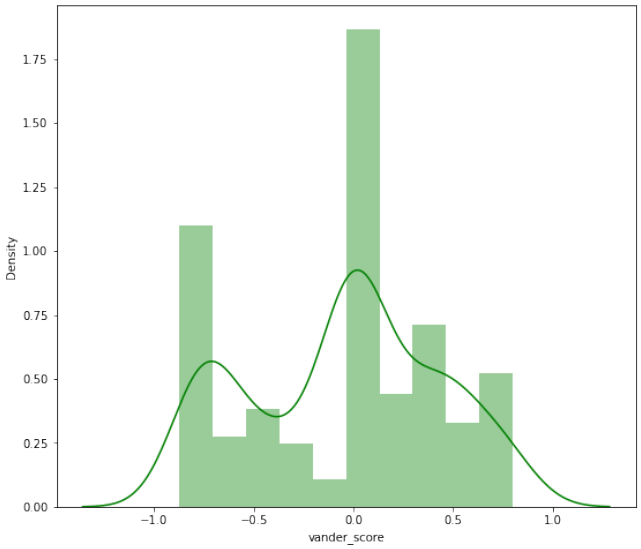

Figure 2.5: Vander sentiment score of Amazon.com company

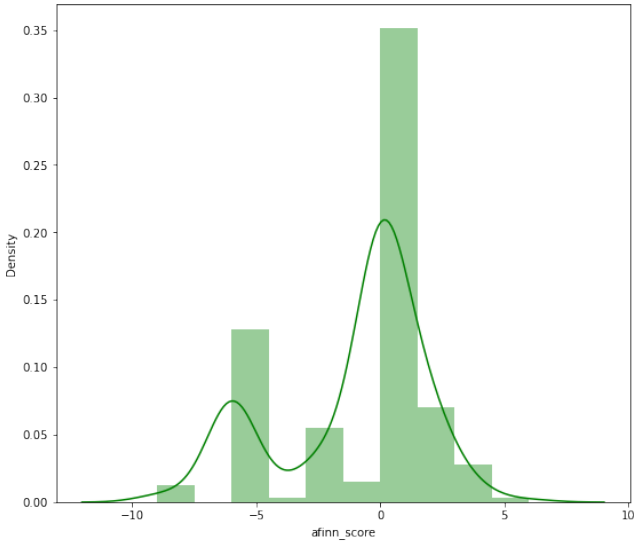

Figure 2.4: Affin sentiment score of Amazon.com company

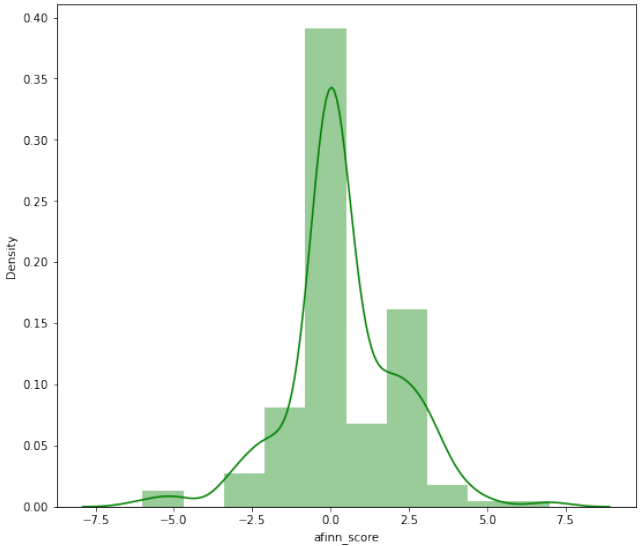

Figure 2.6: Affin sentiment score of Tesla Inc company

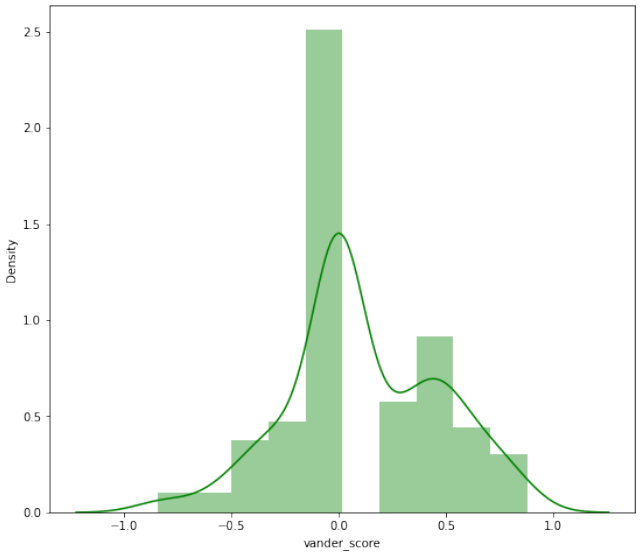

Figure 2.7: Vander sentiment score of Tesla Inc company

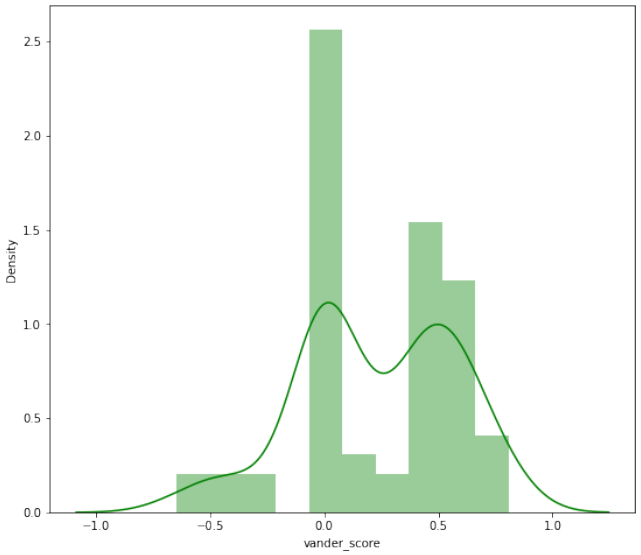

Figure 2.9: Vander sentiment score of Google Inc company

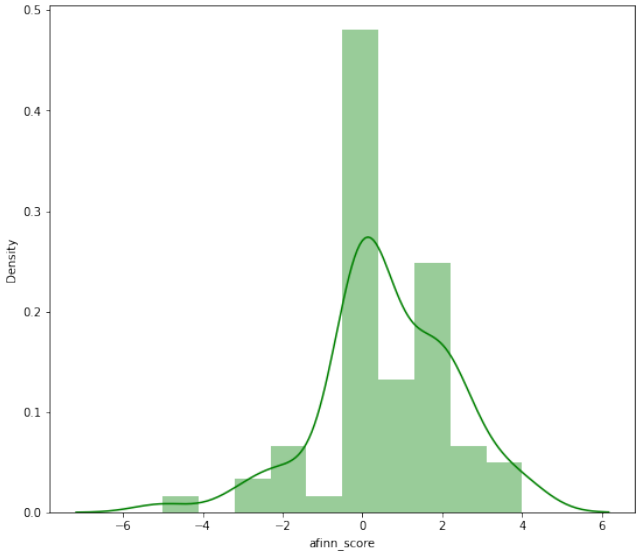

Figure 2.8: Affin sentiment score of Google Inc company

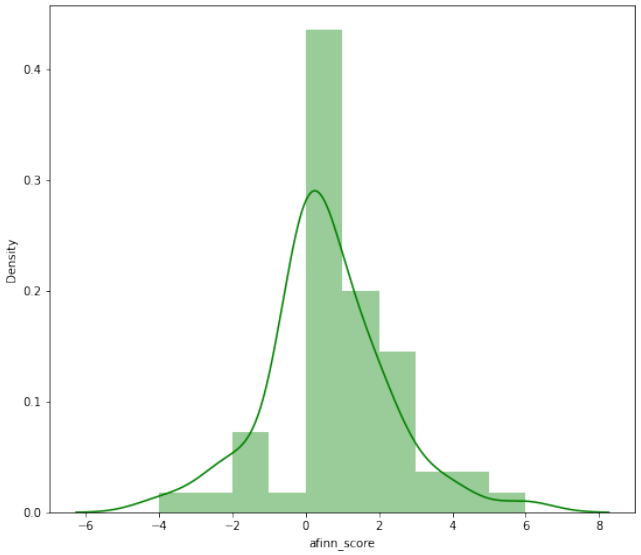

Figure 2.10: Affin sentiment score of Microsoft company

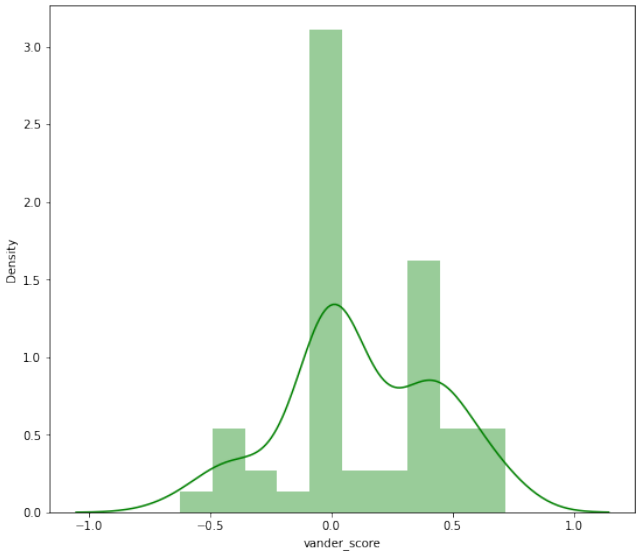

Figure 2.11: Vander sentiment score of Microsoft company

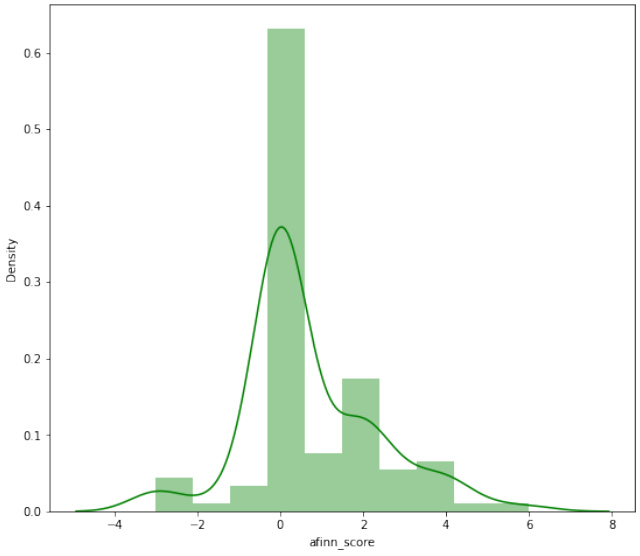

Figure 2.12: Affin sentiment score of Google Inc company

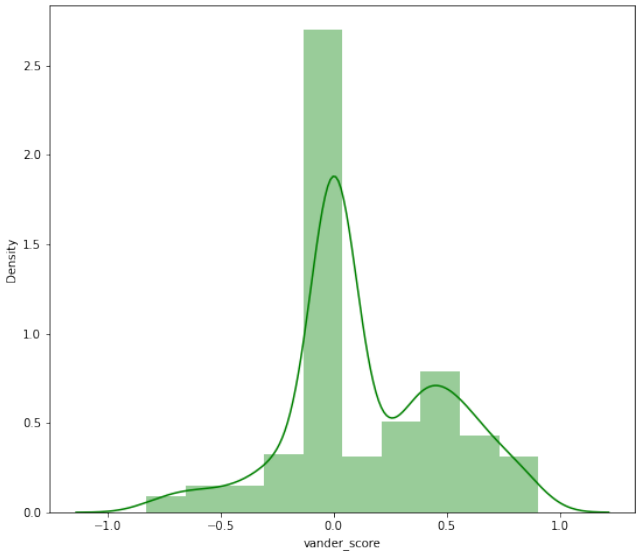

Figure 2.13: Vander sentiment score of Google Inc company

	ticker_symbol	day_date	vander_score	vander_sentiment
326	AAPL	2015-01-01	0.2500	Positive
2258299	GOOGL	2015-01-02	0.0000	Neutral
146	AMZN	2015-01-01	0.5719	Positive
3378294	TSLA	2015-01-02	0.0000	Neutral
3106251	MSFT	2015-01-01	0.4767	Positive
100	AAPL	2015-01-01	0.4389	Positive
3106258	MSFT	2015-01-01	0.0000	Neutral
336	AAPL	2015-01-01	0.0000	Neutral
2258285	GOOGL	2015-01-02	0.0000	Neutral
1864033	GOOG	2015-01-01	-0.4404	Negative

Figure 2.14:Vander Sentiment Analysis Results for Tweets

Discussion of Findings

We can observe that for a company if most of the comments are positive , we can see that there is increase in the prices , so we can say that there is positive correlation between sentiments and the stock prices. With these plots w can see that where most of the afinn scores lie in the distribution. If the peak is at more afinn score , it means that most of the sentiments are positive.

2.2.2 Financials Data

In financials data, we have added 4 derived metrics that would help in analyses and stock prediction. These derived metrics can provide valuable information to model to gain a deeper understanding of stock market trends and take more informed investment decisions. here are the short explanations and potential benefits of each derived metric:

- daily_return: measures the daily percentage change in stock price, which can help investors track stock performance and make informed investment decisions.
- daily_volume_change: measures the change in trading volume from one day to the next, which can indicate changes in investor sentiment or interest in a company's stock.
- price_change_from_open: measures the change in stock price from the opening price of the previous trading day, which can provide insights into intraday price movements and market reactions to news or events.
- price_range: measures the difference between the highest and lowest stock prices of the day, which can help investors identify volatility and trading opportunities.

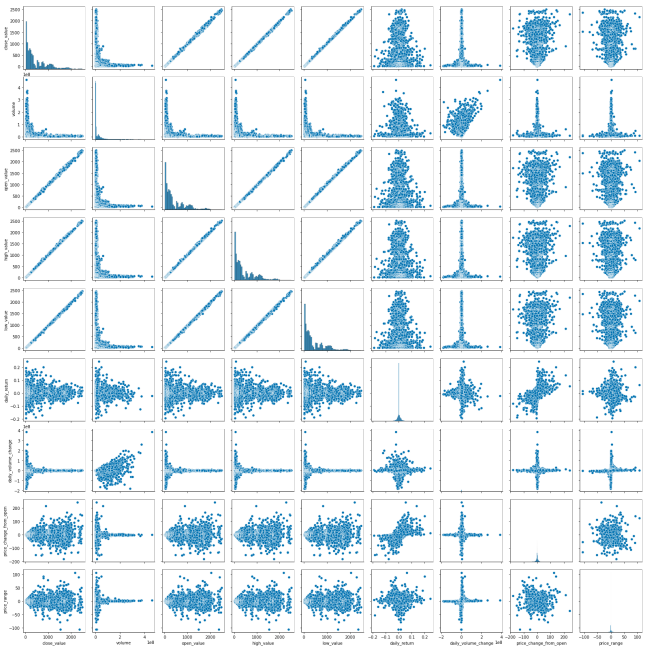

Figure 2.16: Exploratory Scatterplot Matrix for the Financials

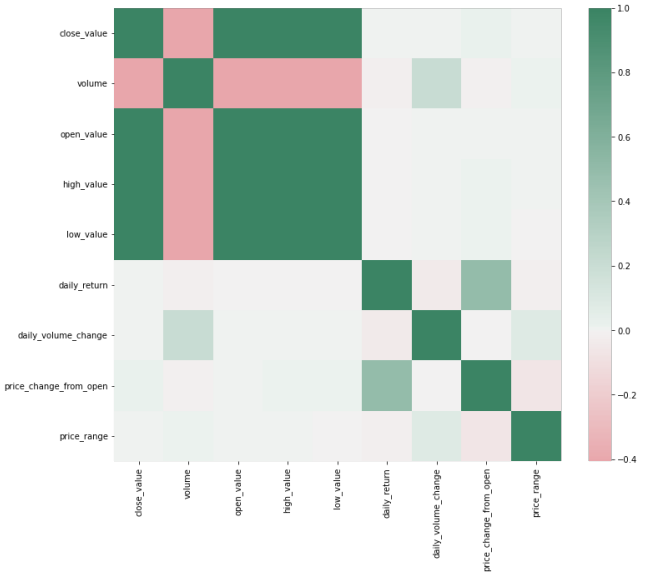

Figure 2.15: Correlation heatmap of numerical columns in dataframe

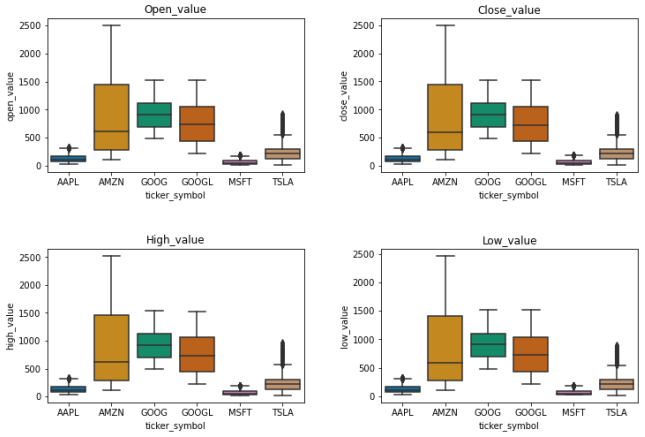

Figure 2.17: Boxplots of Open, Close, High, and Low Values Grouped by Ticker Symbol of top 6 companies

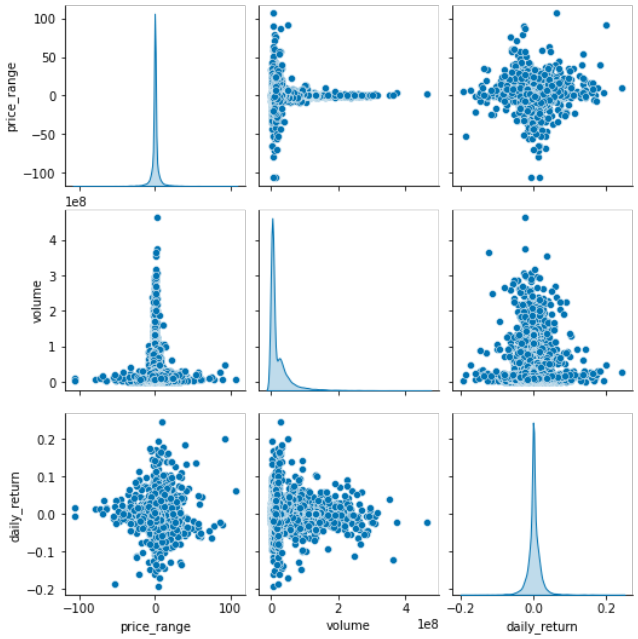

Figure 2.18: Pairwise Scatterplots with Density Plots for price_range, volume, and daily_return

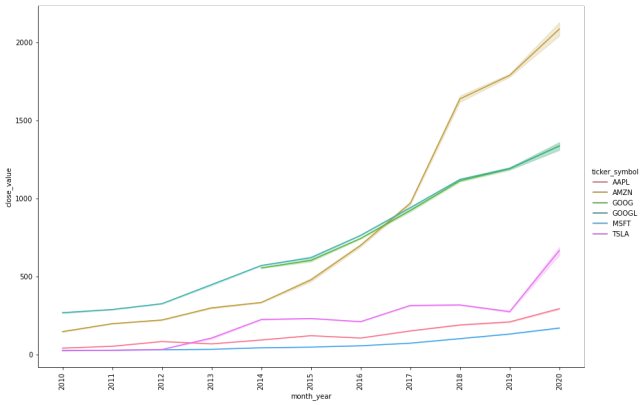

Figure 2.19: Closing Prices over Time for top Tickers

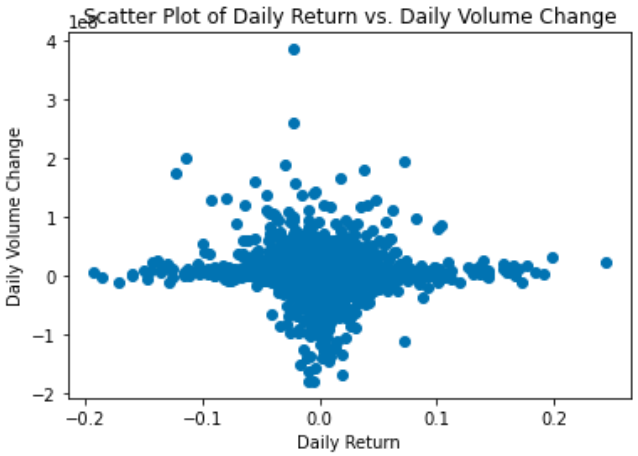

Figure 2.20: Relationship Between Daily Return and Daily Volume Change

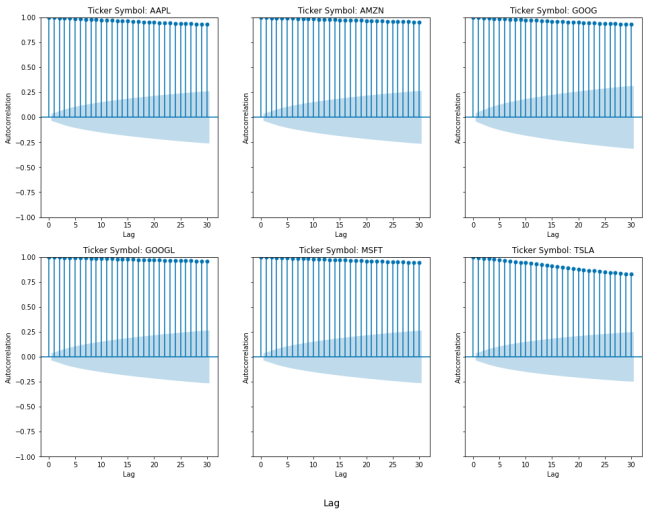

Figure 2.21: Autocorrelation Plot of Close Value for top tickers

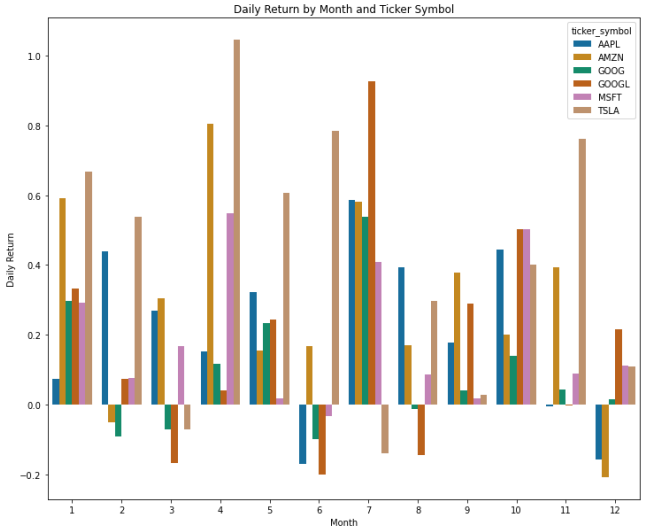

Figure 2.22: Daily Return by Month and Ticker Symbol

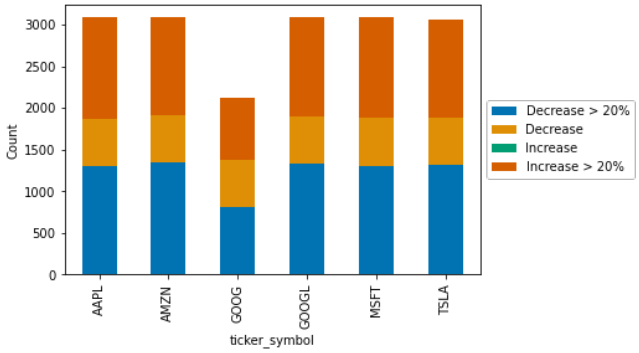

Figure 2.23: Daily Volume Change Categorization by Ticker Symbol

Discussion of Findings

- From Fig. 2.15 and 2.16, it is evident that close_value of a ticker highly correlates with open, high, low values
- From Fig. 2.15 and 2.16, we can also see that price change from open and daily return of a ticker have good correlation
- Over the 10 years period, AMZN had a fluctuated through broad range of values from 100 to 2500 (fig 2.17)
- Pairwise scatterplots for price_range, daily_return have high correlation i.e., their values are concentrated at mostly zero and change proportionally (fig 2.18)
- AMZN had seen rapid increase in stock price by 700% from 250 in 2014 to 2000 in 2020 (fig. 2.19)
- Scatterplots for daily_volume_change, daily_return have high correlation i.e., their values are concentrated at mostly zero and change accordingly with the other metric (fig 2.20)
- Autocorrelation plots in fig 2.21 suggest that all close value for top tickers are statistically significant
- The month of July(7) has observed consistent positive returns over the period of 10 years as per fig 2.22
- GOOG ticker had consistent stock value for the most of time, thus less number of changes in stock volume day over day, while others have observed change in stock volume at least once in a day. It is also evident that volume increased or decreased by 20% more frequently in a day (fig. 2.23)

3. Algorithm and Methodology

In order to get some idea about the stocks that are good for investment, a stock price prediction has to be made for a particular period of time. We are combining the results of stock price forecasting with sentiment analysis that gives the user insights of the investment that he/she wants.

We considered SP 500 companies to train our model. The list of the companies were scraped from Wikipedia using BeautifulSoup from bs4. It parses a tree that has the results from the corresponding web page. Then that table of tickers of the companies are extracted from the sparse tree into a list. Then using the tickers in the list, stock price for each day is extracted twelvedata api. There is a constraint is a of making nine calls per minute in twelvedata api which is solved by using sleep.

3.1 Stock Price Forecating

To predict the stock price based on the previous stock data, we tried different methods. Auto Regressive Integrated Moving Average (ARIMA) is one of the model that we used to predict the stock. These models assume that the future will be like the past, which isn't always true - especially during times of big changes in the economy or technology. Even if we try to use the best possible settings for our ARIMA models and check their accuracy with different data, they are still not be able to keep up with the way the market is changing or account for complex relationships between different factors, making their predictions less reliable. Because of this the performance of the model is not satisfactory.

As an alternative, LSTM networks are machine learning algorithms that can predict stock prices. Unlike ARIMA, these models do not assume that the future will be the same as the past, which makes them more flexible and powerful. By identifying patterns in data over long periods of time, LSTM networks can make better predictions of stock prices than traditional models.

The LSTM network makes predictions by using observations from the past to remember information about the future. In order to accomplish this, special memory cells and gates are used, which enable the network to selectively forget or remember past observations. Data from historical stock prices as well as other relevant variables, such as economic indicators, news articles, and social media sentiment, can be used to train LSTM networks. Traditional models can provide better predictions if they do not take into consideration market movements or other external factors. LSTM networks provide more accurate predictions when they take these factors into account.

LSTM model that we used in our model has a short term memory as 4 followed by a dense connections with adam optimizer. A batch size of 3 is taken. The model trains based on the mean squared error across the epochs and updates the weights. The data that was extracted from the api is sorted according to the date of the stock for each company. The dataset has attributes that are opening and closing stock price, min and max price and volume of data collected. After sorting this data is normalized using minmax scaler from -1 to 1 followed by dividing the data into training and testing dataset. Normalization is commonly used to improve model training and accuracy, and inverse normalization is required to return predicted values to their original scale. When inverting inputs, it is critical to understand the data flow between LSTM layers as well as the pre-processing procedures taken. The sequence of the time series data is reversed, and the number of features is equal the original dataset. Inverting inputs helps with model debugging and understanding how it has learnt.

3.2 Sentiment Analysis

To give more insights to the user about the stock and increase their confidence based on the trending news about the company, we are using sentiment analysis. The sentiments are

extracted from a site ”finviz”. When the sentiment analysis function is called, the request for the required ticker will be sent to the finviz url by appending the ticker name to the url. Then using BeautifulSoup the information is scrapped using html parser and the news table is found which contains the headlines of the companies latest news. Date and time is also extracted from the scrapped data and stored corresponding to the news. Then the name of ticker, date, time (if it exists) and news of the company are returned as the data for the sentiment analysers.

There are different types of sentiment analysers. From this we chose VADER. VADER is a rule-based algorithm for sentiment analysis which uses a lexicon of words with associated sentiment scores to accurately identify the sentiment of a piece of text. It is able to handle complex language features, identify intensifiers and negations, and produce sentiment scores ranging from -1 to 1. It has consistently outperformed other sentiment analysis algorithms in terms of accuracy.

The data extracted from the API is then passed to VADER. VADER gets the news of each company and it calculates the score based on it. This is then plotted as an interactive chart that will give the user an idea of the ticker’s immediate future market based on the news.

3.3 Ensemble

Then we correlate stock prices and sentiment scores. It is important because it can help investors make better-informed decisions about when to buy or sell stocks. By understanding how changes in sentiment are related to changes in stock prices, investors can anticipate market movements and adjust their investments accordingly. Additionally, this approach can also be used to assess market sentiment more broadly, which can be useful for developing macroeconomic forecasts and assessing overall market trends.

3.4 Integration

A backend API is created for fetching the data from the historical data and correlating with the sentiment analysis. Flask API, a light weight restful framework for python is used for this. It provides the object oriented code extension to the data models that are being used. Angular is a web development framework for creating single page application. We have used Typescript for integration of the front end with the data from the API. The front end takes the particular ticker from the user and then passes it to the API. The API passes this user entered data into the data mining mode and predictions are made for that particular ticker. Finally, interactive graphs returned back to the front end that are displayed in the Frontend to the user. From these graphs, the user can analyze how the stock is fluctuating as compared to the sentiment analysis and the historical data.

4. Experiments and Results

4.1 Experiments

For forecasting, initially, we tried RMSprop optimizer for LSTM(4) but the model overfits and performs poorly on the test set. In addition, we also tried LSTM(8) with ’adam’ optimizer, but the model did not converge. So, we have made use of LSTM(4), adam optimizer that gave use desired results solving the issue with convergence and overfitting.

AFFIN score was initially used to calculate sentiment score, but AFINN is a simpler method that does not take context into account. VADER uses a sentiment lexicon and rules to score sentiment, while AFINN uses a lexicon of words with assigned scores. VADER performs better on social media text due to its ability to handle slang, sarcasm, and non-standard language, but no sentiment analysis tool is perfect Due to these reasons, VADER is used to calculate sentiment scores.

4.2 Results

4.2.1 Website

Figure 4.2.1: Home Page

The Figure 4.2.1 is the home page of out web site. It has two parts. One is for ”Predict” and the other for ”Goal”. After clicking on either of them, the page gets redirected to corresponding page.

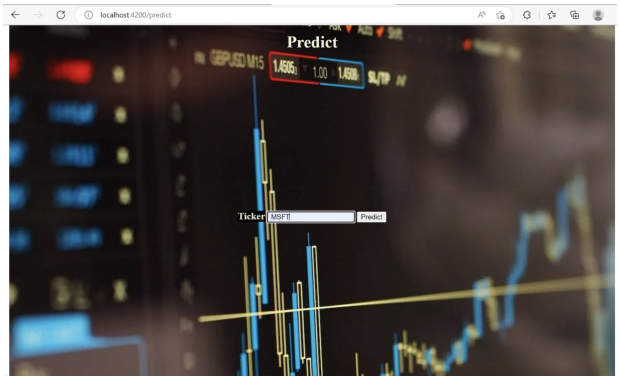

Figure 4.2.1a: Predict Interactive Page

The Figure 4.2.1a is the predict page where user can enter the ticker which he/she wants to predict.

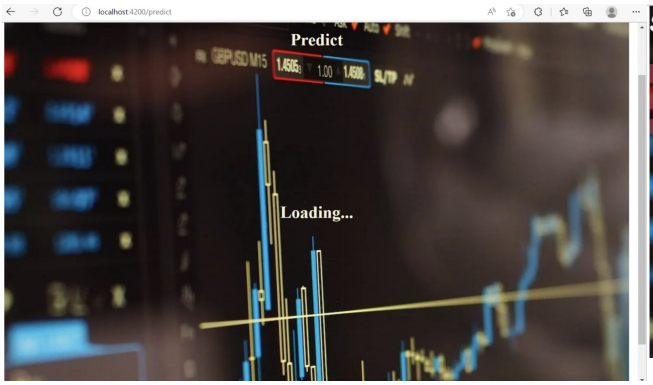

Figure 4.2.1b:Predict Loading Page

The Figure 4.2.1b represents the loading page that comes after user presses enter after giving input when the background process happens.

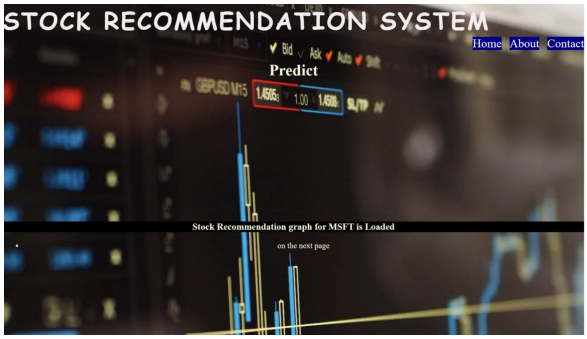

Figure 4.2.1c: Predict Acknowledgement Page

The Figure 4.2.1c is the acknowledgment that prediction results are out

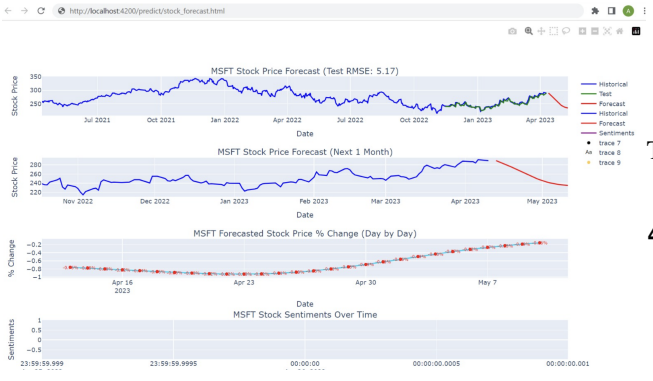

Figure 4.2.1d:Predict Results Page

The Figure 4.2.1d shows the results of prediction that are out

Figure 4.2.1e: Goals Input Page

The Figure 4.2.1e shows the results of prediction that are out.

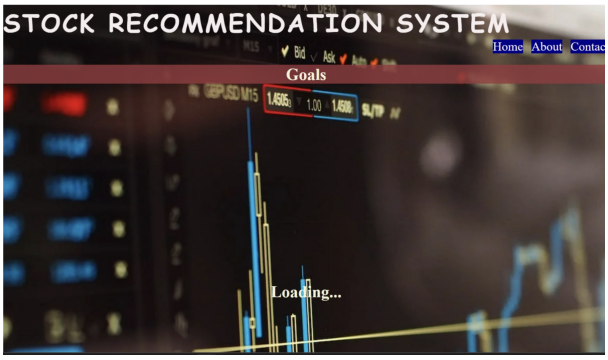

Figure 4.2.1f:Goals Loading Page

The Figure 4.2.1f shows the landing page of goals part.

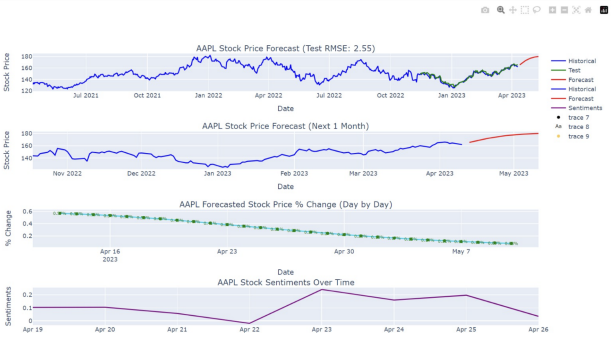

Figure 4.2.1g: Goals Result Page

The Figure 4.2.1g shows the landing page of goals result page.

4.2.2 Plots

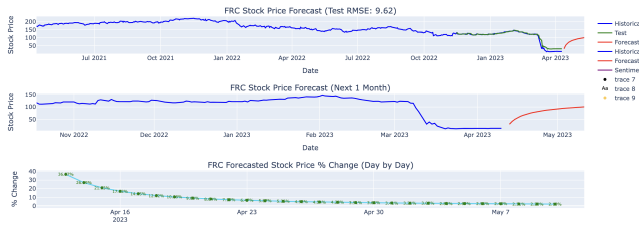

Figure 4.2.2a: Stock Price Forecast

The first subplot in Fig. 4.2.2a shows the historical and forecast stock prices of a particular ticker along with the test RMSE (Root Mean Squared Error) that instills the confidence based on the test data predictions as highlighted in green.

The second subplot in Fig. 4.2.2a shows the historical and forecast stock prices of a particular ticker for the next month.

The third subplot shows Fig. 4.2.2a the percentage change in the forecast stock price day by day.

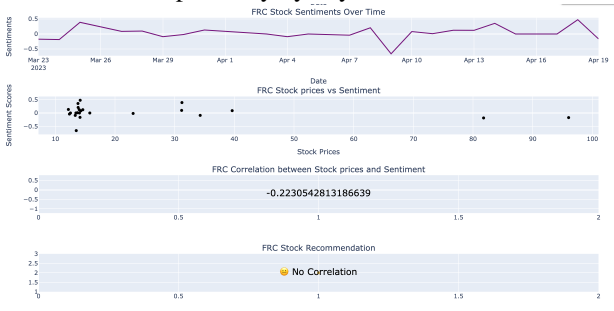

Figure 4.2.2b: Sentiment Scores and Ensemble

The first sub plot in Fig. 4.2.2b adds a line chart that displays the sentiment scores over time. This can help identify trends and patterns in the sentiment scores which may have an impact on the stock prices. For example, if the sentiment scores are consistently negative over a period of time, this may suggest that the stock prices are likely to decrease.

In the second sub plot in Fig. 4.2.2b, a scatter plot is added to display the correlation between stock prices and sentiment scores. This helps to determine the strength and direction of the relationship between the two variables. The correlation coefficient is then calculated, which further quantifies the relationship between the two variables. The corresponding emoji recommendation is displayed based on the correlation coefficient. This can provide a quick and easy way to understand the relationship between the two variables.

The third sub plot in Fig. 4.2.2b adds the correlation coefficient as a text label. This helps to provide a numerical value for the correlation coefficient which can be used for further analysis.

Finally, the recommended emoji is added as a text label in the fourth sub plot in Fig. 4.2.2b. This can be useful in quickly conveying the recommendation to the viewer, without having to read through a detailed report or analysis.

Overall, the subplots and various settings for the plot appearance help to provide a clear and concise visual representation of the relationship between the sentiment scores and the stock prices, making it easier to analyze and understand the data.

5. Deployment and Maintenance

Here is the link to the repository - link

Here are the steps we followed to run our website -

1. Go to the main directory which is the stock recommendation system. This step asks you to navigate to your project's root directory, which is named "stock recommendation system" in this case. Use the command cd stock-recommendation-system in the terminal to change the current working directory to the project folder.

2. Type npm install to install all libraries in the terminal. This command installs all dependencies listed in your project's package.json file. It downloads the required packages and stores them in the node_modules folder.

3. Type ng serve to run in the browser in the terminal. The ng serve command starts a local development server, watches your files, and rebuilds the app when you make changes to those files. By default, it serves the application on http://localhost:4200/. You can also use ng serve –open or ng serve -o to automatically open the application in your default web browser.

4. Click on the generated URL. Visit the generated URL (usually http://localhost:4200/) in your web browser to view and interact with your Angular applicatione

6. Summary and Conclusions

The deployed webpage provides insights for the user about making safe investments in stock. The predict part returns the future predictions of stock price of a Ticker of user choice from SP 500 stock list which is backed up by providing sentiment scores of the particular company. The user can look at the predictions and sentiment of the company and decide the safe stock that can be invested in.

The Goal part takes input from the user the amount of the investment that the user wants to make and duration for which he wants to invest in. Our model returns the best stock that the user can invest in that is calculated based on the stock that has a maximum day to day increase in stock price.

6.1 Future Scope

The present model that we used in Goals returns the most profitable stock that the user can invest in. However the market behaves in such a way that the company that is expected to have high growth can go to negative because of their decisions, market or any other reasons. If this happens, then the user will incur huge losses. This is a risky investment. If the user is ready to make a risky investment, then our model works very well. If they want to make a safe investment by mentioning that he can invest this much amount of money for this much amount of time and expects this much of returns thinking of making a safe investment, then we can make change in the logic to suggest a stock based on the user requirements. This can be implemented in the future. Also, the model can be deployed online and made available to other users as well in the future.

Acknowledgments

We would like to express our sincere gratitude to Prof. Hasan Kurban for his inspiring lectures that motivated our team to undertake the challenging task of creating a real-time stock recommendation project. His insightful guidance and encouragement throughout were invaluable in helping us achieve our goals.

References

- pandas
- LSTM Models for Time Series Forecasting
- Brownlee, J. (2018). Deep Learning for Time Series Forecasting.
- Train/Test Split and Cross Validation in Python
- VADER
- ngserve
- Install NodeJs