Project Title: Stock Market Analysis and prediction

Team Members:

- Likita Chamakura Suresh
- Naga Srija Guntupalli
- Prathyusha Daroor
- Sowrab Sahini

Goals and Objectives:

- **Motivation**

  There are two ways of predicting the stock market – fundamental analysis and technical analysis. So, we would like to combine the power of fundamental and technical analysis to analyze stock market data.

- **Significance**

  As many investment companies and individuals today are very much active in the stock market, they perform regular research by following news, studying the company trends and others before making any investment.

Building models by training it with past and current data is useful to investors and traders to have a proper understanding of the market fluctuation helping them to make better decisions for trading and investment.

- **Objectives**

  The main objective is to improve stock market analysis using sentiment analysis, linear regression, and Neural Networks.

- **Features**

  Given a stock market ticker we analyze the financial documents of the given company and will also predict the stock prices and visualize the prices over time.

Increment:

- **Related Work (Background):**

  For forecasting stock prices, most past research has relied on traditional techniques such as linear regression, Random Walk Theory, Moving Average Convergence / Divergence, and certain linear models such as Autoregressive Moving Average (ARMA). Machine learning has been shown to improve stock market prediction in recent research.

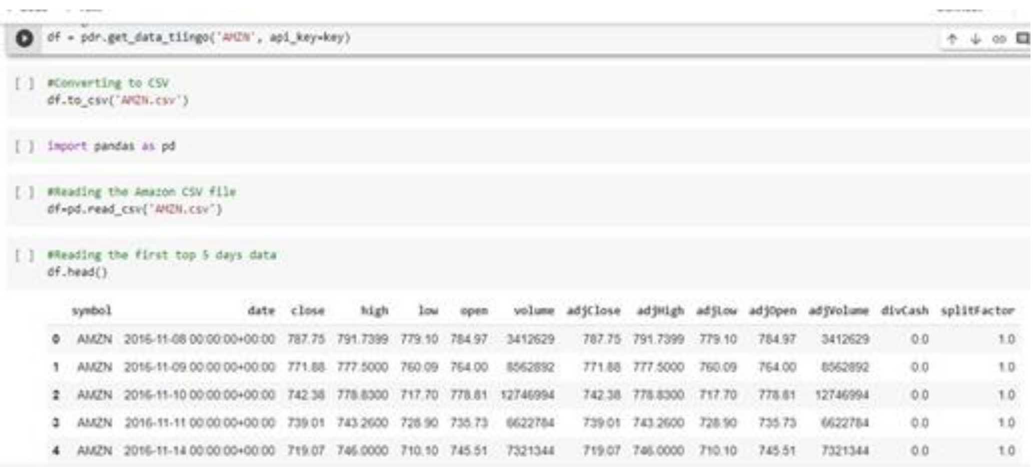
  Some neural network approaches, such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and deep neural networks, such as Long Short-Term Memory (LSTM), have also showed promising results in this kind of predictions.

Roman utilized historical stock market data from five nations to train RNN models, which were then used to forecast stock return trends in Canada, Hong Kong, Japan, the United Kingdom, and the United States. The data consists of daily closing prices of five markets.

Selvin used a comparative examination of several Deep learning approaches to forecast the stock price of NSE listed businesses for the period of July 2014 to June 2015, the data set contains minute-by-minute stock prices for 1721 NSE-listed businesses.

- **Dataset**

TINGO API from panda's data reader. This is used to get the historic data of the specified company using the API key. [3]



```
df = pdr.get_data_tiingo('AMZN', api_key=key)
```

```
[ ] #Converting to CSV
    df.to_csv('AMZN.csv')
```

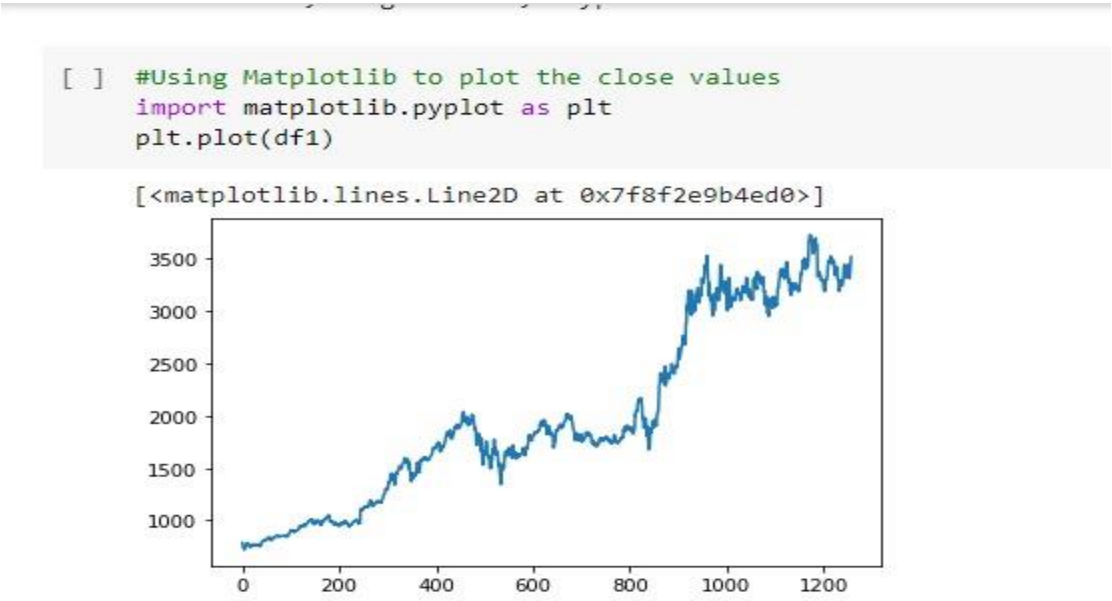
```
[ ] import pandas as pd
```

```
[ ] #Reading the Amazon CSV file
    df=pd.read_csv('AMZN.csv')
```

```
[ ] #Reading the first top 5 days data
    df.head()
```

```
  symbol                       date   close      high    low   open    volume  adjClose   adjHigh  adjLow  adjOpen  adjVolume  divCash  splitFactor
0   AMZN  2016-11-08 00:00:00+00:00  787.75  791.7399  779.10  784.97   3412629    787.75  791.7399  779.10   784.97    3412629      0.0          1.0
1   AMZN  2016-11-09 00:00:00+00:00  771.88  777.5000  760.09  764.00   8562892    771.88  777.5000  760.09   764.00    8562892      0.0          1.0
2   AMZN  2016-11-10 00:00:00+00:00  742.38  778.8300  717.70  778.81  12746994    742.38  778.8300  717.70   778.81   12746994      0.0          1.0
3   AMZN  2016-11-11 00:00:00+00:00  739.01  743.2600  728.90  735.73   6622784    739.01  743.2600  728.90   735.73    6622784      0.0          1.0
4   AMZN  2016-11-14 00:00:00+00:00  719.07  746.0000  710.10  745.51   7321344    719.07  746.0000  710.10   745.51    7321344      0.0          1.0
```

Fig: Top 5 values of the Dataset



```
[ ] #Using Matplotlib to plot the close values
    import matplotlib.pyplot as plt
    plt.plot(df1)
```

```
[<matplotlib.lines.Line2D at 0x7f8f2e9b4ed0>]
```

Fig: Visualizing the dataset

- **Detail design of Features:**

  We are planning to use regular expression to clean the raw data that has all special characters and try to implement different classification and regression algorithms using NLTK's sentimental analysis.

  While analyzing the models which can be used for stock market prediction (time series data) we found out that LSTM is one such deep learning model which can be used to predict stock market data.

- **Analysis**

  Data for prediction is obtained from TINGO API using pandas' data frame.

  The variables that are used to train the model are:

  - Opening Price
  - High Price
  - Low Price
  - Closing Price
  - Trading Volume

- **Implementation**

○ To implement the stock market prediction, we use the data from TIINGO API as our source of historic data of the specified company ticker. [3]

○ We are dividing our dataset into two parts – training and testing datasets. Out of the total 300% of the data we split the dataset into 65% training dataset and the rest into test dataset.

○ We reshape the data to fit into the LSTM model then we use the LSTM to build the model and train the data. [4]. The number of epochs for our model is 300.

○ To a point, the models improve as more training epochs are completed. As the number of epochs increases, the test accuracy increases. If we notice that the accuracy is declining it means that the model is overfitting. We have used epochs 300 to increase the accuracy of our model and did not encounter any data overfitting error at that level of epoch count. After training the model we predict the data.

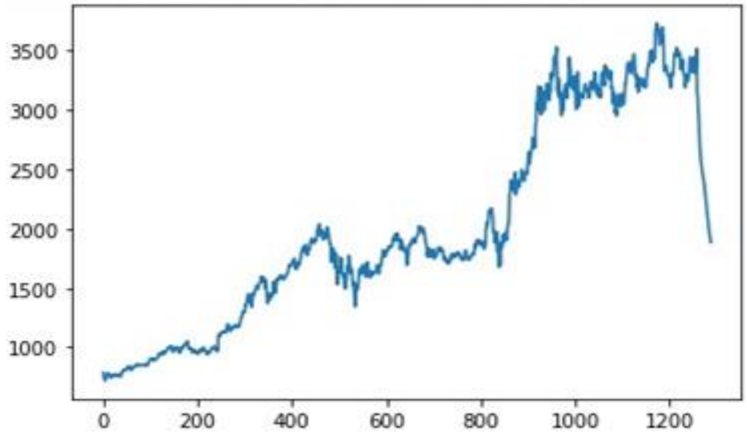


Fig: Prediction for upcoming 30days Stock price.

- **Preliminary Results**
  For example, if we take any company(amazon) tweets and when all the above technics are applied, we are expecting a percentage of positive and negative tweets.

- **Project Management**
  - **Implementation status report**
    - **Work completed:**
      - **Description**: Developed deep learning model to predict upcoming 30days stock price.

      - **Responsibility (Task, Person)**:
        Background research work: Prathyusha Daroor, Sowrab Sahini
        Designing: Prathyusha Daroor, Guntupalli Naga Srija
        Coding: Sowrab Sahini, Likita Chamakura Suresh, Prathyusha Daroor, Guntupalli Naga Srija.
        Documentation for this phase: Likita Chamakura Suresh, Sowrab Sahini

      - **Contributions (members/percentage):**
        Prathyusha Daroor: 25%
        Likita Chamakura Suresh: 25%
        Sowrab Sahini: 25%
        Guntupalli Naga Srija: 25%

    - **Work to be completed**
      - **Description:** Need to implement the random forest regressor algorithm to predict the stock price values for a given period, linear regression to compare the predicted prices with actual prices, classify the data as positive and negative using vader lexicon sentimental analysis.

      - **Responsibility (Task, Person):**
        Background research work: Prathyusha Daroor, Sowrab Sahini
        Designing: Prathyusha Daroor, Guntupalli Naga Srija
        Coding: Sowrab Sahini, Likita Chamakura Suresh, Prathyusha Daroor, Guntupalli Naga Srija.
        Documentation for this phase: Likita Chamakura Suresh, Sowrab Sahini
        Testing: Likita Chamakura Suresh, Guntupalli Naga Srija.

      - **Issues/Concerns:** Being a CSE grad, we faced difficulties to understand and implement the proposed project using deep learning models.

- **References/Bibliography**

[1]
M. Zu, "NLP for Stock market prediction with reddit data," [Online]. Available: https://web.stanford.edu/class/cs224n/reports/final_reports/report030.pdf.

[2] "U.S security and exchange commission," 9 jan 2017. [Online]. Available: https://www.sec.gov/edgar.shtml.

[3] "TIINGO," [Online]. Available: https://api.tiingo.com/documentation/general/overview.

[4] "Keras LSTM layer," [Online]. Available: https://keras.io/api/layers/recurrent_layers/lstm/.

INCREMENT 2

- **Introduction**

  In today's economy, the stock market, often known as the equity market, has a significant effect. The rise or decline in the share price has a significant impact on the investor's profit. Predicting stock prices is a challenging task as it depends on various factors such as the global economy, company's financial reports and performance etc. So by analyzing the trend over the last few years could prove to be highly useful for making stock market movements.

  For forecasting an organization's stock price, two basic methodologies have been offered in the past - technical and qualitative analysis. For predicting the future price of a stock, the technical analysis approach employs past stock prices such as closing and opening prices, neighboring close values, and so on. The qualitative analysis, on the other hand, is based on external elements such as business profile and market scenario.

  In this project, we will look at historical data on a google company's stock prices. We'll use a combination of machine learning algorithms to forecast this company's future stock price, starting with simple approaches like averaging and linear regression and progressing to more complicated techniques like LSTM (long-term short memory).

- **Background**

  For forecasting stock prices, most past research has relied on traditional techniques such as linear regression, Random Walk Theory, Moving Average Convergence / Divergence, and certain linear models such as Autoregressive Moving Average (ARMA). Machine learning has been shown to improve stock market prediction in recent research.[5]

  Some neural network approaches, such as Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and deep neural networks, such as Long Short-Term Memory (LSTM), have also showed promising results in this kind of predictions.

  Roman utilized historical stock market data from five nations to train RNN models, which were then used to forecast stock return trends in Canada, Hong Kong, Japan, the United Kingdom, and the United States. The data consists of daily closing prices of five markets. [5]

  Selvin used a comparative examination of several Deep learning approaches to forecast the stock price of NSE listed businesses for the period of July 2014 to June 2015, the data set contains minute-by-minute stock prices for 1721 NSE-listed businesses.

- **Model**

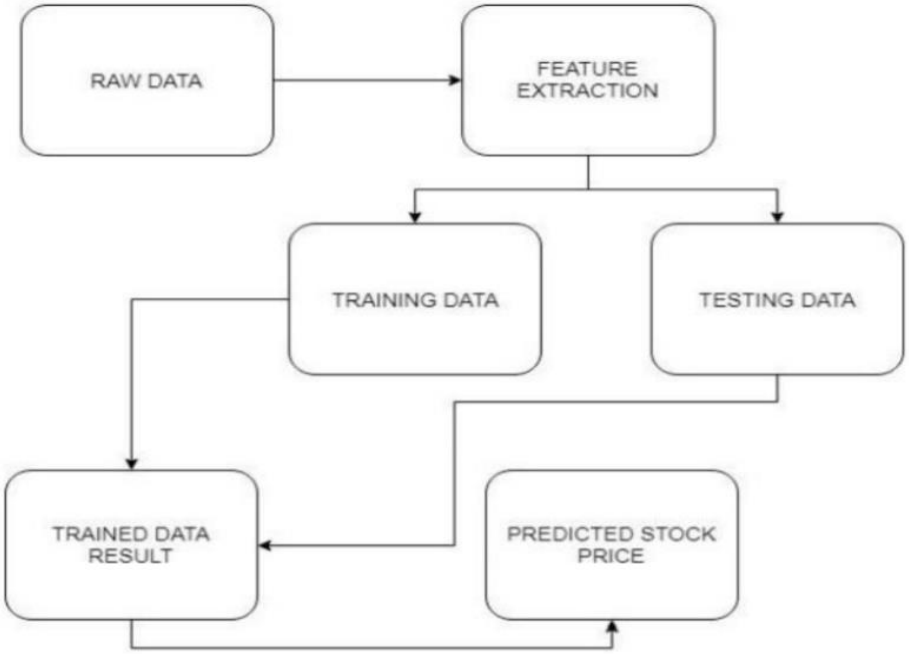


Fig: Architecture Diagram

We have used twitter’s dataset using its API from Twitter API. The data is then categorized into a training and test data set. Model is trained using train data and then it is validated using test data. A well-modified categorization is retrieved from the provided dataset, and a graph set is plotted to obtain the necessary output, which is the stock prediction range. Also, we will be displaying a positive or negative result of stock price.

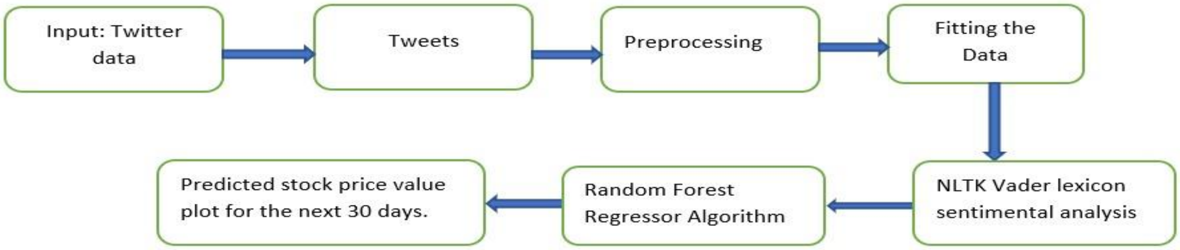

Input: Twitter data
Tweets
Preprocessing
Fitting the Data
NLTK Vader lexicon sentimental analysis
Random Forest Regressor Algorithm
Predicted stock price value plot for the next 30 days.

Fig: Random Forest Algorithm

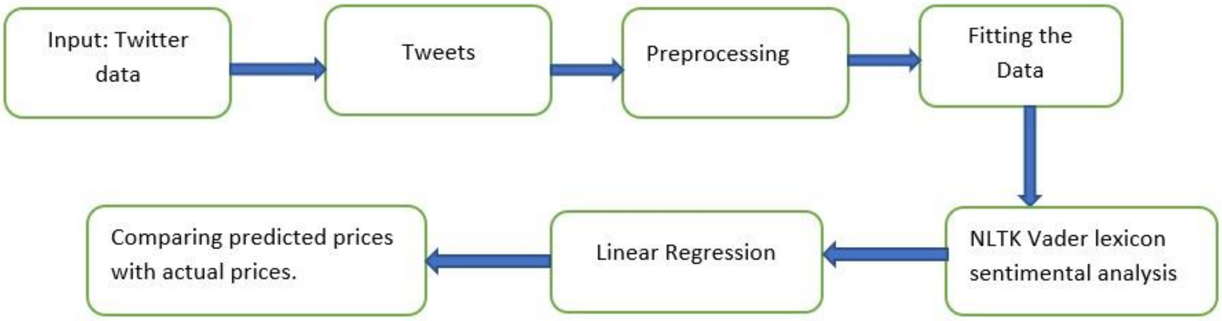

Input: Twitter data
Tweets
Preprocessing
Fitting the Data
NLTK Vader lexicon sentimental analysis
Linear Regression
Comparing predicted prices with actual prices.

Fig: Linear Regression

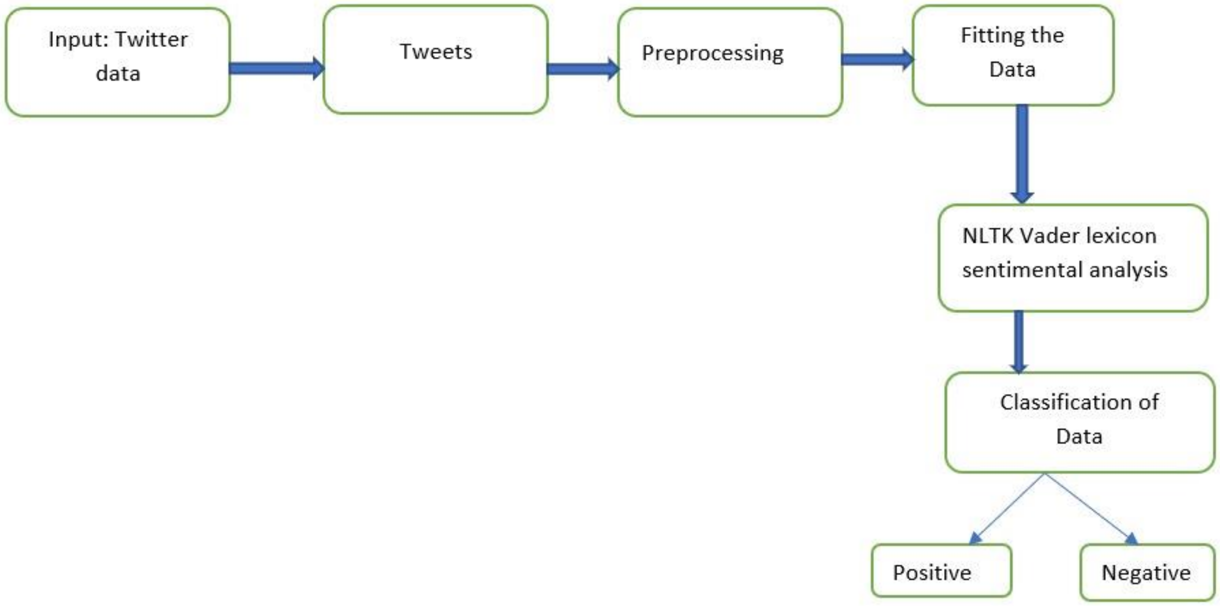


Fig: Workflow Diagram

As shown in the workflow diagrams above, we are fetching the data using twitter's API data from the panda's data library and then we are displaying the data. Then, we use the data to estimate the stock price over a period. We have shown two different ways of plotting results.

We will be displaying if the stock value is positive or negative so that users can decide whether to invest or not. Also, we will be predicting stock price variation for the coming days, and it is displayed in the form of a plot.

Dataset

We are using Tweepy API to fetch results from twitter and store it as a CSV file. We use the access token and secret token with the consumer key and consumer secret key. After authentication is successful, we will be able to access the API to retrieve the tweets.

Tweepy supports both OAuth 1a and OAuth 2. After we retrieve the tweets, we process the tweets and store it in the CSV file.

Design features: We tried to map the daily price of the stock with the daily news that we obtain from twitter. After mapping the price and creating the dataset of the daily closing price with the tweets, we use the NLTK's vader lexicon sentiment analysis tool. Then we calculate the polarity score of every tweet and classify it as positive and negative. After we classify the tweets, we use the random forest regressor then we use the supervised learning algorithm that is the random forest regressor as our model to train the data. Using the random forest regressor we try to predict the prices and compare it with the actual price.

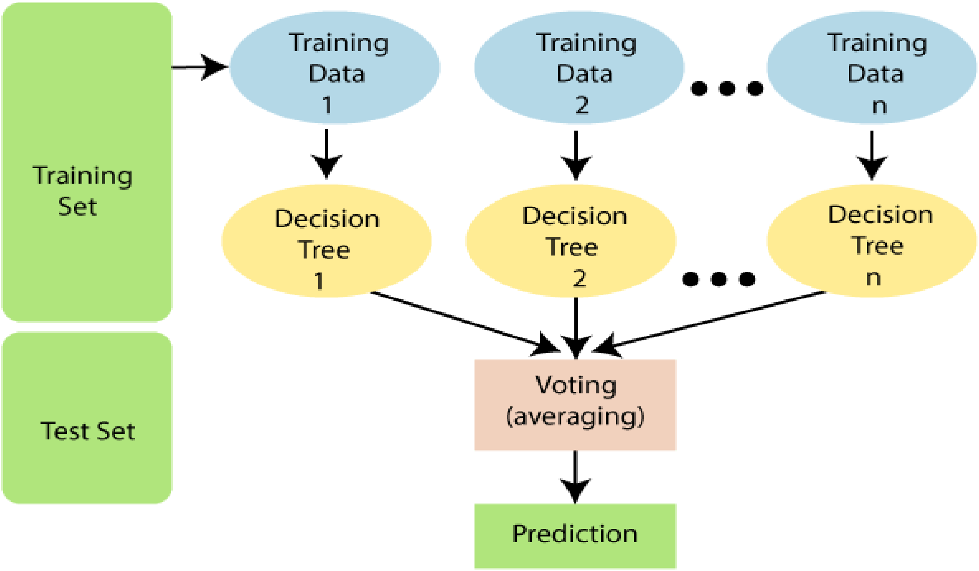


Fig: Random Forest regressor

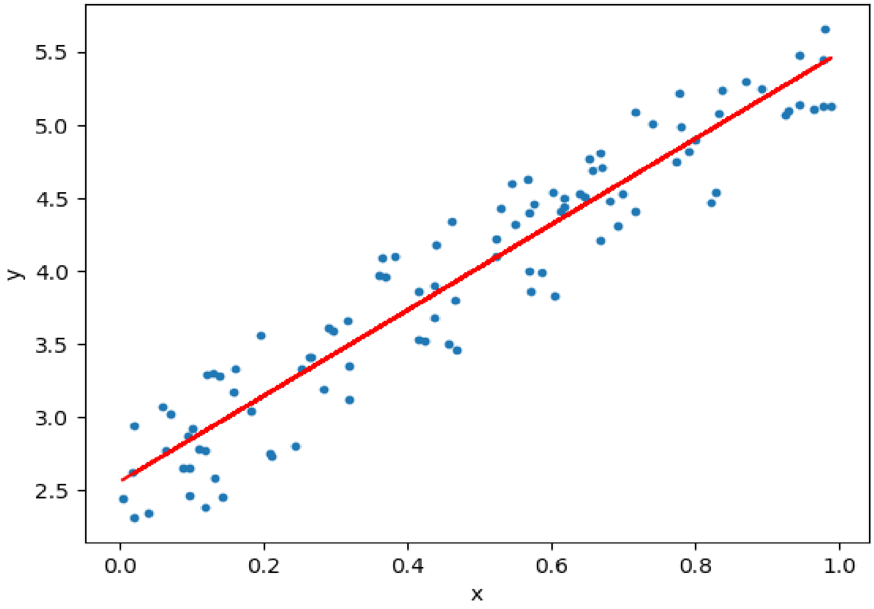


Fig: Linear Regression

- **Analysis of data**

  o Data Pre-processing: First we fetch the data from twitter which might have a lot of special characters, then we use the regular expressions to clean the data to include only alphanumeric character, after we clean the dataset, we append all the tweets of that day and store it in the data frame. Then we use the CSV file from yahoo finance API to map it with the daily tweets.

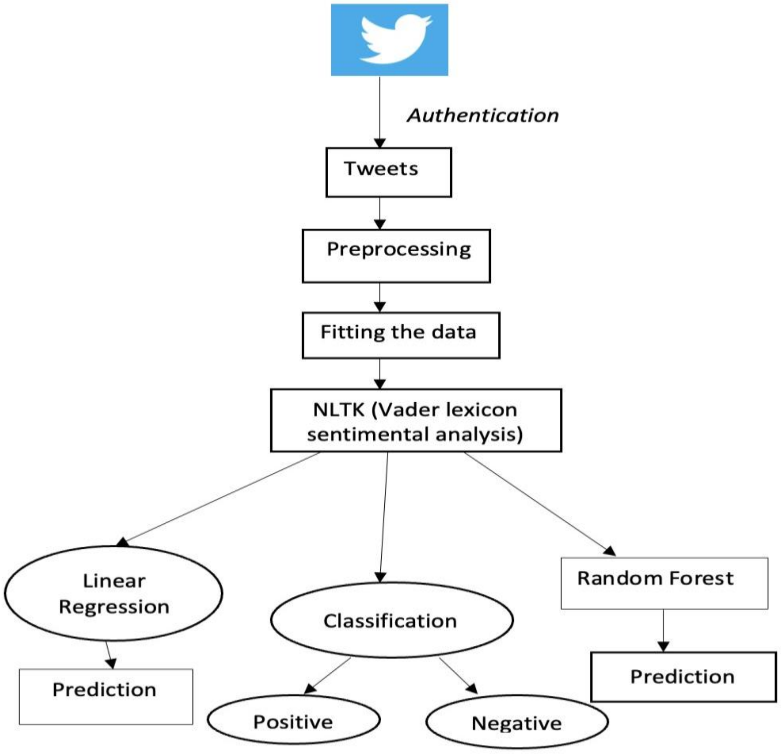


Fig: Graph Model

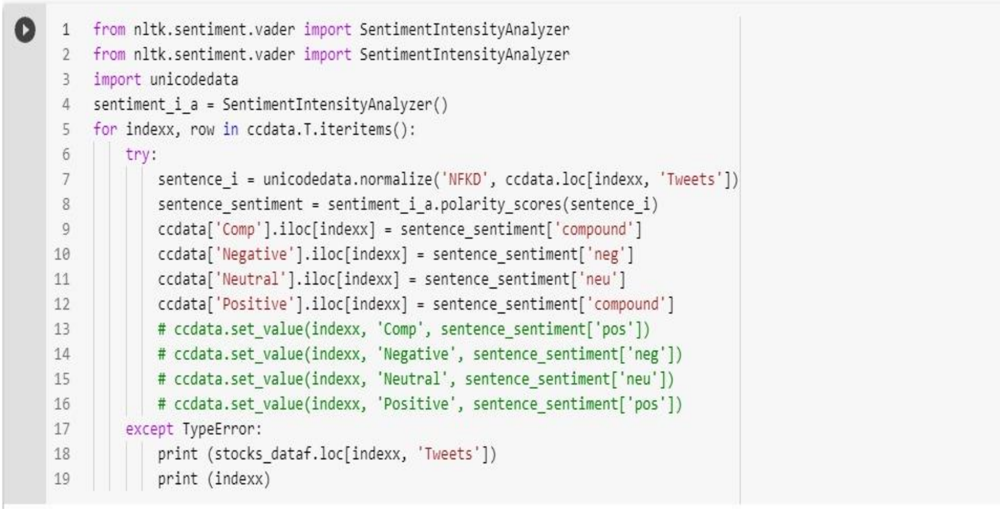
Twitter dataset is used to classify and predict and data by using 3 different techniques such as linear regression, classification, and random forest regressor.
Classification: Classifying the data as positive or negative.
Random forest regressor: Predicting the stock price value for the upcoming 30 days.
Linear regression: Comparing the predicted prices and the actual prices.

- **Implementation**
  - **Algorithms**
    1. Extracting the tweets from the twitter.
    2. Preprocessing: Regular expressions are used to clean the data.
    3. The cleaned data of that day is stored in the data frame.
    4. NLTK (vadar lexicon sentimental analysis) is used to classify the data.
       1. Based on sentimental analysis, the data is classified as positive and negative.
       2. Random Forest regressor is used to predict the stock prices for the upcoming 30days.

3. Using the linear regression, the actual and predicted stock prices are compared.

- **Explanation of implementation**:

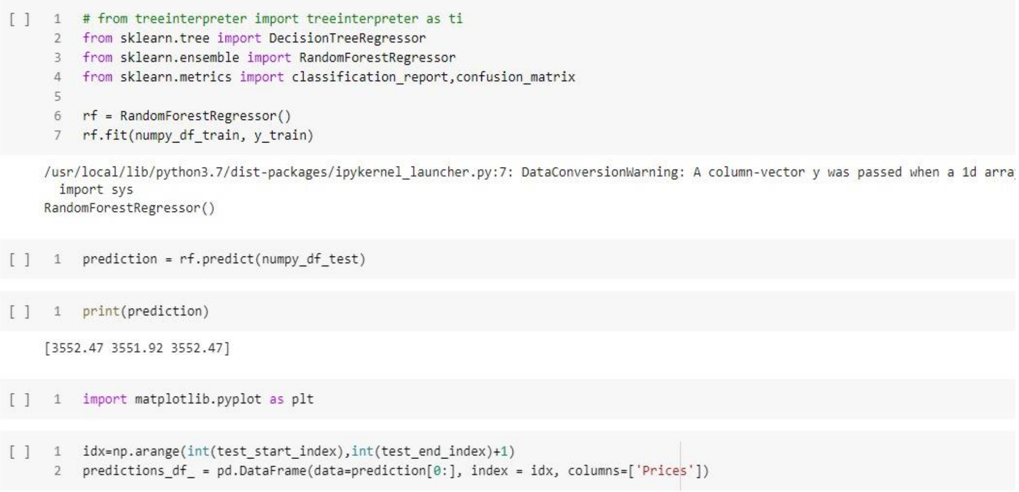
  The data is first fetched from twitter which is raw and has special characters that is cleaned by using the data processing technics. We use regular expressions to clean the data so that it only contains alphanumeric characters, and then we append all the tweets from that day to the data frame. Then we map that to the daily tweets using the CSV file from the Yahoo Finance API.



```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import unicodedata
sentiment_i_a = SentimentIntensityAnalyzer()
for indexx, row in ccdata.T.iteritems():
    try:
        sentence_i = unicodedata.normalize('NFKD', ccdata.loc[indexx, 'Tweets'])
        sentence_sentiment = sentiment_i_a.polarity_scores(sentence_i)
        ccdata['Comp'].iloc[indexx] = sentence_sentiment['compound']
        ccdata['Negative'].iloc[indexx] = sentence_sentiment['neg']
        ccdata['Neutral'].iloc[indexx] = sentence_sentiment['neu']
        ccdata['Positive'].iloc[indexx] = sentence_sentiment['compound']
        # ccdata.set_value(indexx, 'Comp', sentence_sentiment['pos'])
        # ccdata.set_value(indexx, 'Negative', sentence_sentiment['neg'])
        # ccdata.set_value(indexx, 'Neutral', sentence_sentiment['neu'])
        # ccdata.set_value(indexx, 'Positive', sentence_sentiment['pos'])
    except TypeError:
        print (stocks_dataf.loc[indexx, 'Tweets'])
        print (indexx)
```

Fig: Positive/ Negative code

We tried to match the stock's daily price with the daily news we get from Twitter. We utilize the NLTK's vader lexicon sentiment analysis tool after mapping the price and constructing the dataset of the daily closing price with the tweets. Then we calculate each tweet's polarity score and categorize it as positive or negative. We utilize the random forest regressor to classify the tweets, and then we use the supervised learning method, which is the random forest regressor, to train the data. We try to predict prices using the random forest regressor and compare them to the real pricing.

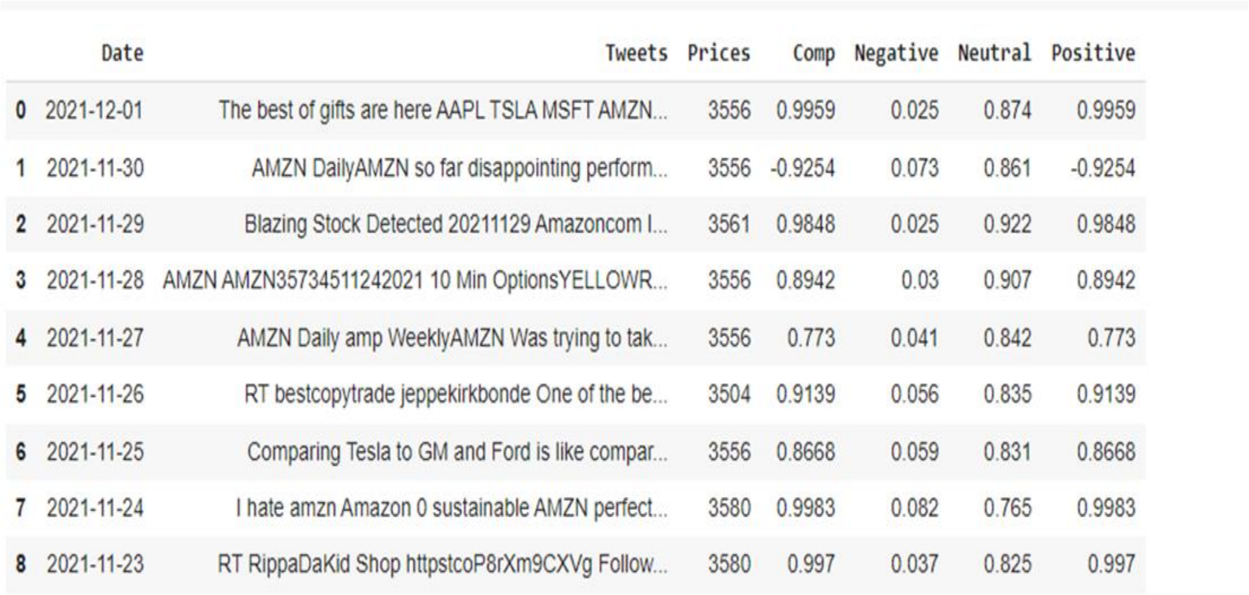


```
[ ]  # from treeinterpreter import treeinterpreter as ti
     from sklearn.tree import DecisionTreeRegressor
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.metrics import classification_report,confusion_matrix

     rf = RandomForestRegressor()
     rf.fit(numpy_df_train, y_train)

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:7: DataConversionWarning: A column-vector y was passed when a 1d arra
  import sys
RandomForestRegressor()

[ ]  prediction = rf.predict(numpy_df_test)

[ ]  print(prediction)

[3552.47 3551.92 3552.47]

[ ]  import matplotlib.pyplot as plt

[ ]  idx=np.arange(int(test_start_index),int(test_end_index)+1)
     predictions_df_ = pd.DataFrame(data=prediction[0:], index = idx, columns=['Prices'])
```

Fig: Random Forest

- **Results**

1)Classifying tweets as positive/negative by using NLTK’s vader lexicon sentiment analysis.



	Date	Tweets	Prices	Comp	Negative	Neutral	Positive
0	2021-12-01	The best of gifts are here AAPL TSLA MSFT AMZN...	3556	0.9959	0.025	0.874	0.9959
1	2021-11-30	AMZN DailyAMZN so far disappointing perform...	3556	-0.9254	0.073	0.861	-0.9254
2	2021-11-29	Blazing Stock Detected 20211129 Amazoncom I...	3561	0.9848	0.025	0.922	0.9848
3	2021-11-28	AMZN AMZN35734511242021 10 Min OptionsYELLOWR...	3556	0.8942	0.03	0.907	0.8942
4	2021-11-27	AMZN Daily amp WeeklyAMZN Was trying to tak...	3556	0.773	0.041	0.842	0.773
5	2021-11-26	RT bestcopytrade jeppekirkbonde One of the be...	3504	0.9139	0.056	0.835	0.9139
6	2021-11-25	Comparing Tesla to GM and Ford is like compar...	3556	0.8668	0.059	0.831	0.8668
7	2021-11-24	I hate amzn Amazon 0 sustainable AMZN perfect...	3580	0.9983	0.082	0.765	0.9983
8	2021-11-23	RT RippaDaKid Shop httpstcoP8rXm9CXVg Follow...	3580	0.997	0.037	0.825	0.997

Fig: Classifying the tweets

2)Pie chart representation showing classification results based on sentimental analysis.

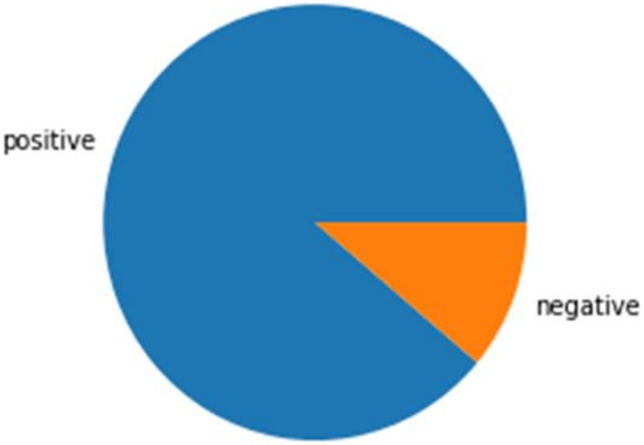


Fig: Classification on sentimental analysis

3)Predicted prices for the next 30 days using random forest generator algorithm.

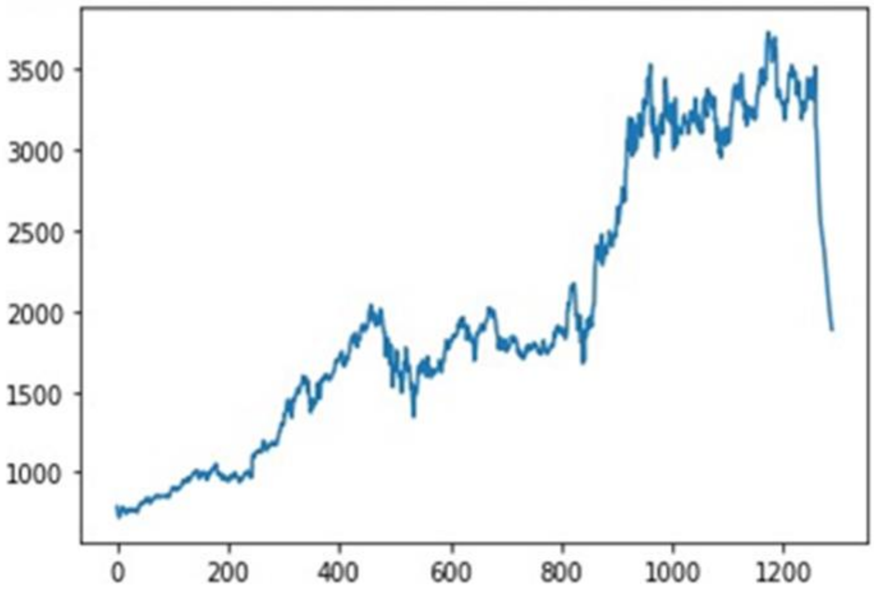


Fig: Prices prediction

4)Comparison between actual price and predicted price

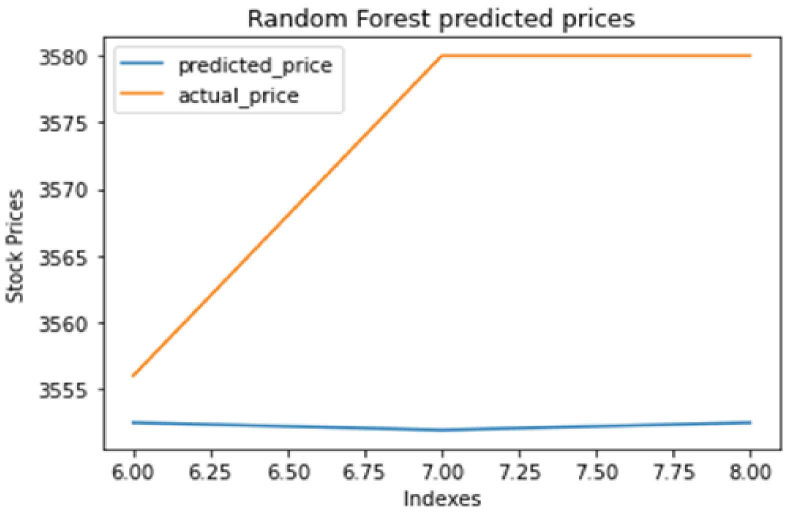


Fig: Price prediction

- **Project Management**
  - **Implementation status report**
    - **Work completed**
    - **Description**:

1. Used random forest regressor algorithm to predict the stock price values for a given period.
2. Used linear regression to compare the predicted prices with actual prices.
3. Classified the data as positive and negative using vader lexicon sentimental analysis.

- **Responsibility (Task, Person)**:
  Background research work: Prathyusha Daroor, Sowrab Sahini
  Designing: Prathyusha Daroor, Guntupalli Naga Srija
  Coding: Sowrab Sahini, Likita Chamakura Suresh, Prathyusha Daroor, Guntupalli Naga Srija.
  Documentation for this phase: Likita Chamakura Suresh, Sowrab Sahini

- **Contributions (members/percentage):**
  Prathyusha Daroor: 25%
  Likita Chamakura Suresh: 25%
  Sowrab Sahini: 25%
  Guntupalli Naga Srija: 25%

Issues/Concerns: We overcame the difficulty that we faced during the increment 1 that is understand and implement the proposed project using deep learning models by doing the proper background research and by having a good teamwork.

- **References/Bibliography**

[1]M. Zu, "NLP for Stock market prediction with reddit data,“ https://web.stanford.edu/class/cs224n/reports/final_reports/report030.pdf.
[2]"U.S security and exchange commission," 9 jan 2017. [Online]. Available: https://www.sec.gov/edgar.shtml.
[3]"TIINGO," [Online]. Available: https://api.tiingo.com/documentation/general/overview.
[4]"Keras LSTM layer," [line]. Available: https://keras.io/api/layers/recurrent_layers/lstm/.
[5]Stock Closing Price Prediction using Machine Learning Techniques Mehar Vijha , Deeksha Chandolab, Vinay Anand Tikkiwalb, Arun Kumar

GitHub Link: https://github.com/sowrab-sahini/SMAP/blob/main/src/Final_Increment/Final_Increment.ipynb