iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** III **Month of publication:** March 2020

DOI:

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

Flask App for Real Time Sentiment Analysis of Tweets on NASDAQ and NSE Stocks

Prof. Amit Narote[1], Kaushik Arvind Jadhav[2], Jay Dharmendra Barot[3], Shubham Santosh Sawant[4]
[1]Assistant Professor, [2, 3, 4] Students, Department of Information Technology, Xavier Institute of Engineering, Mumbai-400016, India

Abstract*: One of the most rapidly expanding research areas in today's world is Sentiment Analysis. Social Media websites like Twitter have become a source of wide variety of information. This is because the people of the modern world prefer blogs or social media pages to post their ideas or opinions about a particular product, service or organization. In this paper, we make use of a popular social media platform, Twitter and implement a system to get insights about the sentiments regarding various NASDAQ (American market) stocks and NSE (Indian market) stocks from the fetched Tweets. Tweets are fetched in Real Time using the Tweepy API and then cleaning is performed to remove unwanted characters from the tweets. TextBlob library is used to assign individual polarities to the tweets and then Overall Polarity for the concerned stock is determined. Flask is used as front end to display the results to the user
Keywords*: Sentiment Analysis, Twitter, Stock Market, Prediction, Stock Exchange, Trading, Tweepy, TextBlob, Python, Tweets, Social Media

I. INTRODUCTION

The stock market of the modern world is very fluctuating in nature. They vary based on various factors such as previous stock prices, present scenario of the market, financial news, rival companies etc. It is important to have accurate prediction of future trends in stock prices for smart investing decisions [1] [2]. However, the fluctuating nature of the stock prices makes it difficult to have a precise estimation. Sentiment Analysis of Real Time Tweets of financial companies is an attempt to estimate the trends in future prices of a company stock [3]. Real Time Tweets for NASDAQ (American market) stocks and NSE (Indian market) stocks were fetched using the Tweepy API. These tweets are preprocessed so as to remove unwanted characters and then passed into the TextBlob library which assigns polarity to individual tweets. Finally, global polarity of the stocks is computed and visualized.

II. LITERATURE REVIEW

A. Machine Learning Techniques and Use of Event Information for Stock Market Prediction

Paul D. Yoo, Maria H. Kim and Tony Jan compared and evaluated some of the existing ML techniques used for stock market prediction. After comparing simple regression, multivariate regression, Neural Networks, Support Vector Machines and Case Based Reasoning models they concluded that Neural Networks offer ability to predict market directions more accurately as compared to other techniques. Support Vector Machines and Case Based Reasoning are also popular for stock market prediction. In addition, they found that incorporating event information with prediction model plays a very important role for more accurate prediction. The web provides the latest and latent event information about stock market which is required to yield higher prediction accuracy and to make prediction in a short time frame [1].

B. Stock Price Prediction Using Financial News Articles

M.I. Yasef Kaya and M. Elef Karshgil analysed the correlation between the contents of financial news articles and the stock prices. News articles were labeled positive or negative depending on their effect on stock market. Instead of using single word as features, they used word couples as features. A word couple consisted of a combination of a noun and a verb. SVM classifier was trained with labeled articles to predict the stock prices. [2].

C. Forecasting of Stock Market Indices Using Artificial Neural Network

Dr. Jay Joshi, Nisarg A Joshi in his work, used artificial neural network (ANN) to predict the stock prices in reputed indexes of Bombay Stock Exchange (BSE) Sensitive Index (Sensex). They conducted experiments and case studies to compare the performance of neural network with random walk and linear autoregressive models. They reported that neural network outperforms linear autoregressive and random walk models by all performance measures in both in-sample and out-of-sample forecasting of daily BSE Sensex returns.The model forecasted the desired target with an average accuracy of 82% The model was efficient in forecasting stock prices with minimum amount of data and yielded valuable insights regarding future trends in stock prices [3].

D. Social Media Monitoring using Sentiment Analysis of Twitter Data

Rupawari Jadhav, M.S. Wakode discussed the prediction of future stock prices using the sentiment values for every stock. The paper focuses on two different techniques, Word2vec and N-gram, for deriving insights from sentiments in tweets. The author made use of sentiment analysis in addition to machine learning algorithms so as to tweets extracted from Twitter and analyzing the correlation between stock market movement of company and sentiments in tweets. It proposes a hybrid approach which combines unsupervised learning to cluster the tweets and then performing supervised learning methods for Classification. The author applied different machine learning techniques: Naive Bayes (NB), Maximum entropy, support vector machine (SVM) etc. and concluded that NB, SVM with 89.4% accuracy outperforms the other techniques in sentiment classification. They looked for a correlation between twitter sentiments with stock prices and determined which words in tweets correlate to change in stock price by doing a post analysis of price change and tweets [4].

III.METHODOLOGY

Sentiment Analysis of stock market tweets is a complex operation and needs to be broken down into a set of relatively simple steps. This section focuses on planning and summarizing the workflow of the project for efficient Sentiment Analysis of tweets and for extracting valuable insights from the collected tweets. Each individual step is a sequence of operations used to process the tweets and to derive useful insights from these tweets. These steps are summarized in the flow chart shown below. The final result is in the form of a Flask application which displays the initial raw tweets that are fetched and the Overall polarity and distribution of positive, negative and neutral sentiments obtained upon processing those tweets.

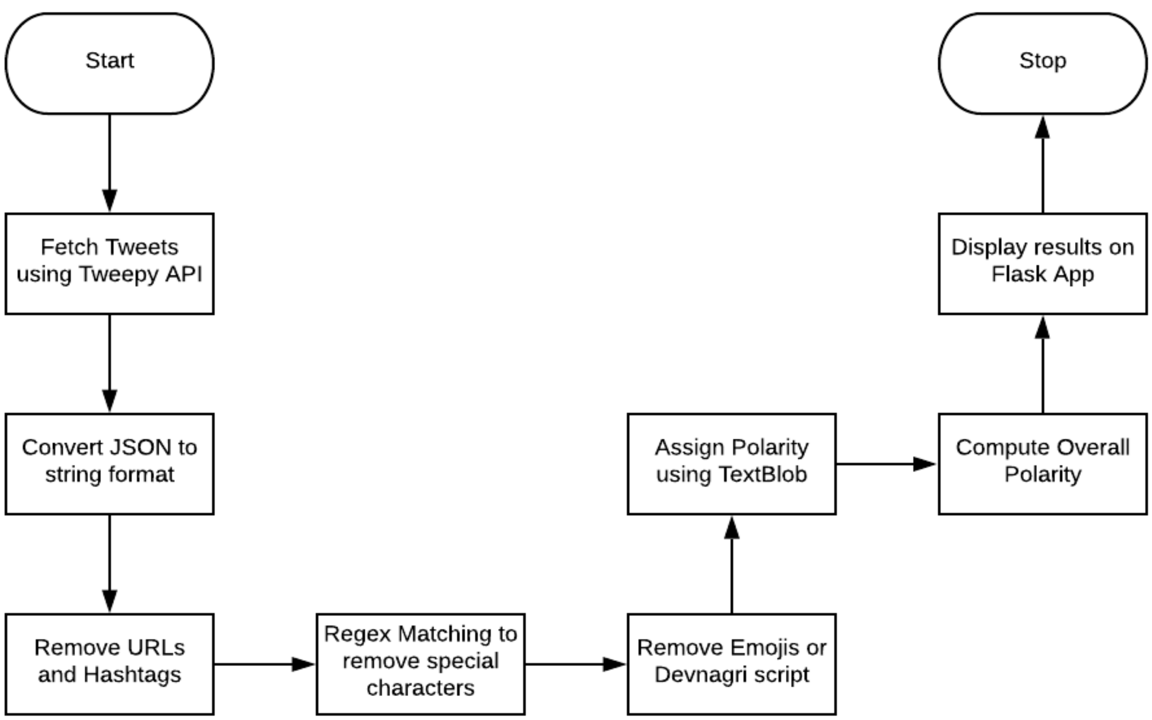


Fig 3.1: Methodology and Process Work Flow Diagram

First, tweets about the concerned NASDAQ and NSE stocks are fetched from Twitter using the Tweepy API. The raw tweets fetched are in JSON format. In order to perform computations on them, they need to be converted into pure string format. These tweets contain several unwanted trash characters such as hashtags, usernames, tags, hyperlinks, etc. These need to be removed by performing Data Cleaning on the tweets. However, the cleaned tweets might still contain special characters like ':' and '&' which need to be removed by Regular Expression Matching. The tweets might also contain Emojis and other non-ASCII characters which need to be removed. Once the tweets are properly processed, TextBlob library is used to assign polarity to individual tweets. This polarity may be a positive or negative integer or it may be zero. The Overall Polarity is computed by taking the sum of individual polarities and finally, the last step involves displaying all the results via the Flask app to the user.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Data Collection

The stock ticker for NASDAQ and NSE stocks is taken as input from the user via the Flask App. For the stock ticker input by the user, 1000 real time tweets are fetched from Twitter by using the Tweepy API. The tweets are fetched and contained in JSON format. In order to crawl the tweets from Twitter, API keys for Twitter app are needed. These API keys along with the number of tweets to be fetched are specified in a separate constants file as shown below. The following code is used for authenticating to Twitter and fetching tweets using Tweepy API.

```
constants.py*

consumer_key= 'E0pFYVai9VaOhqLiRBEC6gpGF'
consumer_secret= 'XAMh4l9XL5nwFK3MN5tAjtXA2YgDN1tw5f7L2n6dz5ib8VYlbm'

access_token='3261604734-86c7DOJP98GwNeFWzvgPQKFUTyHn1ZFwlloJP3v'
access_token_secret='eXEmlEAdxaFjueVP03jsAWeOeNMkI7ToiDQkyvLDa6eX7'

num_of_tweets = int(1000)
```

Fig 4.1 API keys and Configuration for Tweepy

```
auth = tweepy.OAuthHandler(ct.consumer_key, ct.consumer_secret)
auth.set_access_token(ct.access_token, ct.access_token_secret)
user = tweepy.API(auth)

tweets = tweepy.Cursor(user.search, q=str(symbol),
                       tweet_mode='extended',
                       lang='en').items(ct.num_of_tweets)
```

Fig 4.2: Code snippet for Tweepy authentication and fetching tweets

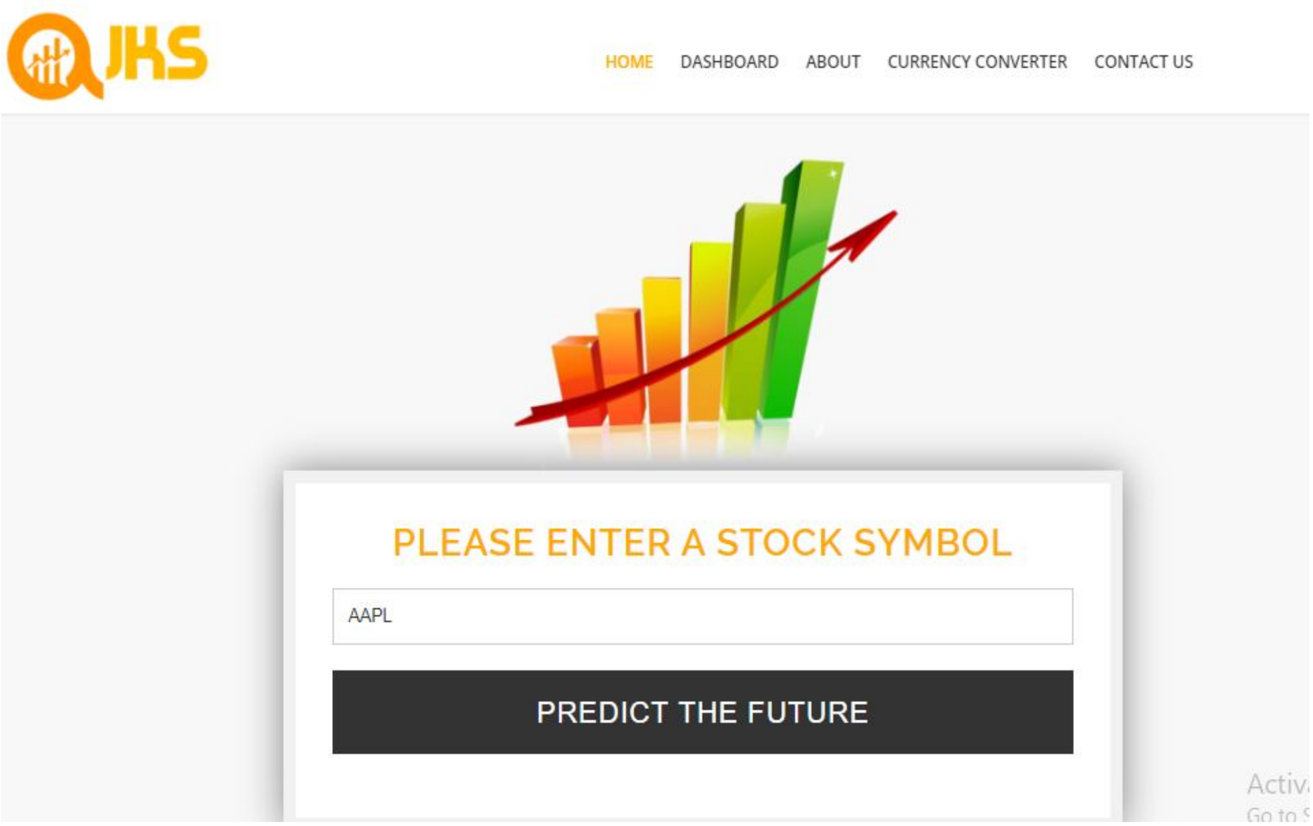


Fig 4.3 Taking Stock ticker as input from user via Flask App

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

B. JSON to String Conversion

The tweets collected from Twitter using the Tweepy API are in JSON format and they need to be converted into string format to perform further computations and processing on it. This is done using the full_text method provided by Tweepy. The Python code to convert the fetched tweets to string format is shown below.

```
for tweet in tweets:
    count=20 #Num of tweets to be displayed on web page
    #Convert to JSON to string format
    tw = tweet.full_text
    blob = TextBlob(tw)
```

Fig 4.4: Code snippet to convert JSON to string format

C. Data Cleaning

The raw tweets in string format contain several unwanted characters which need to be removed. These characters may be in the form of hashtags, usernames, tags or hyperlinks. In order to remove these characters, the tweet-preprocessor library in Python has been used. The Python code used for cleaning the raw tweets is shown in the below code snippet

```
for tweet in tweets:
    count=20 #Num of tweets to be displayed on web page
    #Convert to Textblob format for assigning polarity
    tw2 = tweet.full_text
    tw = tweet.full_text
    #Clean
    tw=p.clean(tw)
```

Fig 4.5: Code snippet for cleaning the tweets

```
------------------------------RAW TWEET-----------------------------
Breaking: President @realDonaldTrump’s number 1 priority is 2 protect the
 Stock Market &amp; will continue 2 lie 2 the American people in response
 2 the threat of Coronavirus □ □ □ □ □□□□□□□□ $SPY $SPX $QQQ $IWM $DJI $A
A $FB $AMZN $JPM $GOOGL  $XLE $BA $BABA $UVXY  #ES_F #coronavirus https:/
/t.co/JW6nseMbVM
-------------------------------CLEANED TWEET------------------------------

Breaking: President ’s number priority is protect the Stock Market &amp;
will continue lie the American people in response the threat of Coronavir
us □ □ □ □□□□□□ $SPY $SPX $QQQ $IWM $DJI $AAPL $TSLA $FB $AMZN $JPM $GOOG
 $BA $BABA $UVXY
```

Fig 4.6:Tweet after cleaning

D. Regular Expression Matching

Even after cleaning the tweets using the tweet-preprocessor library, certain unwanted characters are still present in the tweets. These characters include symbols used in natural language like ‘:’ and ‘&amp’. Such characters are removed by pattern matching using Regular Expressions. The Python code for Regular Expression Matching is shown in the below code snippet.

```
#Replace &amp; by &
tw=re.sub('&amp;','&',tw)
#Remove :
tw=re.sub(':','',tw)
```

Fig 4.7: Code snippet for Regular Expression Matching

```
-------------------------------CLEANED TWEET------------------------------

Breaking: President ’s number priority is protect the Stock Market &amp;
will continue lie the American people in response the threat of Coronavir
us □ □ □ □□□□□□ $SPY $SPX $QQQ $IWM $DJI $AAPL $TSLA $FB $AMZN $JPM $GOOG
 $BA $BABA $UVXY
-------------------------------TWEET AFTER REGEX MATCHING----------------
-------------
Breaking President ’s number priority is protect the Stock Market & will
continue lie the American people in response the threat of Coronavirus □
□ □ □□□□ $SPY $SPX $QQQ $IWM $DJI $AAPL $TSLA $FB $AMZN $JPM $GOOGL $XLE
$BABA $UVXY
```

Fig 4.8: Tweets after Regular Expression Matching (& replaced by &)

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

E. Remove Emoticons or Devnagri Script (Non-ASCII Characters)

The processed tweets might still contain Non-ASCII characters like Emoticons or Devnagri script which provide no real sentiment value. These characters need to be removed by encoding and decoding back the format of your tweets, ASCII in our case. The Python code to remove Non-ASCII characters like Emoticons or Devnagri Script is shown in the below code snippet.

```
#Remove Emojis and Hindi Characters
tw=tw.encode('ascii', 'ignore').decode('ascii')
```

Fig 4.9: Code snippet for removing Non-ASCII characters

```
-------------------------------TWEET AFTER REGEX MATCHING----------------
-------------
Breaking President ’s number priority is protect the Stock Market & will
continue lie the American people in response the threat of Coronavirus □
□ □ □□□□ $SPY $SPX $QQQ $IWM $DJI $AAPL $TSLA $FB $AMZN $JPM $GOOGL $XLE
$BABA $UVXY
-------------------------------TWEET AFTER REMOVING NON ASCII CHARS------
-----------------------
Breaking President s number priority is protect the Stock Market & will c
ontinue lie the American people in response the threat of Coronavirus  $S
PY $SPX $QQQ $IWM $DJI $AAPL $TSLA $FB $AMZN $JPM $GOOGL $XLE $BA $BABA $
UVXY
```

Fig 4.10 Tweets after removing Non-ASCII characters

F. Assign Polarity using Text Blob

The processed tweets are now assigned a polarity using the Textblob library. This polarity is in the form of a positive or negative integer or zero. If it is a positive integer, the sentiment of the tweet is positive. If it is a negative integer, the sentiment of the tweet is negative. If it is zero, the sentiment of the tweet is neutral. The Python code for assigning polarity to individual tweets is shown in the below code snippet.

```
blob = TextBlob(tw)
polarity = 0 #Polarity of single individual tweet
for sentence in blob.sentences:

    polarity += sentence.sentiment.polarity
    if polarity>0:
        pos=pos+1
    if polarity<0:
        neg=neg+1
```

Fig 4.11: Code snippet for assigning polarity using TextBlob

```
##############################################################################

Positive Tweets : 153 Negative Tweets : 47 Neutral Tweets : 0
##############################################################################
```

Fig 4.12: Number of Positive Negative and Neutral Tweets for NASDAQ (AAPL) stock

```
##############################################################################

Positive Tweets : 161 Negative Tweets : 39 Neutral Tweets : 30
##############################################################################
```

Fig 4.13: Number of Positive Negative and Neutral Tweets for NSE (HDFCBANK) stock

G. Compute Overall Polarity

The individual polarities of the individual tweets are summed up and normalised so as to compute the global polarity of all the tweets. This global polarity provides us with the overall sentiment value of the stock ticker input by the user.

```
        global_polarity += sentence.sentiment.polarity
global_polarity = global_polarity / len(tweet_list)
```

Fig 4.14: Code snippet to compute Overall Polarity

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

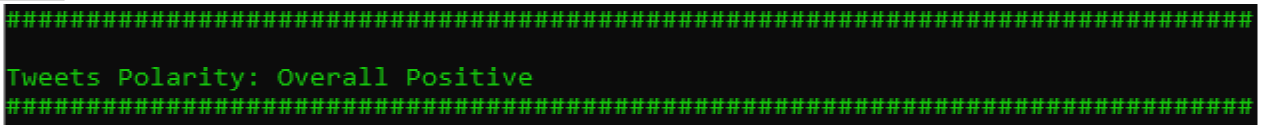


Fig 4.15: Overall Polarity for NASDAQ (AAPL) stock

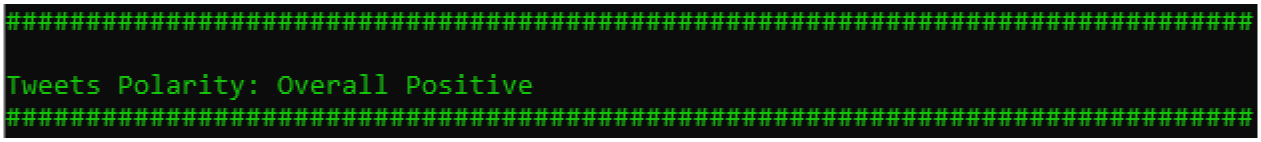


Fig 4.16: Overall Polarity for NSE (HDFCBANK) stock

H. Visualising Results

The Positive, Negative and Neutral tweets for NASDAQ and NSE stocks can be visualised using pie charts for efficient analysis. The Python code to plot pie charts and visualise results is show in the below code snippet.

```
labels=['Positive','Negative','Neutral']
sizes = [pos,neg,neutral]
explode = (0, 0, 0)
fig = plt.figure(figsize=(7.2,4.8),dpi=65)
fig1, ax1 = plt.subplots(figsize=(7.2,4.8),dpi=65)
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
plt.savefig('static/SA.png')
plt.close(fig)
```

Fig 4.17: Code snippet to plot pie charts

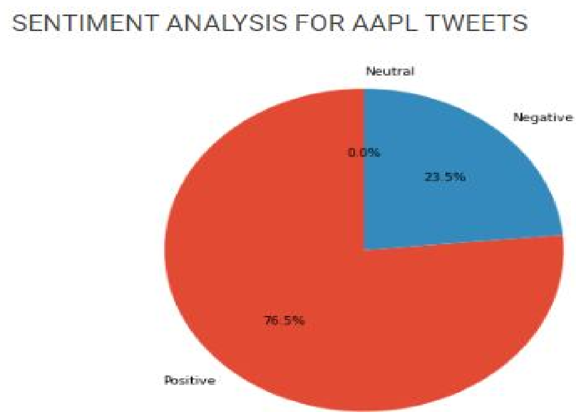


Fig 4.18: Pie chart for NASDAQ (AAPL) stock

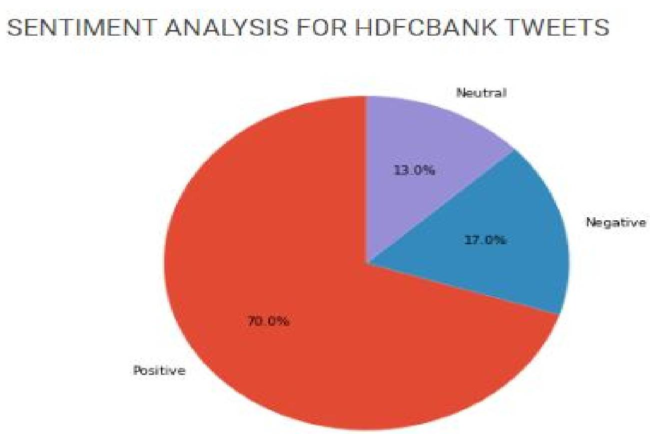


Fig 4.19: Pie chart for NSE (HDFCBANK) stock

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

I. Deploy Results to Flask App

Finally, the unprocessed, JSON tweets fetched, Pie Chart, Overall Polarity and the final recommendation are sent to the Web App to display them to the user. The web app is interfaced with Python using Flask framework which is deployed on the server.

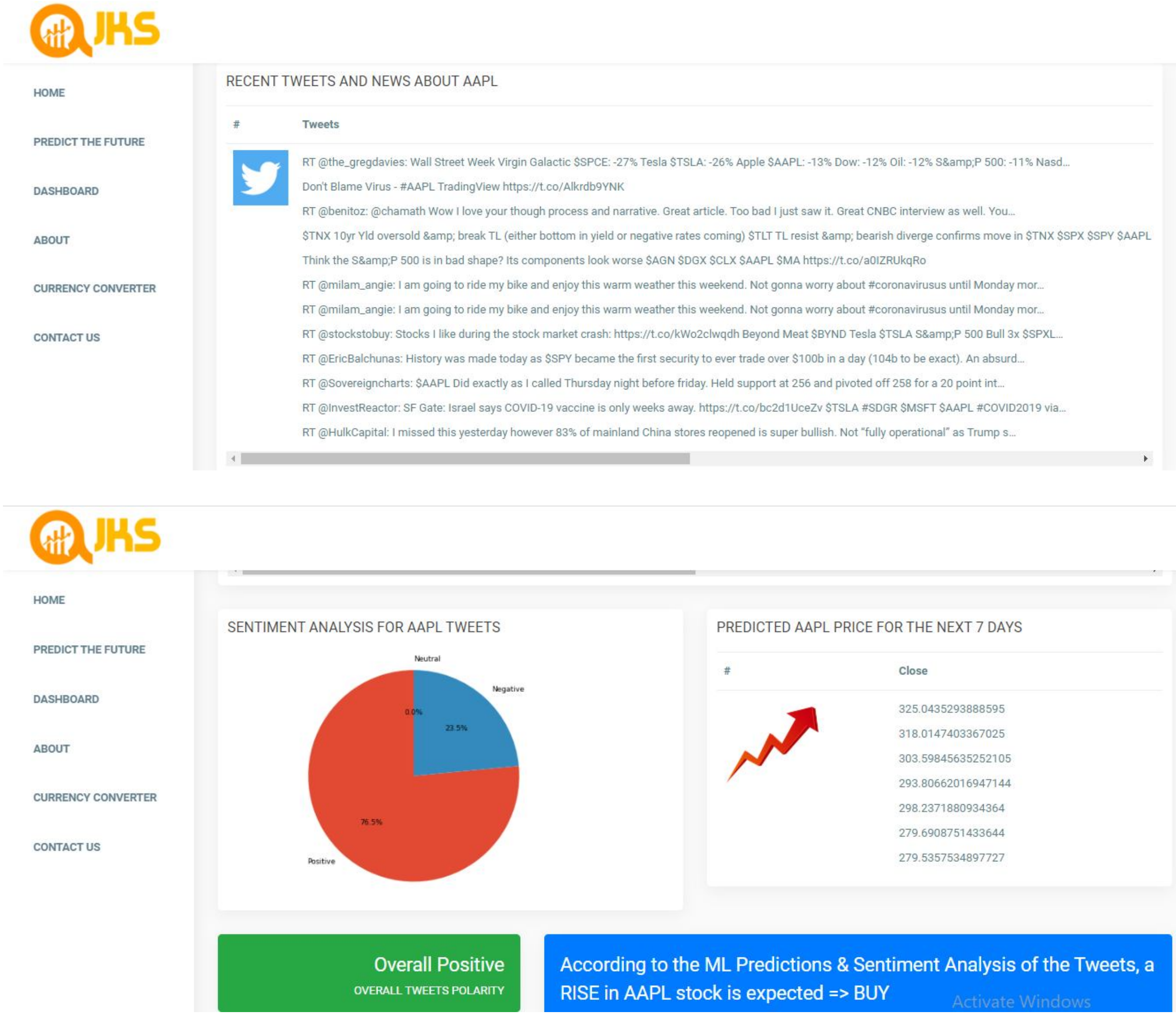
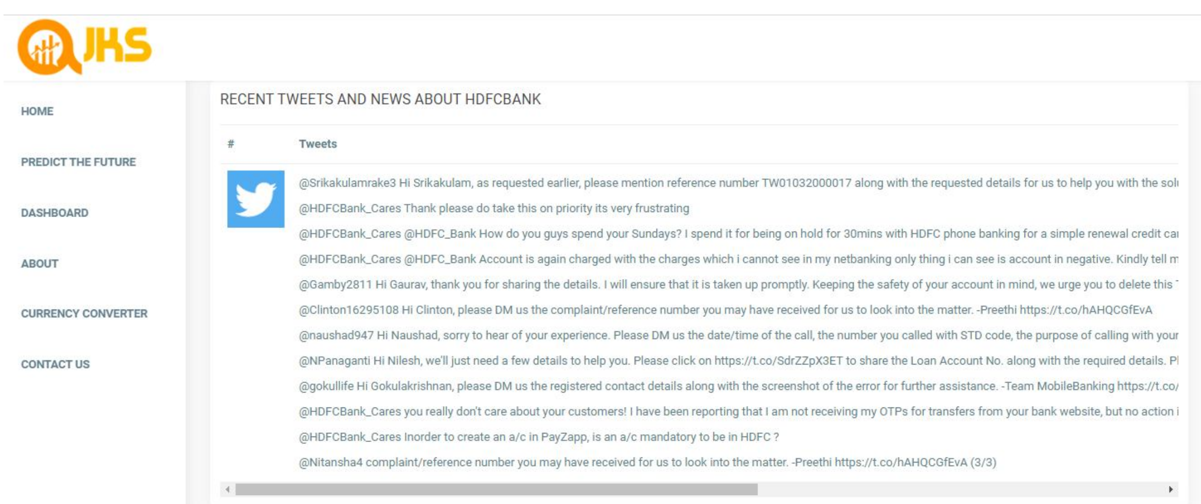


Fig 4.20: Flask App output for NASDAQ (AAPL) stock



International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue III Mar 2020- Available at www.ijraset.com

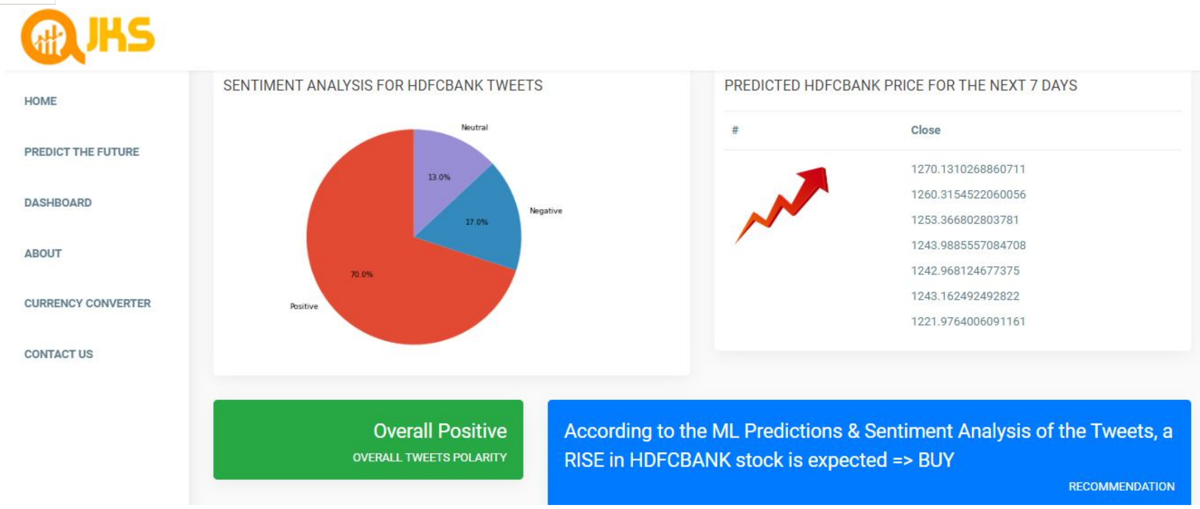


Fig 4.21: Flask App output for NSE (HDFCBANK) stock

V. CONCLUSION

The proposed system worked very well with the financial tweets of NASDAQ and NSE stocks. It focuses on analyzing the relationships between various events that have happened recently and studying their effects on NASDAQ and NSE stock prices. It enables us to gather useful insights and derive meaningful results from the twitter posts and tweets made by people around the world. The computation of overall sentiment value along with the distribution of positive, negative and neutral tweets helps one to get a broader idea about the current market scenario. However, simply fetching and directly applying computations is not enough. The tweets need to be properly processed so as to remove unwanted characters and then taken into account for polarity computations. The large amount of tweets processed enables us to get the sentiments of whether it is feasible or not to invest in a particular stock.

REFERENCES

[1] P. D. Yoo, M. H. Kim and T. Jan, "Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation," International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, 2005, pp. 835-841.

[2] M. İ. Y. Kaya and M. E. Karsligil, "Stock price prediction using financial news articles," 2010 2nd IEEE International Conference on Information and Financial Engineering, Chongqing, 2010, pp. 478-482.

[3] Hedayati, Amin & Moghaddam, Moein & Esfandyari, Morteza. (2016). Stock market index prediction using artificial neural network:. Journal of Economics, Finance and Administrative Science. 10.1016/j.jefas.2016.07.002.

[4] Rupawari Jadhav, M. S. Wakode, “Survey : Sentiment Analysis of Twitter Data for Stock Market Prediction”, March 2017, IJARCCE Volume-6, Issue-3 [Online]. Available: https://ijarcce.com/upload/2017/march-17/IJARCCE%20129.pdf

[5] V. S. Pagolu, K. N. Reddy, G. Panda and B. Majhi, "Sentiment analysis of Twitter data for predicting stock market movements," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, 2016, pp. 1345-1350.