INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 1)
Available online at: www.ijariit.com

Stock prediction marketing using machine learning

Rupali Borole
rupaliborole642@gmail.com
Shivajirao S. Jondhale College of Engineering,
Thane, Maharashtra

Sonali Govilkar
sonaligovilkar92@gmail.com
Shivajirao S. Jondhale College of Engineering,
Thane, Maharashtra

Dipali Duble
dipali.duble11@gmail.com
Shivajirao S. Jondhale College of Engineering,
Thane, Maharashtra

Manisha Sonawane
manisha2810@gmail.com
Shivajirao S. Jondhale College of Engineering,
Thane, Maharashtra

ABSTRACT

Share market has traditionally been the proving grounds for machine learning applications. There is a lack of an algorithm which can find the heuristic reasoning of humans based on current events/trends. The proposed system is an attempt to reconcile computed sentiments alongside traditional/more common data mining. Datasets consisting of historical data as well as recent headlines will be mined to ascertain stock price movement. In the proposed system it is to be hoped, more accurately predict stock price movement by emulating instinctual reasoning by implementing sentiment analysis.

Keywords— *Sentiment analysis, Stock market, Machine learning, Regression, AFFIN algorithm*

1. INTRODUCTION

In the business sector, it has always been a difficult task to predict the exact daily expense of the stock market index. Therefore, there is a great deal of research being conducted regarding the prediction of the direction of the stock price index movement. Many factors as political events, general economic conditions, and trader's expectations may have an influence on the stock market index. The successful prediction of a stock's future price could yield a significant profit to a large group of people as well as independent organizations.

1.1 Introduction to existing system

Sentdex is a sentiment analysis algorithm, which is termed by the meshing of "sentiment" and "index". It understands the people's emotions which are used in their online communication and then it is translated into computer language. This data can be used to gain a deeper understanding of the world.

At its heart, Sentdex is a bot which reads the news like you or me. As Sentdex reads the articles, it checks what are known as "Named Entities" through a natural language process called "named entity recognition." Once Sentdex has decided that what is an article, paragraph, or even just a sentence talking about, it just then look for opinions. This has been a more challenging part of using Natural Language Processing to derive sentiment from written-language. At the core, this process involves a lot of what is called "chunking" to group bits of text into noun-phrases, which contain adjectives and adverbs, along with some other information about what is being said.

From here, Sentdex is able to decide, which mainly gives the adjectives whether or not the author of the text has a positive attitude or a negative attitude towards the subject, or the Named Entity, in question.

2. PROPOSED SYSTEM

Our system basically involves the interface of real-time comments from Twitter and other valuable sources and data from these sources are then analyzed with google sentiment analysis and IBM Watson tone analyser and with the help of neural network and dataset from yahoo finance we can do a couple of real-time dynamics stock market analysis

3. SYSTEM ARCHITECTURE

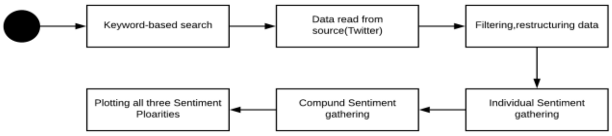

Keyword-based search
Data read from source(Twitter)
Filtering,restructuring data
Individual Sentiment gathering
Compund Sentiment gathering
Plotting all three Sentiment Ploarities

Fig. 1: System Architecture

The model starts with an input keyword. The user could simply add the desired company name as the keyword, for instance, GOOGLE" would fetch tweets with the word 'Google' in them. Tweets normally have a bunch of arbitrary elements such as emoji, symbols and numbers as well in our case. So we filter proper sentences out of the tweets using a regular expression. This brings more accuracy to the predictions and reduces neutral prediction cases. These tweets are then passed through a module called ‘textblob’. Textblob is basically a lexicon-based polarity assignment classifier. Based on its own dictionary, it assigns negative statements with a negative polarity and positive statements with a positive polarity. We make use of these assignments to predict the individual tweet sentiments. These individual sentiments are then added together to produce a resultant compound sentiment for the input keyword. These polarities are plotted on a graph for the user to compute the overall sentiment and how it's performing in the market.

4. IMPLEMENTATION

The stock market is the most dynamic environment and may tool are being generated to predict it, but predicting usually involves handling creating design patterns from the previous data of last 10 years and that involves a large analysis of data and storage of data with business patterns. But there are always certain parameters and circumstances which affects the market overnight and can never be caught in data analysis which are usually termed as political change, product fall, ban on products which directly affects the stock market and analysis dependent upon that is still not exists. Our system basically involves the interface of real-time comments from Twitter and other valuable sources and data from these sources are then analyzed with google sentiment analysis and IBM Watson tone analyser and with the help of neural network and data set from yahoo finance we can do a couple of real-time dynamics stock market analysis.

4.1 AFFIN Algorithm

AFINN is a list of words rated for valence with an integer between minus five (negative) and plus five (positive). Sentiment analysis is performed by cross-checking the string token s(words, emoji’s) with the help of the AFINN list and getting their respective scores. The comparative score is simply will be: the sum of each token/number of tokens. So, for example, let's take the following are:

I love cats, but I am allergic to them**.**

That string results in the following:

```
{
   score: 1,
   comparative: 0.1111111111111111,
   tokens: [
      'i',
      'love',
      'cats',
      'but',
      'i',
      'am',
      'allergic',
      'to',
      'them'
   ],
   words: [
      'allergic',
      'love'
   ],
   positive: [
      'love'
   ],
   negative: [
      'allergic'
   ]
}
```

Returned Objects

In this case, we have observed that love has a value of 3, allergic has a value of -2, and the remaining tokens are neutral with a value of 0. As the string has 9 tokens the resulting comparative score looks like: (3 + -2) / 9 = 0.111111111

This approach leaves you with a mid-point of 0 and the upper and lower bounds are constrained to positive and negative 5 respectively (the same as each token!). For example, let's assume that "positive" string with 200 tokens and where each token has an AFINN score of 5. Our result score would look like this:

(max positive score * number of tokens) / number of tokens

(5 * 200) / 200 = 5

4.2 Quandl

Quandl is the premier source for financial, economic, alternative data sets, and serving investment professionals. Quandl’s platform has been used by over 250,000 people, including analysts from the world’s top hedge funds, asset managers and investment banks. There are two methods for retrieving data in Python are the Quick method and the detailed method. The latter is more suitable for application programming. Both methods work with Quandl's two types of data structures are: time-series (dataset) data and non-time series (data table).

The following quick call can be used to retrieve a dataset are:

```
import quandl
data = quandl.get('NSE/OIL')
```

This example finds all data points for the dataset NSE/OIL and stores them in a pandas data frame.

A similar quick call can be used to retrieve a data table are:

```
import quandl
data = quandl.get_table('ZACKS/FC', ticker='AAPL')
```

The above example retrieves all rows for ZACKS/FC where ticker='AAPL' and stores them in a pandas data frame.

4.3 TextBlob

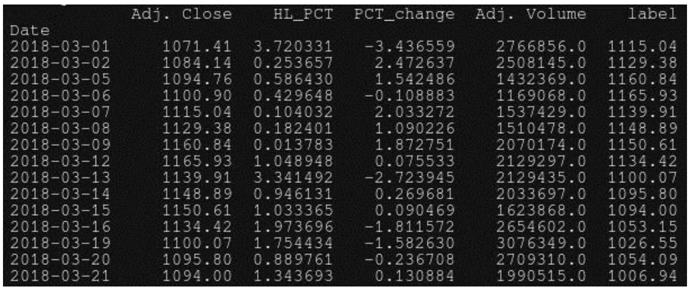
TextBlob is a Python library which is used for processing textual data. It will provide a simple API for diving into common natural language processing tasks like part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

NLTK is a leading platform which is used for building Python programs to work with human language data. It will provide easy-to-use for interfaces to over 50 corpora and lexical resources such that WordNet, along with a suite of text processing libraries which will be used for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, and wrappers for industrial-strength NLP libraries, and an active discussion forum.

5. RESULT

(a) Run the main.py file

(b) First, the Machine Learning part of the model will start running and the dataset of the historic values and predictions based on that appear on the output console as below.

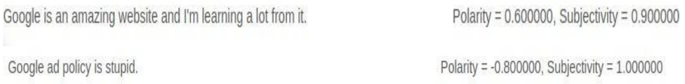


```
            Adj. Close   HL_PCT  PCT_change  Adj. Volume    label
Date
2018-03-01     1071.41 3.720331   -3.436559    2766856.0  1115.04
2018-03-02     1084.14 0.253657    2.472637    2508145.0  1129.38
2018-03-05     1094.76 0.586430    1.542486    1432369.0  1160.84
2018-03-06     1100.90 0.429648   -0.108883    1169068.0  1165.93
2018-03-07     1115.04 0.104032    2.033272    1537429.0  1139.91
2018-03-08     1129.38 0.182401    1.090226    1510478.0  1148.89
2018-03-09     1160.84 0.013783    1.872751    2070174.0  1150.61
2018-03-12     1165.93 1.048948    0.075533    2129297.0  1134.42
2018-03-13     1139.91 3.341492   -2.723945    2129435.0  1100.07
2018-03-14     1148.89 0.946131    0.269681    2033697.0  1095.80
2018-03-15     1150.61 1.033365    0.090469    1623868.0  1094.00
2018-03-16     1134.42 1.973696   -1.811572    2654602.0  1053.15
2018-03-19     1100.07 1.754434   -1.582630    3076349.0  1026.55
2018-03-20     1095.80 0.889761   -0.236708    2709310.0  1054.09
2018-03-21     1094.00 1.343693    0.130884    1990515.0  1006.94
```

Fig. 2: Qandl Result Data Sets

This dataset is the stock of Google from 1st March 2018 till 21st March 2018. The label column depicts the predicted output just from the historical stock prices of Google.

(c) After the historical prediction is made, tweets related to Google start loading in on the output console and it shows the polarity of the tweet between values -1 and 1 as below



```
Google is an amazing website and I'm learning a lot from it.          Polarity = 0.600000, Subjectivity = 0.900000

Google ad policy is stupid.                                          Polarity = -0.800000, Subjectivity = 1.000000
```

Fig. 3: Textblob sentiment analysis performance

(d) When both of the module processing is done, a single screen plot of both the predictions appear (graphically) as below

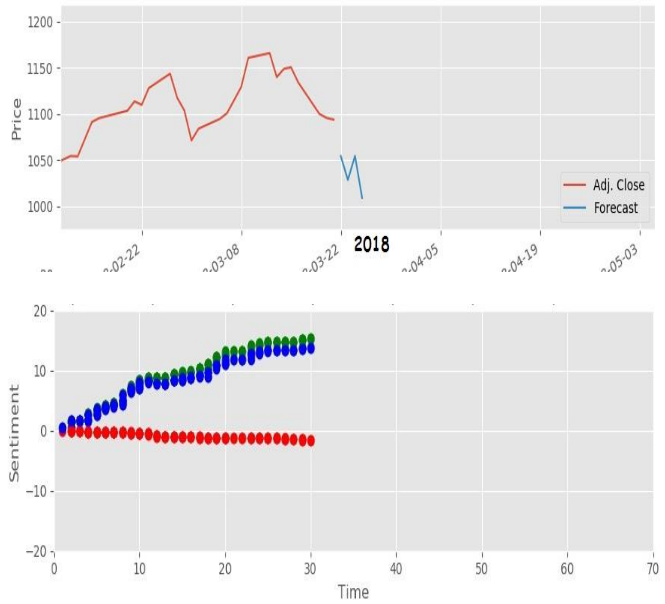


Fig. 4: Compound Sentiment Plots

6. FUTURE SCOPE

Going forward, the system would heavily benefit from fine-tuning the sentiment target to improve the accuracy and value of sentiment analysis to the process. In addition, it would be highly beneficial to improve upon the historical dataset's availability to smooth over even Quandl's minor network/key issues to minimize data unavailability even more. To round things out, the implementation of a neural network at the end of the program to accept the binary outputs of machine learning and sentiment analysis to generate an even more human-like response to said stimuli would do well to hammer in the point of bringing human common sense to the cold hard logical approach of the traditional machine learning system. Aside from these direct improvements, the system could also use better internet connection and processing power to improve upon its already remarkable runtime. Efforts should be made to incorporate alternative dataset sources to test out their feasibility, for example, Google/Yahoo Finance News, instead of Twitter. If the Neural Network is implemented, testing various combinations of layer sizes would go a long way in determining the best possible system while also looking out for the risk of overfitting.

7. CONCLUSIONS

The proposed algorithm worked on the views opinions of their viewers on the shares. Stock Market is such a field where views of the users will matter. The views of the experts have affected a lot to the traders those who want to enter into the market. The unsupervised and supervised learning dependent methods help to find the results in a better way. The combinational study is done to get better accuracy, further their optimizations can be done in sequence to get improved results.

8. REFERENCES

[1] J. Bollen and H. Mao, “Twitter mood as a stock market predictor”. IEEE Computer, 44(10):91–94.

[2] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines.” ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

[3] G. P. Gang Leng and T. M. McGinnity, “An on-line algorithm for creating self-organizing fuzzy neural networks.” Neural Networks, 17(10):1477–1493.

[4] A. Lapedes and R. Farber, “Nonlinear signal processing using neural network: Prediction and system modeling.” In Los Alamos National Lab Technical Report.

[5] E. Stefano Baccianella and F. Sebastiani, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining.” In LREC. LREC.

[6] https://finance.yahoo.com/quote/GOOG/history?p=GOOG/

[7] https://finance.google.com/finance/market_news

[8] www.dataschool.io/comparing-supervised-learning-algorithms/

[9] https://www.dezyre.com/data-science-programming-tutorial/support-vector-machine-tutorial/