Stock Movement Prediction Using Machine Learning on News Articles

B R Ritesh
Department of Computer Science and Engineering
BMS College of Engineering
Bangalore, India
riteshritu1995@gmail.com

Chethan R
Department of Computer Science and Engineering
BMS College of Engineering
Bangalore, India
rchethan001@gmail.com

Harsh S Jani
Department of Computer Science and Engineering
BMS College of Engineering
Bangalore, India
harsh.hsj@gmail.com

Abstract—Stock market prediction involves determining the movement of stock prices. It has a wide range of applications including finance and determining the state of the economy. In our work, we aim to present a way by which we can relatively accurately predict whether a stock's price will increase or decrease on a day to day basis. For the same, we use news articles and give the predictions on a daily basis. We concentrate on two aspects: (1) Daily prediction of a stock's price; and (2) A system which provides a relatively high return on investment using our prediction of stocks.

***Keywords*-**SVM, NLP, Machine Learning, Finance, Pattern Recognition, Simulation*.*

I. INTRODUCTION

Stock Market is always evolving and it is very important to keep up with the latest trends. Day trading in stocks is quite risky, more so if you are untrained. Trading with stocks is definitely a tricky process because a stock's value keeps on changing continuously. In such situations, one would like to find out whether they should buy a stock or sell one. Stock analysis will surely give a good idea on the same.

Earlier, only numerical data was used in the prediction of changes in stock prices. Prediction using numerical data is highly inaccurate and of extremely low technical quality and does not give a proper clarity on whether a stock will move up or down. A more accurate way to go about this is through the use of fundamental analysis [1] i.e. to factor in the real economy into the predictions.

One way of predicting the movement of stock market is by using the news articles. News articles about a particular company usually give a fair idea of how that company is performing and also what will happen to the shares of the same. As more financial data is available to people, at a much faster pace, it is plausible to utilize it into fundamental analysis for the prediction of changes in stock prices.

Different methods have been implemented to utilize these articles. One of the commonly used methods is the sentiment analysis of these articles and their correlation in the behaviour of stock prices[2]. This when coupled with simulation will tend to give higher returns for a given stock when compared to the normal and traditional methods.

Hence, in this paper, we describe how the news articles are being used and how they can give an efficient analysis in predicting the stock movements. Different algorithms and working models have been used to predict the stock movements.

II. RELATED WORK

The indicators of changes or trends in stock prices include textual data comprising of newspaper articles and numerical data comprising of previous stock prices. Over the past years, the indicators from numerical data have been extensively analyzed to develop more accurate means of predicting the movement of stock prices. The indicators from textual data have been comparatively underutilized[3].

Gidofalvi, in 2001, proposed the idea of using textual indicators (newspaper articles) [4] to predict the change in the price of a stock. He assigned three movement classes- up, down and unchanged. He used a Naive

Bayesian text classifier to extract the indicators. Given a news article, d, the probability that a movement class c follows is given by:

$$\square(\square = \square|\square) = \frac{\square(\square|\square=\square).\square(\square=\square)}{\square(\square)} \quad (1)$$

Kari Lee and Ryan Timmons in 2007 trained predictors to simulate stock trading using a Bag of Words algorithm and a Maximum Entropy algorithm[5]. The evaluation was based on the results of two trading systems:

1. Simulated trading based on news articles predictions where each day a news article referenced at least one company in question, the "money" for that day was divided among the stocks referenced.

2. Baseline trading – all the stocks on the list received an equal share at the beginning of the month, held for the entire period, and sold at the end regardless of any change in price.

Qicheng Ma in 2008 classified the stock movement prediction as a NLP classification task[6]. A Naive Bayes classifier and a Maximum Entropy classifier was being used. For every word occurred in d, where d is restricted to a paragraph, the probability of a conditional class P(c|d) is given as:

$$\square(\square|\square) = \prod_{\square\in\square} \square(\square|\square)\square(\square) \quad (2)$$

Kalyani Joshi, Prof. Bharathi H. N and Prof. Jyothi Rao also did a prediction of stock movement prediction using news articles[7]. Different techniques such as Naive Bayes Classifier, Random Forest algorithm, sentiment analysis and SVM were being used.

The test data was utilized by three classification algorithms- Naive Bayes Classifier, Support Vector Machine and Random Forest Classifier. All three classifiers were implemented for various degrees of cross validation, data splitting as well as including new test data. Predictably, the Naive Bayes method obtained the weakest degree of classification accuracy at 75% accuracy on new data. The Random Forest method performed slightly better at 80%. Support Vector Machine proved to be the most accurate classifier for new test data, achieving 90%.

III. SYSTEM DESIGN

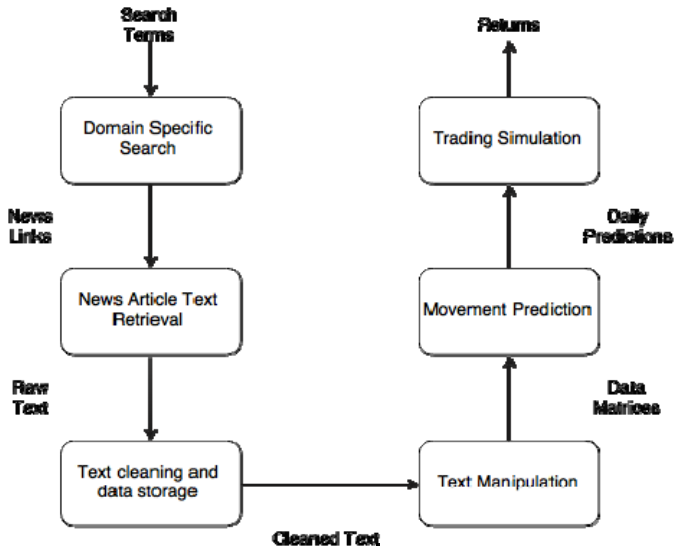


Fig. 1. High level design of the systems used.

The block diagram of our system used for predicting stock movements and trading simulation is shown in Fig. 1. above. The different blocks in the system are:

1. **Domain Specific Search**: The first system is the Domain-specific Search system. This is actually just the system which is used by each specific web domain from which we obtain our news articles from. For our research, we used NASDAQ for the most part. The benefit of choosing a specific web domain is that all the articles are of a particular structure. This makes the process of cleaning data much easier in comparison to extracting data from multiple sources. Moreover, using this domain in particular is beneficial to the goal as the articles themselves are of a more scientific nature when it comes to the stock market rather than most other news sources which may contain heavily unrelated data Over on NASDAQ, the articles for each stock are conveniently placed on that particular stock's page. This makes retrieval of relevant data extremely easy. The data available on NASDAQ dates back approximately six months, which was satisfactory for our purposes.

2. **News Article Text Retrieval**: After choosing NASDAQ as our domain to obtain the news articles for our predictions, we extracted the news article links for each company. A python library called BeautifulSoup serves the purpose. Then, we traversed each page of the website that contains the news article links for that given company and for each link. The data we extracted from each page was- the title, the date and the article's text itself.
3. **Text Cleaning and Data Storage**: From the raw textual data mined, string replacements were applied to clean up unnecessary text or special symbols. Page advertisements were selectively removed. The storage format used for the text was comma separated values. The reason behind this is that the .csv format is extremely easy to read and write to as well as manipulate. Also, reading over thousands of articles, using a simple format such as that lets the algorithm run faster.
4. **Text Manipulation**: The next step we followed was text manipulation. Here, we applied an algorithm known as the Stemming algorithm. Stemming refers to a process in which the tail of some of the words is chopped out so that we get only the stem or the base form of the given word. A simple example would be If we have two words, Discouraging and Discourage, then both the words would be stemmed down to Discourag using the stemming algorithm. The reason for the same is explained in the next paragraph. Next, we used the TF-IDF algorithm.TF-IDF stands for term frequency - inverse document frequency. So, those words that are frequently repeated in the new articles like "the", "is", "of", etc. will have less or no weightage while those that are not frequent will have higher weightage. Also, words like discouraging and discourage mean the same and must have the same weightage given to them. Hence, the stemming process precedes the TFIDF process.
5. **Movement Prediction**: Once the text cleaning and manipulation was done, we partitioned the processed and cleaned news articles for the given company into training and testing portions respectively. Then, we fed the same into different classifiers, namely Naive Bayes classifier, Random Forest Classifier, Perceptrons(Artificial Neural Networks) and SVM(Support Vector Machine). We then found out the accuracies for the different classifiers. For each set of predictions on a daily basis, the algorithm used a voting procedure to generate the close-value prediction for that particular day. If on a particular day, predictions using a set of 11 articles generated 8 upward movement predictions and 3 downward movement predictions, through voting, the final prediction for the day used would be the mode value.
6. **Trading Simulation**: For the second objective of our research, we simulated day-to-day trading of stocks using the predictions we obtained using the classifiers earlier. The core components of the simulation were- the strategy itself, the predictions from the classifiers, and the trading system itself. The objective of the simulation was to maximize return on investment. We implemented it for a single security at a time rather than bunch together multiple ones. We used this to generate graphs detailing the comparisons between changes in security value compared to our own portfolio.

We considered 5 major companies for our research. The reason for doing so was due to the abundant news articles as well as data available for those in particular. The five stocks we used were:

1. Apple
2. Amazon
3. Google
4. Microsoft
5. Tesla

The stock data for each of these was obtained by mining the Google Finance page for each one of them.

IV. CLASSIFIERS

For classification purposes, we used four major classifiers:

Naive Bayes - The majority of the previous literature used it, so we decided on using it as well because of how well known it is as well as to provide a benchmark to compare the classification power of other supervised learning algorithms. For

Naive Bayes, we made what's called the Naive Bayes assumption - we assumed that all the xi's are conditionally independent given y.

Therefore,

p(x1, . . . , xn)

=p(xl|y) · p(x2|xl, y). . . p(xn|xl, . . . , xn−l, y) =p(xl|y) · p(x2|y). . . p(xn|y)

=∏□□=□ □(□ □|□)

Now, to figure out whether the stock price will go up (y = l) or go down (y = −l), we will calculate the probability

p(y = 1|x)

$$=\frac{\square(\square|\square=\square)\square(\square=\square)}{\square(\square)}$$

$$=\frac{(\prod_{\square=\square}^{\square}\ \square(\square\ \ \square|\square=\square))\square(\square=\square)}{\prod_{\square=\square}^{\square}\ \square(\square\ \ \square|\square=\square)\square(\square=\square)+\prod_{\square=\square}^{\square}\ \square(\square\ \ \square|\square=-\square)\square(\square=-\square)}$$

Then, we will calculate the same probability for stock price going down, i.e. $\mathbf{p(y = -1|x)}$. Now, we can classify whether **y = 1** or $\mathbf{y = -1}$ based on which one has a higher probability.

Artificial Neural Networks - We used perceptron based artificial neural networks as our second classifier. The number of iterations over the initial training data was set to 5 as a default value and multiple values were tried. Perceptrons are useful for learning a binary classifier; exactly something we needed for our purposes. The mathematical representation of this is defined as follows:

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

In our case, the output of the perceptron classifier was set to a 0 or 1. The 0 represented a downwards movement whereas a 1 corresponded to an upwards movement.

Support Vector Machine - We also applied the universal learner, the support vector machine. The reason for doing so was to see if there existed a nonlinear relationship between the text documents themselves. Support Vector Machines are based on the *Structural Risk Minimization* principle from computational learning theory[8]. We used an SVM classifier with a Linear kernel.. A particularly useful property of SVMs is that they have the ability to learn independently of the dimensionality of the feature space[9]. The mathematical representation of the Linear SVM kernel can be written as:

Linear: $(x.x')$.

Random Forest Classifier - The random forest is an ensemble approach that can also be thought of as a form of nearest neighbor predictor. Random Forest works on the concept of decision trees. A decision tree has a input at the top and as we traverse down, the data gets split into smaller sets. Each tree has 'n' nodes. In random forest, when a new input is entered into the system, all the trees are generated. The result may either be an average or weighted average of all of the terminal nodes that are reached, or, in some cases, a voting majority.

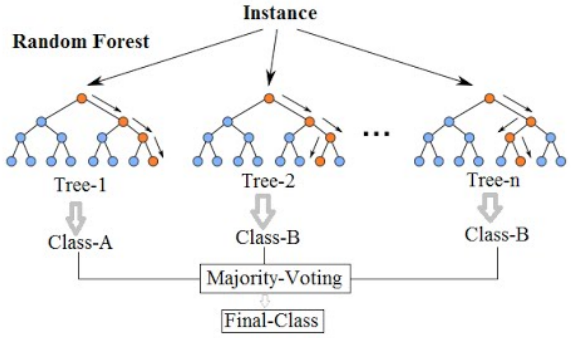
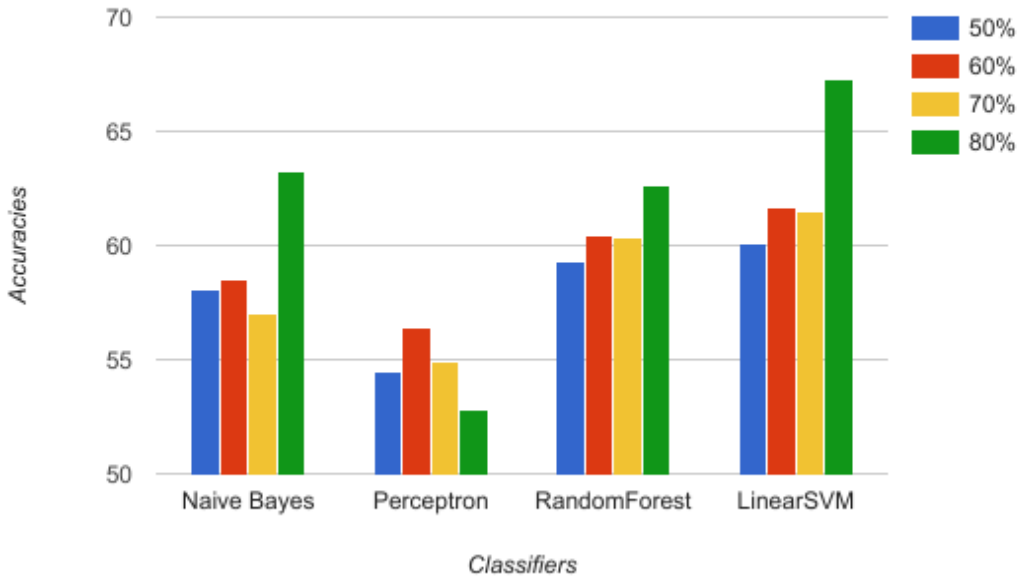


Fig. 2. Graphical Representation of Random Forests

V. IMPLEMENTATION

The text which we mine from *NASDAQ* is categorized per security and store in their respective files. For each stock, we stem the words in all of the articles. Afterwards, for every stock, we apply TF-IDF on the set of all documents. This gives us a sparse matrix containing the TF-IDF values for every single word in each document. This sparse matrix, along with the stock movement value for that article's date is fed as the input data for each of the classifiers. *NASDAQ* is open only for five days a week: Monday to Friday. So if an article comes out on the weekend, we only consider it for the coming Monday. Each stock movement was calculated as the difference between the closing value of the previous day and the closing value of the current day. If it was a weekend, then on the Monday, the movement was calculated using previous Friday's closing value. While dividing our data into training and testing sets, we were extremely careful in only selecting testing data with dates after the end of the training data. This was done in order to prevent the usage of future data for training. Using future data for training and then applying it against a testing set of data which was present chronologically would be illogical for our purposes. We tried four different ratios of training data to testing data- 50%, 60%, 70% and 80%.



. Fig. 3 Results of prediction accuracies of classifiers with different training to testing split ratios

As we can see from the graph above, the Linear SVM classifier dominated the accuracy results. It was closely followed by the Random Forest classifier, then the Naive Bayes classifier and finally, Perceptrons in an ANN. A total of 5653 articles were considered, based on these 5 stocks: Apple, Amazon, Google, Microsoft and Tesla. As we can see, when the training to testing dataset split ratio was increased, the accuracy generally increased. Most noticeably, there were spikes from 70% to 80%. One particularly interesting observation was the constant low performance of the perceptron classifier. Some of the reasons for this could be inefficient weight initialization, overfitting as well as insufficient amount of iterations over the first layer. We were quite satisfied with the results obtained.

VI. TRADING SIMULATION

The final part of our research involved simulating trading sessions with the purchasing and selling of stocks based on our predictions for such. A simple block diagram describing the process is displayed below.

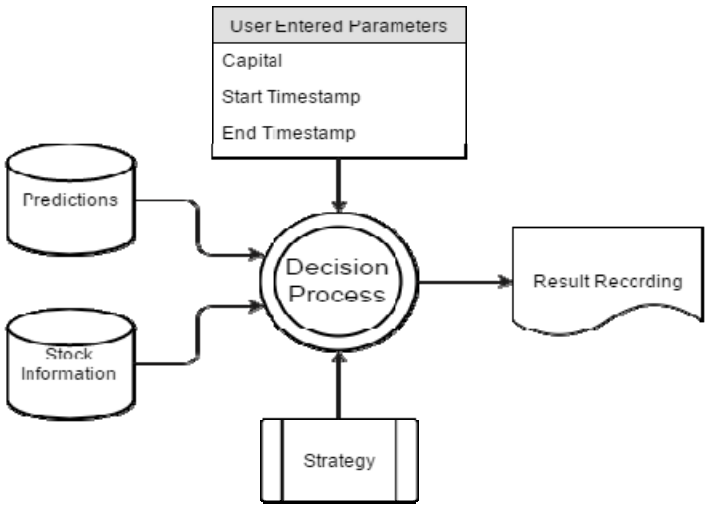


Fig 4. Block diagram of Trading Simulation

We did not dive too deep in this portion as the techniques of trading deviate from our main purpose - machine prediction of stocks. The figure below shows a comparison between our portfolio value along with the value of the stock day-by-day.

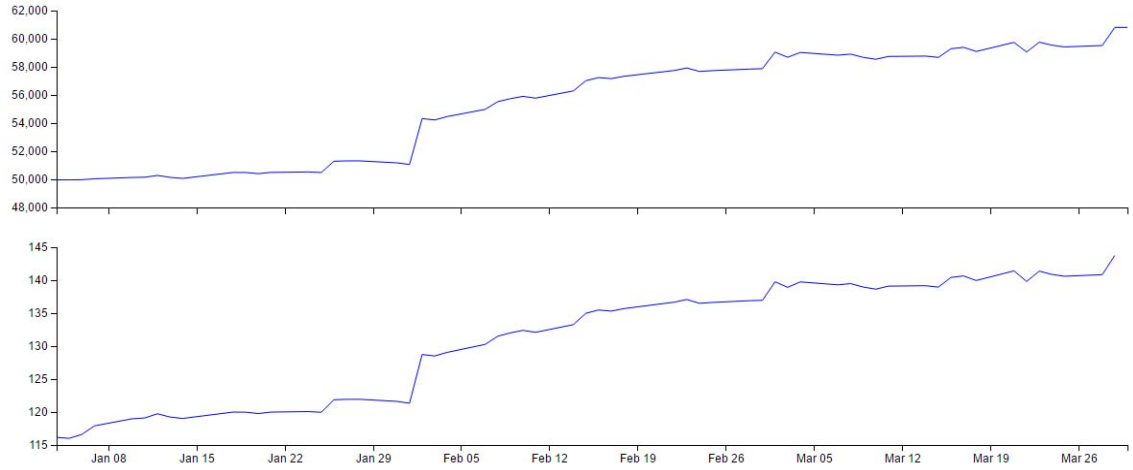


Fig 5. A comparison between portfolio and market price of the stock.

We employed comparisons between a hold and wait strategy which buys out a stock compared to a momentum based strategy. The momentum based strategy is based on the predictions and buys more of the stock is the predictions indicate upward movements of the stocks. It also sells stocks depending on the amount of capital remaining or a downward movement.

VII. Conclusion And Future Work

In our approach, we predict the stock movements of using online news articles. From the results we achieved as well as observations, there existed a correlation between the news articles and the movement of the stock on a day-to-day basis. This agrees with the opinion that one of the major contributing factors towards a change in stock movement is the public perception of the company itself.

This implementation itself can be improved on. We focussed mostly on letting the classifiers learn from the pure textual data. Sophisticated sentiment analysis techniques can be applied to this textual data along with other natural language processing methods to further provide a cleaner base for the classifiers to learn from. Moreover, machine learning itself can be applied to the trading simulation process to determine their ability to make decisions with such volatile processes.

Acknowledgment

We are grateful to BMS College of Engineering for having provided us with the facilities needed for the successful completion of this paper. The work reported in this paper is supported by the college through the TECHNICAL EDUCATION QUALITY IMPROVEMENT PROGRAMME [TEQIP-II] of the MHRD, Government of India.

References

[1] Gabriel Pui Cheong Fung, Jeffry Xu Yu, Hongian Lu. “The Predicting Power Of Textual Information On Financial Markets”.

[2] Chen, Jerry and Aaron Chai, Madhav Goel, Donavan Lieu, Faazilah Mohamed, David Nahm, Bonnie Wu, Predicting Stock Prices From News Articles.

[3] Gidofalvi, Gyozo, “Using news articles to predict stock price movements.” Department of Computer Science and Engineering,University of California, San Diego.

[4] Aase, Kim-Georg, “Text mining of news articles for stock price predictions.” (2011).

[5] Timmons, Ryan and Kari Lee, “Predicting the stock market with news articles.” CS224N Final Report (2007).

[6] Ma, Qicheng, “Stock price prediction using news articles.” CS224N, Final Report (2008).

[7] Joshi Kalyani, H. N. Bharathi and Rao, Jyothi, “Stock trend prediction using news sentiment analysis.” (2016).

[8] Vladimir N. Vapnik. The Nature of Statistical Learning Theory. Springer, New York, 1995.

[9] Text Categorization with Support Vector Machines: Learning with Many Relevant Features, Cornell, Thorsten Joachims, 1998

[10] NASDAQ: Stock Exchange, URL: www.nasdaq.com [Last Accessed: May, 2017]

[11] Google Finance, URL: finance.google.com [Last Accessed: May, 2017]

[12] Zipline: Trading Simulation Library, URL: www.zipline.io [Last Accessed: April, 2017]