**News based prediction of Stock price**

By

Vishnu Kalyan Mylavarapu

A Creative Component submitted to the graduate faculty in fulfillment of the requirements for the degree of MASTER OF SCIENCE

Major: Information Systems

Program of Study Committee:

Major Professor: Dr. Anthony Townsend

**ACKNOWLEDGEMENTS**

I would like to express my sincere and deepest gratitude to my Major Professor Dr. Anthony M Townsend for his guidance and support throughout my graduate studies and related research for the Creative Component.

I would also like to Professor Jon W Beard for his motivation and continuous guidance without which I would not have been able to complete my master’s degree with such confidence.

I am extremely grateful to my parents for their support, love and care that make me the person I am today. I would also like to thank my friends, teachers and peers for their timely help and encouragement.

**ABSTRACT**

We are living in an age where machine learning and data science in general are influencing our decision-making capabilities in all aspects of life. We now depend heavily on historic data to take crucial decisions about our health like whether we want to go ahead with a surgery or choose alternate paths. One field where the influence of such high-end cutting-edge technologies is playing a crucial part is the financial sector. Whether it is the use of Reddit pages to create a significant shift away from large financial corporations or the use of algorithmic trading to enhance profitability of our investment portfolio. The stock market is one of the biggest investment hubs for everyone and it is influenced by not only the way a company performs but also on the sentiment of the people towards the company and its products. One significant analysis that can be performed to predict the stock values better is the news-based sentiment analysis of a stock. The methodology is to get data from various news castors using an Application programming Interface (API), clean the data and perform a complete sentiment analysis to understand the correlation between stock value and news. Using models like Linear Discriminant Analysis (LDA), Linear Regression etc. we will try to find out whether we can predict a stock’s value increase or decrease in future. The model evaluation metrics used are Training and Testing accuracies, Precision, Recall and Confusion matrix.

1. **INTRODUCTION:**

Since the late 17th century there have been reports of financial market data analysis originating in Amsterdam to predict the Dutch financial market. Asia and European markets saw a rise in statistical predictions in the 18th and 19th centuries. In late 19th century Charles Dow, an American Journalist with Wall Street Journal came up with a theory called the “Dow theory” which predicted that there were patterns and cycles in the data obtained from the financial markets. Ever since investors always tried to predict and time markets to get the best profits possible. In order to increase accuracy in the 21st century a lot of machine learning models were built and tested. The influence of user perception on a stock is a significant factor and user perception is heavily influenced by the news about a particular company that is published in the daily newspapers and social media platform. L.I. Bing [3] proposed an algorithm to predict the stock price with an accuracy of 76.12% by analyzing tweets from public Twitter accounts.

In this project my key focus area would be analyzing and predicting stock prices of six major stocks [Microsoft, Apple, Nike, Goldman Sachs, Coca-Cola and Tesla] that are listed on the DOW jones index using daily news articles. Mining of the news for this project is done using the NewsAPI, an application programming interface with parameters set to get the news about these stocks for a period of one month. Once the data is mined, we will deploy methods like tokenization, stop-word and punctuation removal and Lemmatization to clean the data. We will then use TextBlob package from python to the Polarity and Subjectivity of the news articles. Then I will move onto perform a descriptive analysis using WordCloud package to get the most crucial words that are influencing the news. Finally, an attempt will be made to do statistical predictive analysis where data is split and models are built to predict whether a stock will go up or come down.

# **LITERATURE REVIEW:**

A number of previous attempts have been made to predict the stock market behavior and prices. Some of them focused on the tweet-based sentiment analysis while approached the problem using a more numerical approach where historic data was used to make predictions. Many such analysis proved an existence of a strong correlation between financial news and stock prices.[1][2].

1. *News based studies to predict stock value:*

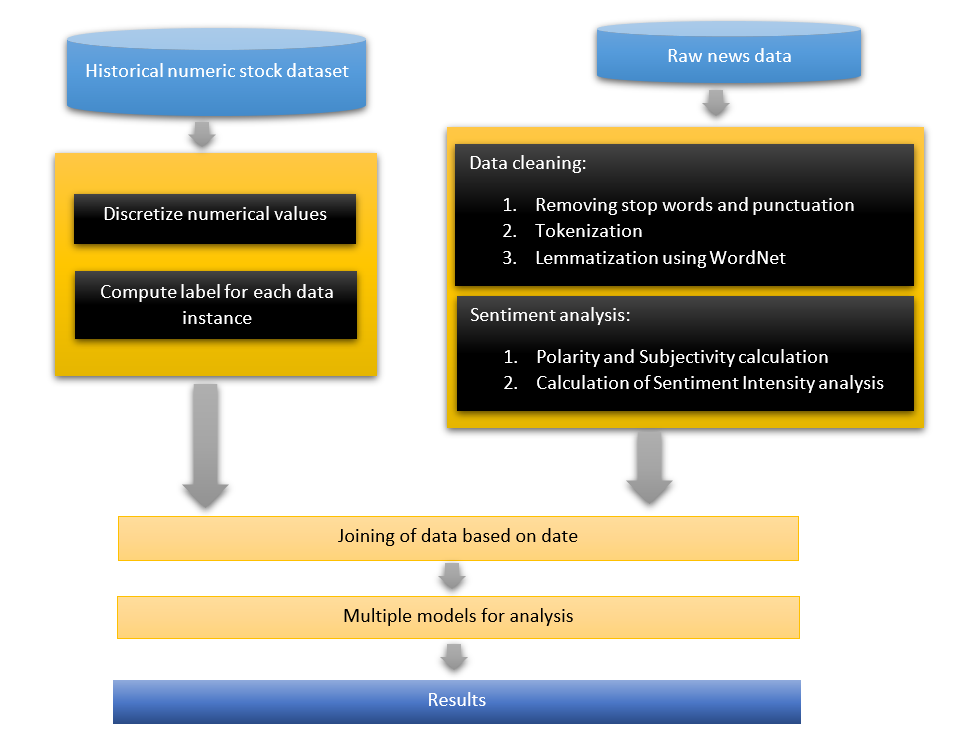
Y. Shynkevich [4], in his paper analyzed the relevance of specific stocks with their specific sub industry. In this paper he used Multiple Kernel Learning (MKL) which made use of neural network to analyze the data. The research was done to stocks from S&P 500 and a subcategory analysis was performed to get better results. The research was done to understand if creating subcategories will further enhance the predictability of the stock values. They were able to get an accuracy of 79.59% using polynomial kernel.

Association rule mining was used by Umbarkar[5] to find the whether a stock should be bought, held or sold on a given day. The prediction was based on all the technical indicators that were developed exclusively to predict the probability of value of the stock.

A general pattern that is observed is the change of customer sentiments based on the news that each individual media covers which indirectly establishes a correlation between stock price and the news that is being published. In order to understand how significantly a particular stock is influenced by news over the time we will first generate the polarity of the articles and the headlines that are being generated and then find the relation to come up with predictive analytics for each of the stock. Khedr’s[2] analysis is taken as a reference point for this project and minor modifications have been made in order to improve the data cleaning and extraction processes and a lot more amount of news will be used by this project as compared to his research.

1. **METHODOLOGY:**

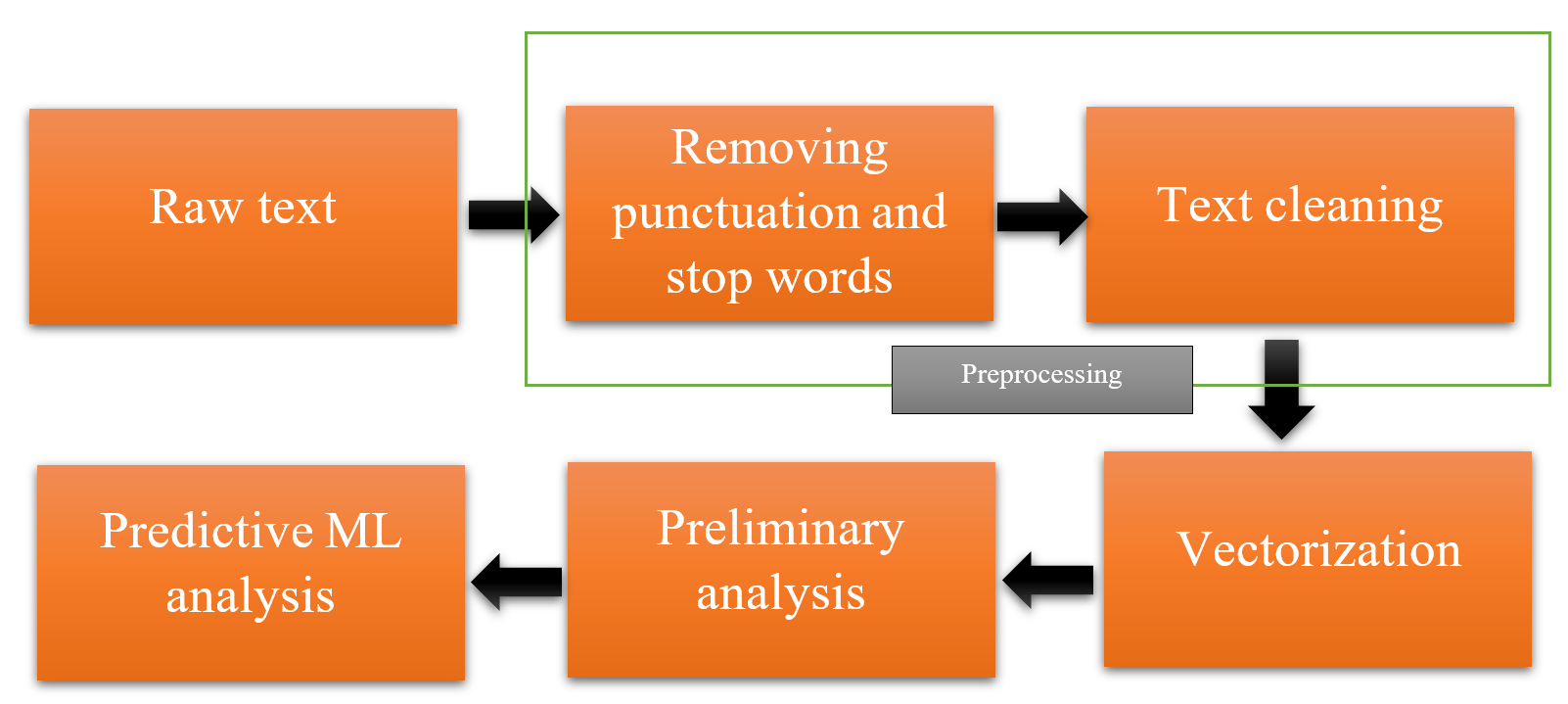
The proposed model predicts a stock based on various numeric parameters like Opening price, high price, low price along with the NLP model to evaluate corresponding news articles and make sure investors decrease risk in investment. Open price is the value of the stock when the market opens, high price is the maximum value of the stock on a given trading day and low price is the least amount that a stock hits on a given trading day. The aim of the project is to predict whether the price will go up or down at the end of the day. To achieve this task NLP pipeline with the architecture shown in figure 1 was built.



News API

*Figure1. Architecture of the NLP pipeline*

An in-depth look at the process followed for building the sentiment analysis module of the architecture is depicted in figure 2. In this we understand how the raw text is converted into numerical data to make sure it can be used by ML-algorithms to interpret and use while creating a model.



*Figure 2. NLP module of architecture*

**3.1Extraction of Raw data**

News API [6] was used to extract the data from multiple sources. It has similar mode of operation as Google News API had. It returns data in JSON (JavaScript Object Notation) format. Python was used to capture all the data that the API call returned and then converted the data into a pandas dataframe. This dataframe was then cleaned and stored in the form of Excel datasheet. The news of each individual stock was captured separately to make the process of combining the datasets easier and text processing faster.

To capture the stock data Yahoo finance API[7] was used. It has data about each stock in the following order.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Open | High | Low | Close | Adj. Closed | Volume |

Out of the data so obtained adj. Closed and Volume are not of significance to our analysis so these columns were dropped from the dataset created and the data of each stock was obtained separately and stored into CSV format using python Pandas.

**3.2 Removing punctuations for sentiment analysis:**

For sentiment analysis to be performed we need to remove punctuations from the sentences as these could emphasize the words that they are used next to and such an approach will lead to assigning wrong sentiments to the data that was input. The python module of Regex and string were used to remove all the punctuation mark (there is a set of all the punctuation marks pre-built in the string module of python).

**3.3 Tokenizing the sentences:**

Many of the operations that we need to perform on the text for completing the sentiment analysis needs data in the form of tokens rather than strings. Tokens are individual words in a sentence that are woven into an array of words. Regex was used to split the data and store them into a list in python. These lists were then added to the dataframe.

**3.4 Removing stop words:**

Words that do not add significant meaning to a sentence are called stop words. There are 3 kinds of stop words in English language.

1. Determiners- Determiners tend to mark nouns where a determiner usually will be followed by a noun. Examples- the, a, an etc.
2. Coordinating conjunctions- Coordinating conjunctions connect words, phrases and clauses. Examples- for, an, nor, but, or, yet, so etc.
3. Prepositions- Prepositions express temporal or spatial relations. Examples- in, under, towards, before etc.

These words tend to be ineffective in determining the sentiment of a particular sentence. Hence, we drop these words from the sentences. NLTK (Natural language toolkit) [9] is a prebuilt module in python that has the stop words stored for each language. NLTK module was imported and stop words were downloaded from the corpus of all the words. This list was then used iteratively over the Tokenized sentences and all the stop words from the sentences were removed. The remaining tokens were combined to form a new set of tokens that was attached back to the original dataframe.

**3.5 Stemming and Lemmatization:**

Stemming is the process of reducing inflected words to their word stem, base or root form. Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single term. The most common algorithm for stemming is the porter’s algorithm. Stemming tends to truncate and produce words that have no particular meaning as per dictionary. For example, the porter stemmed version of the word goose and geese will be goos and gees. Though these words are related stemming generally cannot infer meanings hence we use lemmatization in our project. Lemmatization reduces all the words into the root lemmas (base words). In our previous example both the words goose and geese will be converted to goose (which is the lemma).

The words generated after lemmatization will have a set meaning as per dictionary. The only disadvantage of lemmatization is the processing power that is needed (processing power was acquired and used for this project). The NLTK module in python has many pre-built algorithms for lemmatization. WordNet algorithm was used to build the lemmatized version. Once lemmatized all the tokens were combined as further analysis needed strings rather than tokens. A special function was built to join all the tokens.

**3.6 Sentiment analysis:**

One of the most critical part of the project was performing sentiment analysis on cleaned data. The process of sentiment analysis can be broadly be divided into three sub-processes:

1. Polarity calculation
2. Subjectivity calculation
3. Sentiment Intensity analysis

3.6.1Polarity calculation:

Polarity is the numerical measure of sentiment expressed in the sentence. The values range from -1 to +1. -1 means that the sentence expresses a negative sentiment in general. A value of +1 indicates that the emotion expressed is positive and 0 means the emotion expressed is neutral.



*Figure3:Polarity measure*

3.6.2 Subjectivity calculation:

Subjectivity is a numerical value that expresses some personal feeling, views or beliefs. Subjectivity comes in many forms like opinions, allegations, desires, beliefs, suspicions and speculation. The numerical value of subjectivity ranges from 0 to 1. 0 being non-subjective approach and 1 being complete subjective approach.

In this project both polarity and subjectivity are measured using TextBlob module in python. The package has built-in calculators for sentiment polarity and sentiment subjectivity for a particular text. Once the calculation is complete two new columns were appended to the dataframe in python to store the values.

3.6.3 Sentiment Intensity analysis:

Sentiment intensity analysis is a compound measure of the Positive, neutral and negative sentiment that are expressed in a given sentence. The total SIA is always equal to 1 and is very useful measurement for sentiment analysis as it adds a secondary tone to the given sentence thereby reducing the error rate. In this project a prebuilt module called vanderSentiment was imported with a method called SentimentIntensityAnalyzer. This method was used to add 4 new columns to the dataset namely, compound negative, positive and neutral. The entire process of cleaning data and adding the sentiments was repeated for all the 6 selected stocks and separate datasets were collected.

**3.7 Joining sentiment analyzed data:**

Once the datasets have been cleaned and the required values are derived, we then combine all the 6 stocks sentiment data with the stock market value data. This one final dataset is then used for all further analysis. This dataset contains 12 columns of data. This dataset is then used to build models and execute all the ML algorithms.

**3.8 Machine learning models applied:**

The final dataset is fed into the last step of the pipeline and the following analysis were performed:

1. Logistic regression
2. Linear Discriminant Analysis (LDA)
3. Decision tree
4. Random Forests

a. Logistic regression [10]:

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) that in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), although many more complex [extensions](https://en.wikipedia.org/wiki/Logistic_regression#Extensions) exist. In [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), logistic regression (or logit regression) is [estimating](https://en.wikipedia.org/wiki/Estimation_theory) the parameters of a logistic model (a form of [binary regression](https://en.wikipedia.org/wiki/Binary_regression)). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an [indicator variable](https://en.wikipedia.org/wiki/Indicator_variable), where the two values are labeled "0" and "1". In the logistic model, the [log-odds](https://en.wikipedia.org/wiki/Log-odds) (the [logarithm](https://en.wikipedia.org/wiki/Logarithm) of the [odds](https://en.wikipedia.org/wiki/Odds)) for the value labeled "1" is a [linear combination](https://en.wikipedia.org/wiki/Linear_function_(calculus)) of one or more [independent variables](https://en.wikipedia.org/wiki/Independent_variable) ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a [continuous variable](https://en.wikipedia.org/wiki/Continuous_variable) (any real value). The corresponding [probability](https://en.wikipedia.org/wiki/Probability) of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The [unit of measurement](https://en.wikipedia.org/wiki/Unit_of_measurement) for the log-odds scale is called a [logit](https://en.wikipedia.org/wiki/Logit), from logistic unit, hence the alternative names.

b. Linear Discriminant Analysis (LDA)[11]:

Linear discriminant analysis (LDA), normal discriminant analysis (NDA), or discriminant function analysis is a generalization of Fisher's linear discriminant, a method used in [statistics](https://en.wikipedia.org/wiki/Statistics) and other fields, to find a [linear combination](https://en.wikipedia.org/wiki/Linear_combination) of features that characterizes or separates two or more classes of objects or events. The resulting combination may be used as a [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier), or, more commonly, for [dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction) before later [classification](https://en.wikipedia.org/wiki/Statistical_classification).

c. Decision tree [12]:

Decision tree learning or induction of decision trees is one of the predictive modelling approaches used in [statistics](https://en.wikipedia.org/wiki/Statistics), [data mining](https://en.wikipedia.org/wiki/Data_mining) and [machine learning](https://en.wikipedia.org/wiki/Machine_learning). It uses a [decision tree](https://en.wikipedia.org/wiki/Decision_tree) (as a [predictive model](https://en.wikipedia.org/wiki/Predictive_modelling)) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). Tree models where the target variable can take a discrete set of values are called [classification](https://en.wikipedia.org/wiki/Classification) [trees](https://en.wikipedia.org/wiki/Decision_tree); in these tree structures, [leaves](https://en.wikipedia.org/wiki/Leaf_node) represent class labels and branches represent [conjunctions](https://en.wikipedia.org/wiki/Logical_conjunction) of features that lead to those class labels.

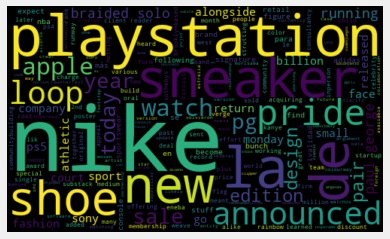
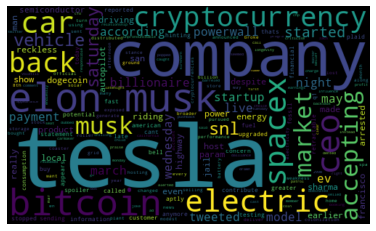
d. Random forests [13]:

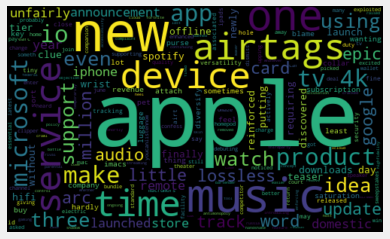
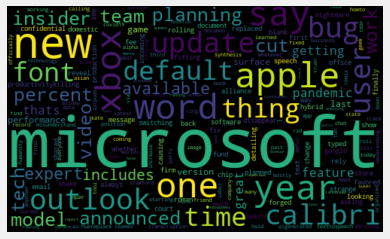
Random forests or random decision forests are an [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning) method for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and other tasks that operates by constructing a multitude of [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning) at training time and outputting the class that is the [mode](https://en.wikipedia.org/wiki/Mode_(statistics)) of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set).

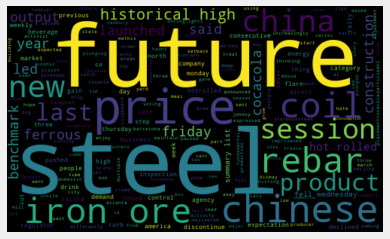
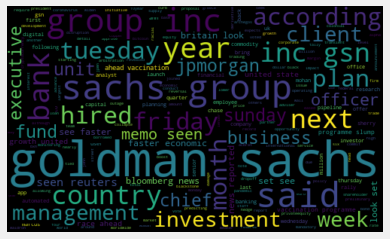
In this project the use of all the above methods is done using the sklearn machine learning module available in python. The dependent variable is a binary variable called “stock\_value” which is 0 when the price of the day went down and 1 when the price of the stock went up during the given trading day.

1. **PRELIMINARY ANALYSIS:**

Bilinear analysis of all the news articles for the six stocks using the WordCloud module in python revealed that the following words were the most important ones for each of the companies.

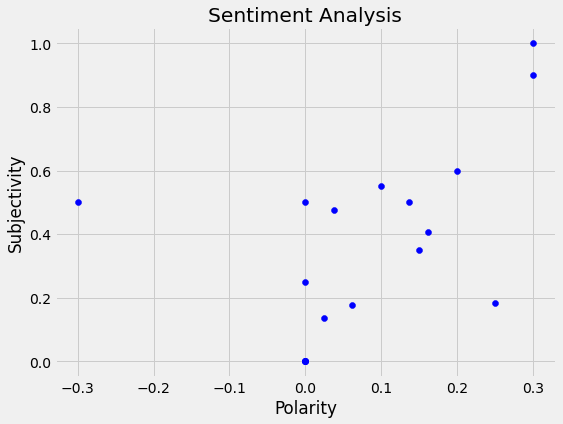
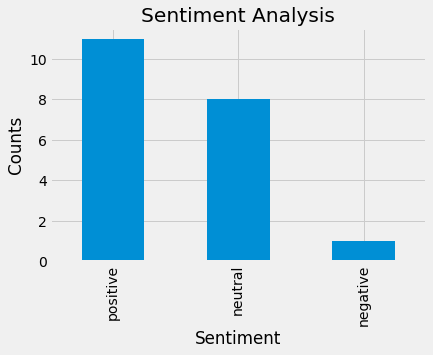


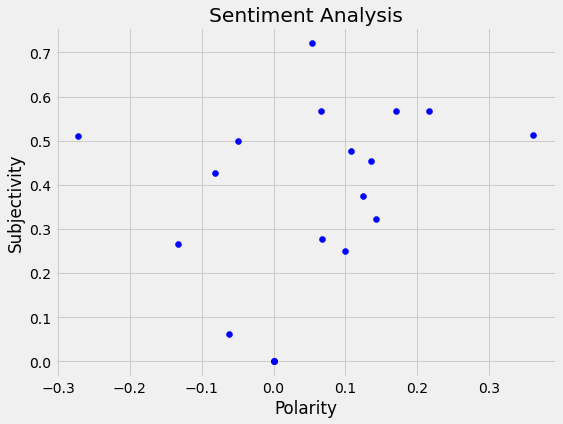
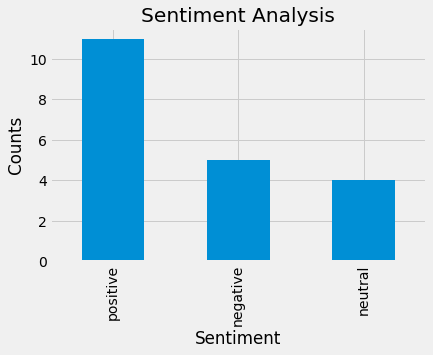


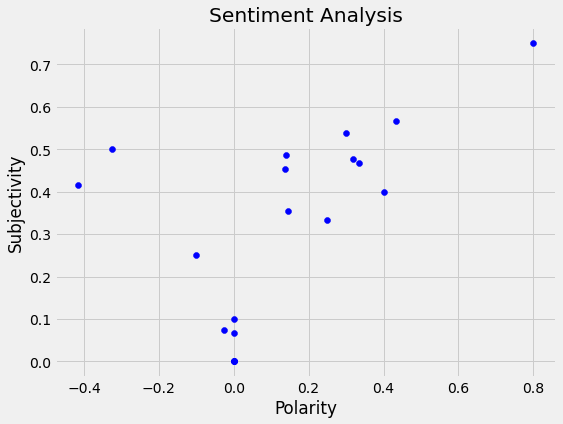
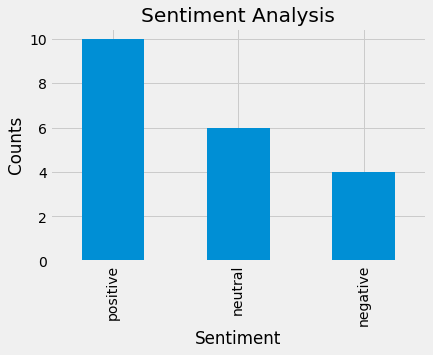


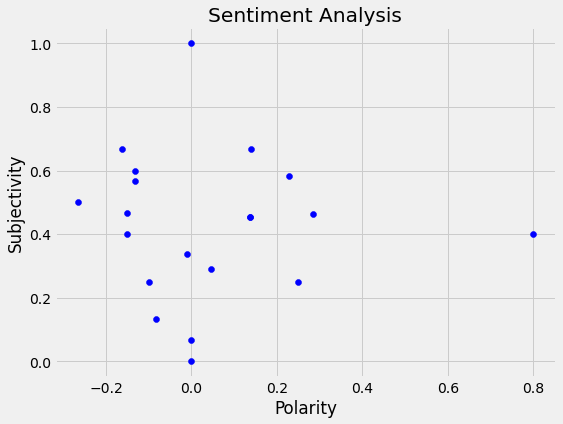
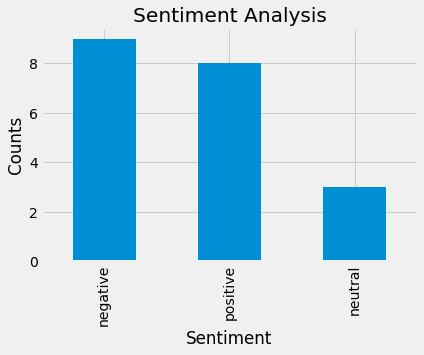
*Figure 4: Most popular words among news about the stock top to bottom a) tesla b) Nike c) Microsoft d) Apple e) Goldman Sachs f) Coca-Cola.*

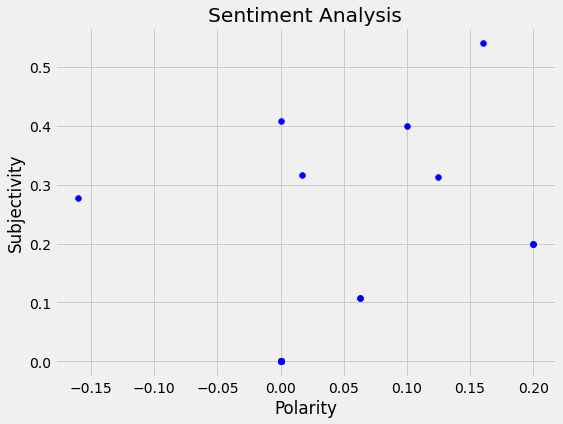
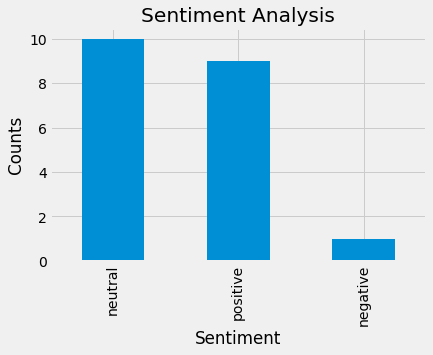
A plot of subjectivity vs. Polarity of each of the stock for each article published by the media houses looked like the following:

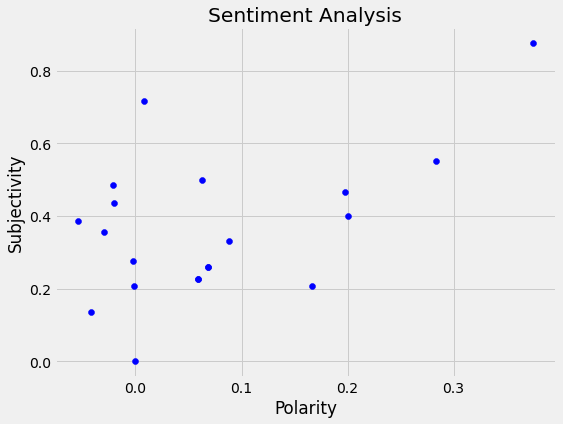
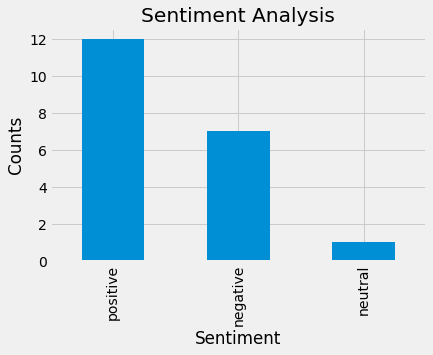
 

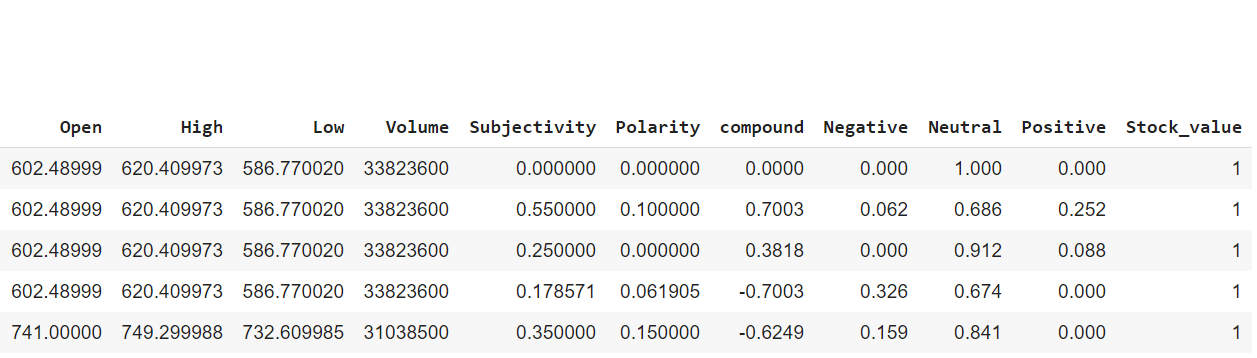
 

*Figure 5: Polarity v/s subjectivity and SIA top to bottom a) tesla b) Nike c) Microsoft d) Apple e) Goldman Sachs f) Coca-Cola.*

1. **PREDICTIVE ANALYTICS:**

A screen shot of the first 5 data points of the dataset on which final analysis was performed is shown below. In this dataset the target variable was the “Stock\_value”. Here we can observe that the Close data point has been removed and predicting that is what the model will be built for. The final dataset is a combination of data from all the stocks and hence any previous bias that was formed for one stock will be eliminated by this approach. The dataset was split into testing and training datasets where training dataset had 70% of the data and testing dataset had 30% of the data. All of the models that are used in these predictions are from the sklearn module of python.



*Figure 6: first 5 rows of the final data set that was analyzed*

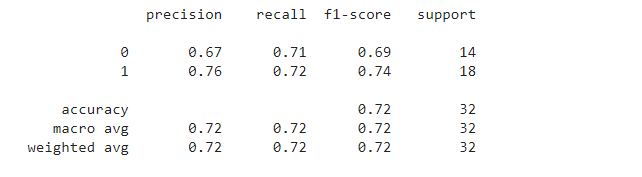
The following are the results of each of the machine learning algorithm when they are implemented:

1. Logistic regression:

Logistic regression predicted that the entire output was 1 as majority of the data points had a 1 as the output. This data is not very useful as predicting 1 irrespective of input would be very inefficient and hence, we need to discard it. The efficiency was just 52%

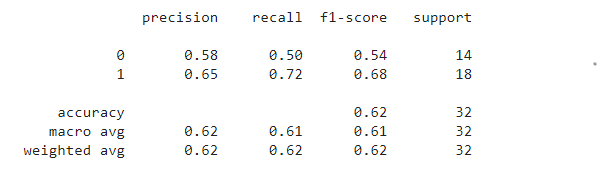
1. Linear Discriminant analysis:

With the linear discriminant analysis, we were able to increase the prediction accuracy to 72%. Precision and recall have improved a lot as compared to logistic regression as the error rate in prediction fell. The following is a classification report of LDA:



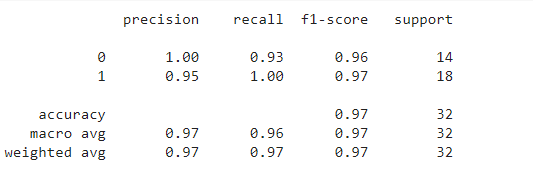
1. Decision tree:

With decision tree the accuracy fell to 62% and it was mainly due to the distribution of various data points and this could be eliminated if random forests was used. The following is the classification report for decision tree model:

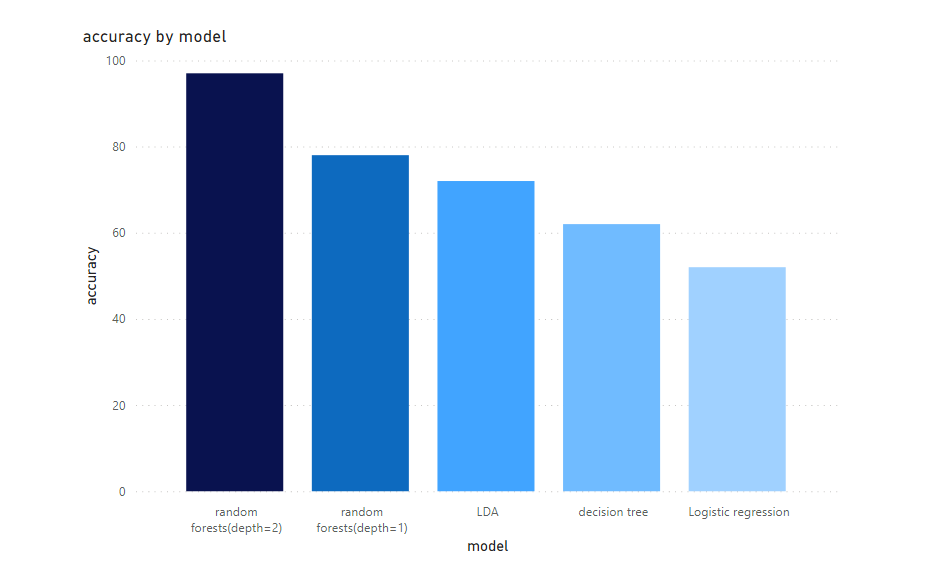


1. Random forests:

This is the model that gave the maximum efficiency in predicting whether the stock will go up or down. Even reducing the depth to just 2 made the model predict the output with an accuracy of 97%. This is a significant improvement as compared to any previous estimates we had. The following is the classification report of the random forest with a depth of 2:



In the end based on this data we can say that random forests can predict whether the stock will end higher or lower on a given day with some numerical data with an accuracy of 97%.



*Figure 7: Accuracy of each of the models compared*

|  |  |
| --- | --- |
| Model | Accuracy |
| *Logistic Regression* | *52%* |
| *Linear Discriminant analysis* | *72%* |
| *Decision tree* | *62%* |
| *Random forests (depth = 1)* | *78%* |
| *Random forests (depth = 2)* | *97%* |

1. **CONCLUSION:**

The project explored the effects of daily news along with historical data on predicting whether the stock will close higher or lower than the opening on a given day. Two categories were considered in this project: news about the stock and financial numerical data about the stocks that are published by the company. The proposed model was 2 staged, firstly we determine the sentiment the news is carrying and second is the use of various Machine learning models like Logistic Regression, LDA, Decision trees and Random forests to predict the output of dependent binary variable. A combination of polarity, subjectivity, Sentiment Intensity parameters and market numerical data together produced an accuracy of 97% with random forests. This project concludes the strong correlation between news of the day and the stock price change on a given day.

1. **REFERENCES:**
2. W. Walter, K. Ho, W. R. Liu, and K. Tracy, “The relation between news events and stock price jump : an analysis based on neural network,” 20th Int. Congr. Model. Simulation, Adelaide, Aust. 1–6 December 2013 <https://www.mssanz.org.au/modsim2013/>, no. December, pp. 1–6, 2013.
3. Khedr, Ayman & S.E.Salama, & Yaseen, Nagwa. (2017). Predicting Stock Market Behavior using Data Mining Technique and News Sentiment Analysis. International Journal of Intelligent Systems and Applications. 9. 22-30. 10.5815/ijisa.2017.07.03.
4. L. I. Bing and C. Ou, “Public Sentiment Analysis in Twitter Data for Prediction of A Company’s Stock Price Movements,” IEEE 11th Int. Conf. E-bus. Eng. Public, 2014.
5. Y. Shynkevich, T. M. McGinnity, S. Coleman and A. Belatreche, "Stock price prediction based on stock-specific and sub-industry-specific news articles," 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1-8, doi: 10.1109/IJCNN.2015.7280517.
6. Umbarkar, S.S., & Nandgaonkar, S. (2015). Using Association Rule Mining: Stock Market Events Prediction from Financial News.
7. News API – Search News and Blog Articles on the Web. n.d. News API - A JSON API for live news and blog articles. https://newsapi.org/, May 26, 2021.
8. Staff, R. A. P. I., Malika, Susana, Malika, Rao, V., & Uddin, B. (2021, April 16). How To Use the Yahoo Finance API (in 2021) [Tutorial]: RapidAPI. The Last Call - RapidAPI Blog. https://rapidapi.com/blog/how-to-use-the-yahoo-finance-api/.
9. Ganesan, K. (2020, July 30). What are Stop Words? Kavita Ganesan, Ph.D. https://kavita-ganesan.com/what-are-stop-words/#.YK6-gahKiUk.
10. Natural Language Toolkit. Natural Language Toolkit - NLTK 3.6.2 documentation. (n.d.). https://www.nltk.org/.
11. Wikimedia Foundation. (2021, May 22). Logistic regression. Wikipedia. https://en.wikipedia.org/wiki/Logistic\_regression#:~:text=Logistic%20regression%20is%20a%20statistical,a%20form%20of%20binary%20regression).
12. Wikimedia Foundation. (2021, May 6). Linear discriminant analysis. Wikipedia. https://en.wikipedia.org/wiki/Linear\_discriminant\_analysis.
13. Wikimedia Foundation. (2021, May 6). Linear discriminant analysis. Wikipedia. https://en.wikipedia.org/wiki/Linear\_discriminant\_analysis.
14. Wikimedia Foundation. (2021, May 6). Random forest. Wikipedia. https://en.wikipedia.org/wiki/Random\_forest.