Prediction of the Dow Jones Industrial Average with Reddit News Headlines using Natural Language Processing (NLP) in Python

By: Shaheryar Afroze

Contents

[1. Abstract 3](#_Toc46784603)

[2. Literature Review 3](#_Toc46784604)

[3. Data Description and EDA (Exploratory Data Analysis) 8](#_Toc46784605)

[Commands that were used in initial data exploration 8](#_Toc46784606)

[4. Approach 14](#_Toc46784607)

[Data Collection 14](#_Toc46784608)

[Creating the class label using DJIA dataframe 14](#_Toc46784609)

[Reorganizing Reddit News Dataset 15](#_Toc46784610)

[Merging Reddit News and DJIA Datasets 15](#_Toc46784611)

[Data Exploration 15](#_Toc46784612)

[Feature Selection and Preprocessing 16](#_Toc46784613)

[Attribute Selection 16](#_Toc46784614)

[Text Preprocessing 16](#_Toc46784615)

[Feature Engineering and Model Building 17](#_Toc46784616)

[5. Results 19](#_Toc46784617)

[6. Conclusions 24](#_Toc46784618)

[7. References 25](#_Toc46784619)

[8. Common NLP Terminology 26](#_Toc46784620)

# Abstract

Data analytics can be useful in the stock markets because there are vast amounts of continual influx of data. News could be a factor in stock prices since news headlines can influence people’s decisions in the stock market.

In this capstone project, the theme was on prediction of market direction, classification problem, of market direction based on news headline text using NLP (natural language processing) techniques. The model will use historical news headlines against the stock movement to train to train its prediction algorithm. Next, the machine attempted to predict the outcome of the movement of the Dow Jones based on the untested news headlines to test the effectiveness in the algorithm to predict the direction of the market.

Though this project is not using live feed of news, this exercise should provide a basis of NLP and machine learning concepts and can later be applied to live news for better accuracy and applications towards predictions.

Hence, this project would be able to answer following questions: Is there a relationship between news headlines and the movement of the stock markets? And to what varying degree? Was the news source used by the dataset an effective source? Is there an optimal time from the release of the headlines to the reaction of the markets?

The dataset can be found on Kaggle: https://www.kaggle.com/aaron7sun/stocknews

Sun, J. (2016, August). Daily News for Stock Market Prediction, Version 1. Retrieved [June 04, 2020] from https://www.kaggle.com/aaron7sun/stocknews.

The tools used for this project are: Google Colabs for Python environment, Pandas to deal with dataframes, matplotlib and ggplot for visual representations, sklearn and NLTK package for NLP and machine learning.

The code for the project can be found on Github:

https://github.com/QOneK/Ryerson-Data-Analytics-Final-Project-for-Kyuhwan-Kim

# Literature Review

**Anthony (Tony) Cox, L., Jr. (2017), Misbehaving: The Making of Behavioral Economics by Thaler, Richard. Risk Analysis, 37: 1796-1798. doi:**[**10.1111/risa.12871**](https://doi.org/10.1111/risa.12871)

The author of the book was Richard Thaler who was a nobel prize winner (<https://www.nobelprize.org/prizes/economic-sciences/2017/thaler/facts/>)

He found proof that companies can be over or undervalued debunking the **Efficient Market Hypothesis**: share prices reflect all information and that they trade at fair market value.

Thaler conducted an experiment by building two theoretical portfolios of stocks: “winners” and “losers”. The “winners” portfolio was made up of companies whose stock price had recently performed exceptionally well. Unlike “losers” was made up of stocks that had recently performed exceptionally poor. Thaler’s theory was that these movements were fueled by investors being over-enthusiastic about the “winners” and overly-pessimistic about the “losers”. This theory would prove to be correct if all these extreme prices would regress back towards the mean, causing the “losers” portfolio to bounce back and perform better.

This seems to indicate that stock prices can be influenced by herd behavior. This form proves the concept of **value investing**: choosing stocks that appear to be underestimated by the market.

Reflections towards Project

The find if the project is a feasible investigation, there were studies done on market theory. Based on this evidence, it appears that sentiment analysis can have potential to influence the market due to collective sentiment towards the participants of the market. There is evidence that extrinsic factors can have impact on stock price and news headlines could be of use towards prediction of the market.

**Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems With Applications*, *42*(24), 9603–9611. https://doi.org/10.1016/j.eswa.2015.07.052**

Since the dataset is from a social media news headline, exploring a study using social media sentiment towards predicting stock price movement would be valid. One of the challenges in this study was that texts found in social media contexts are usually short, contains many misspellings, uncommon grammar constructions and so on. Previous studies report that sentiments from social media have no predictive capabilities while other researchers have reported either weak or strong predictive capabilities. The author thinks that use of opinions in social media for stock price predictions is still an open problem.

To better increase accuracy, it is important to recognize what topics are being discussed which can be coined with ‘topic-sentiment’.

**EMH** (Efficient Market Hypothesis) and **random walk theory** were some concepts that were gained. EMH says that the current stock market reflects all available information and that price changes are merely due to new information or news. Because news happens randomly and is unknowable in the present, stock prices should follow a random walk patter and the best for the next price is the current price and are not predictable with more than 50% accuracy. On the other hand, there is still a debate whether the market is predictable and a study claimed that degrees of directional accuracy at 56% hit rate is a satisfying result for stock predictions. Based on the article describing other studies, combining textual content with historical prices through the **linear regression model** is an approach to do this.

In this study, to obtain historical prices, they were retrieved from Yahoo Finance. They have used **adjusted prices** which are close prices which are adjusted for dividends and splits. The reason is because adjusted close price is often used for stock market predictions in other researches. The messages were collected from the corresponding stocks’ message boards. The time period to collect data was 1 year. This study has decided not to use twitter data because there is too much noise and finding sentiments related to specific stock difficult.

The method used was **SVM (Support Vector Machine)** which has been recognized as being able to efficiently handle high dimensional data and perform well on **classification**. Once the **stop words** were removed, the **lemmatization** was done using **Stanford CoreNLP** with **feature representation** is the **bag of words** from the title and the content of the message. The feature weighting is **TF-IDF. SVM** with linear kernel was chosen as the classification model.

The result has found 54.41% average accuracy for 18 stocks. Has claimed that performs much better than other methods for the stocks that are difficult to predict with only past prices. The limitation of this model is that it is insufficient to forecast drastic movement of the stock market. Another weakness is that only the historical prices and sentiments are considered.

Reflections towards Project

Trying to develop a model which is able to predict the stock market is very difficult since it is affected by many factors. Though there were take-aways, such as using linear model classification and using adjusted close price, there are some differences. For instance, this study is using financially related comments board to extract the sentiment data directly from the corresponding stock. In my project, top viewed reddit news headlines will be used which has a more general outlook. Also, instead of comparing the stock related comment to the stock, I will be comparing top 25 viewed news headlines to the movement of the DJIA (index which sees the movement of top 30 companies). This article has allowed to me gain a perspective in methods used and to think more critically about my dataset.

**Thorat, S., Deshpande, P., & Shaga, V. (2017). REVIEW OF SENTIMENT ANALYSIS ON TWITTER DATA USING PYTHON. *International Journal of Advanced Research in Computer Science, 8*(9) Retrieved from http://ezproxy.lib.ryerson.ca/login?url=https://search-proquest-com.ezproxy.lib.ryerson.ca/docview/1980479225?accountid=13631**

Though I am not using twitter data, this paper is significant enough since it relates to text data and python programming language. This paper provided an overview of the steps involved with text data and some overview of classifiers available such as: KNN and Naïve Bayes.

Text mining is preprocessed data for text analytics. It is a process of exploring textual data and finding patterns. Text mining process the text itself while NLP process with the underlying metadata and NLP is one of its components. The steps towards preprocessing and classifiers are explained in the paper.

Reflection towards Project

There are various text data related project such as sentiment analysis. There seems to be some similarity and follow under an umbrella “text mining”; which is defined as: data mining process which follows some pattern analysis in that must give meaningful information regarding the project. I recognize that the main objective for the dataset is to do textual analysis on the binary classifier. But, many more things can be applied beyond the scope of what was intended.

**Madnani, N. (2007). Getting started on natural language processing with Python. *XRDS: Crossroads, The ACM Magazine for Students*, *13*(4), 5-5.**

This article was useful in understanding NLP (Natural Language Processing) concepts with Python. This paper was helpful in understanding the basic terminologies that are used in NLP are described here such as: tokenization, corpus, parsing and etc.

Some of the tools for NLP were mentioned such as **NLTK** (Natural Language Toolkit). Such tools are necessary since Python alone is not powerful enough for most NLP tasks and the benefit is that it is an open sourced and is entirely self contained. The corpus text corpora that are used are: Brown, Gutenberg, Stopwords. Brown has about 1 million words of American English texts and it mentioned that much manual work went into create a POS tagged version.

It went to explain about **Zipf’s law** that there should be a relationship between frequency of a word and its position in the list which can be expressed mathematically f\*r (f = frequency, r = rank). For example, 5th most frequent word should occur exactly two times more frequently than 10th most frequent word.

A POS (Part Of Speech) tagged corpus will come in a list of 2 tuples, (token, tag). For example, a sentence like, "The ball is green," from a tagged corpus, would be represented inside NLTK as the list [(*'The'*, *'AT'*), (*'ball'*, *'NN'*), (*'is'*, *'VB'*), (*'green'*, *'JJ'*)].

Reflection towards Project

When doing my project, with the count function, I should be able to test whether Zipf’s law would hold in the dataset provided. Also, the machine learning model used should have a POS tagged corpus.

**G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller. 2015. Scikit-learn: Machine Learning Without Learning the Machinery. *GetMobile: Mobile Comp. and Comm.* 19, 1 (January 2015), 29–33. DOI:https://doi.org/10.1145/2786984.2786995**

While exploring other tool kits, Scikit-learn was also mentioned for NLP. This paper was explaining and introducing this open source package which possesses many powerful scientific and numeric tools including text processing leveraging other toolsets like Numpy, SciPy, and matplotlib.

This article explained the concept of **overfitting** well. This occurs when machine is learning its prediction from the noise of the data and will show considerably different (usually higher) prediction error on new data compared to the training data. In our project, what we want to do is create an accurate prediction which doesn’t overfit. Hence, what we want to do is to create the most accurate prediction model possible as there should be similar prediction results from testing and training data.

**SVM (Support Vector Machines)** learns its decisions with linear functions and the risk in this case would be **underfitting** (algorithm not fully using the richness of the data) that don’t follow the linear law since the algorithm has lower model complexity because the number of parameters to learn from the data is much smaller.

The “art” of machine learning consists of choosing the right algorithm and models to describe the data and to find the “sweet spot” between under and over fitting. **Curse of Dimensionality** arises as the number of features and parameters grows, the risk of overfitting increases. In other words, as data gets more complex, the risk of overfitting grows.

When setting up models, training data undergoes fit method which accepts (n x p) data matrix represented as a NumPy array. The goal of **supervised models** is prediction of value of interest and its performance is measured on the ability to correctly predict new data. Data can be split into training and testing data and **cross validation scheme** (data is repeatedly split into train and test subsets). Scikit learn provides integrated support for cross validation using **cross\_val\_score** function.

Reflection for Project

There are more than one machine learning model that can be used for the project. It would take experimentation to determine which model would best reflect on the project. Also, over and underfitting is an issue. To avoid ‘curse of dimensionality’, I would need to clean the dataset as much as possible to minimize the noise for the ML model. My dataset would be part of supervised machine learning since the class label is already determined.

**Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.**

This is another scikit learn paper. The significance of this paper is that on the package’s documentation page that can be found at: <https://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/about.html>

Hence, this can be considered as the main thesis paper for scikit learn. The paper mentions that the main purpose of the development of the package is to fill the growing need for statistical data analysis by non-specialists. Scikit-learn provides about 300 page user guide with more than 60 examples while trying to minimize machine learning jargon.

The underlying technologies are: Numpy, Scipy and Cython. The code is designed ‘objects specified by interface, not by inheritance’.

Reflections for the Project

Since this package is meant for non-computing specialists, this package would be a useful tool to be used towards the project. Also, there would be some examples to follow to get started to understand how the package can be used. The paper explaining sci kit learn was still difficult to comprehend so much practice on the coding will be required to successfully complete the project.

**Yamashita, Y., Joutaki, H., & Takahashi, H. (2013). Analysing the influence of headline news on the stock market in japan. *International Journal of Intelligent Systems Technologies and Applications, 12*(3-4), 328-342. doi:10.1504/IJISTA.2013.056539**

This paper was used since news headlines were used to assess the stock market. Unlike other methods prior, this paper used **text mining** techniques. They found a significant relationship between the stock market changes and the negative/positive news classified by keywords.

Headline news is one of the main sources of information accessed by institutional investment fund managers. Prior to this study, it has been reported that a significant statistical relationship has been found between headline news and interest rate change and investment strategy using such news can be made. This paper sued ‘Quick’ which is one of the best media sources for the business of asset management in Japan. In this analysis, positive and negative keywords are selected in advance. Headline news is classified into positive/negative news.

The result from the study found that there is a significant statistical relationship between news headlines and stock index returns. Also, the list of keywords used in the analysis had a big influence on classification accuracy through analysis.

Reflections for the Project

In this study, an asset management news channel ‘Quick’ was used. This would be different from the peer chosen top reddit news chosen by peer users. Also, Dow Jones is in U.S. while this paper assessed the Japanese stock market. Also, the dictionary is not likely to be premade for the analysis. Therefore, differences between this analysis and the project is to be likely.

A major take away is that how the machine learning model analyzes the text matters in the end result of the classification.

Kavšek, B. (2017). Using words from daily news headlines to predict the movement of stock market indices. *Managing Global Transitions, 15*(2), 109-121. doi:10.26493/1854-6935.15.109-121

This was a difficult find but this journal article used the same dataset that will be used for the project. The objective that the author had was “does a word or combination of words from news articles, such that their presence or absence tells us something about stock price movement?”. It was found that the approach used was not sentiment analysis. What was found was that the news data were aligned to the stock data on the same day but experiments have showed that predicting stock value from ‘yesterday’s’ news provides higher predictive accuracy.

The work was divided into 4 phases: data transformation, modelling technique selection, quantitative evaluation and qualitative evaluation. WEKA was the tool used to.

After the data was pre-processed, **Bag-of-Words** technique was used which takes text as input and produces a vector in which every element represents the number of appearances of the word in the corpus. They were all converted to lowercase and eliminated **stop words**. Then **stemming** was done.

The ‘date’ feature was removed since the goal is to model the dependency of the DJIA index from information contained in the news headlines, rather than explore the time series nature of the data. The data was split in 80% training and 20% to test. The paper states the high importance of splitting the data in a sequential date split since we want to simulate the process of predicting ‘new’ (unknown) events from ‘old’ (already known) events. The algorithms used were: **Naïve Bayes**, **SVM, k-nearest neighbours** and **random forests**. **Majority class classifier** and ‘one-rule’ classifier or **OneR** was used to find the independent variable that is most correlated with the label variable.

Reflections for the Project

There are some distinct differences in how the study conducted in this paper and the project will be. My project will be coded in Python. The algorithms used might be different. However, there are similarities in that the data needs to be preprocessed. This paper has provided some insights such as using bag of words, eliminating stop words, stemming and removing features. It also provided a reason to split the data and to test the model sequentially. Since Python will be quite different from that of WEKA, exploration of code and algorithms will need to be done.

# Data Description and EDA (Exploratory Data Analysis)

The dataset can be found on Kaggle: <https://www.kaggle.com/aaron7sun/stocknews>

Sun, J. (2016, August). Daily News for Stock Market Prediction, Version 1. Retrieved [June 04, 2020] from https://www.kaggle.com/aaron7sun/stocknews.

The dataset contained 3 files which are in .csv format:

RedditNews.csv

DJIA\_table.csv

CombinedNewsDJIA.csv

## Commands that were used in initial data exploration

\*These commands were organized in order of occurrence

.info() – this outputs the type (the files that were dealt with were pandas dataframe files), range (ex. 0 to 73607), and displays the columns (with counts and data types) and miscellaneous details such as the size of the file

.min() – finds the minimum value ex. earliest date in a column

.max() – find the maximum value ex. latest date in a column

.nunique() – displays how many unique values are present in a column

.describe() – displays rudimentary statistics, works only with numeric datatypes since count, mean, std, min, max values are outputted

.sort\_values – organizes data ascending/descending order

.rest\_index – when the data is resorted asc/des order, the index remains the same. Therefore, this command was necessary to have the indexes in chronological order

.value\_count() – counts the number of values. For instance, in the class label, this command was used to count how many 1 and how many 0 were present.

.isnull() – checked to see if null values were present

**RedditNews.csv**

This .csv file was declared as variable ‘reddit’.

Historical news headlines from Reddit WorldNews Channel (/r/worldnews). These news headlines were ranked from users’ votes. The Top 25 Headlines are compiled for a single date and are listed. The range of dates are from 2008-06-08 to 2016-07-01. The data has 2 columns: Date, News.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Column Name | Count | Minimum | Maximum | Data Type |
| 0 | Date | 73608 | 2008-06-08 | 2016-07-01 | object |
| 1 | News | 73608 |  |  | object |

The number of dates present were 2943 dates.

This seems to be correct since using a ‘Days Calculator: Days Between Two Dates (<https://www.timeanddate.com/date/durationresult.html?m1=06&d1=08&y1=2008&m2=07&d2=01&y2=2016>) calculator has found that there are 2945 dates between 2008-06-08 and 2016-07-01. And another interesting find was that there were 73537 unique articles; this is interesting since if each article were unique, then 73608 articles should be present. The reason still cannot be confirmed but a plausible hypothesis can be that there can be duplicates or for some reason, the 71 articles were not readable.

**DJIA\_table.csv**

This .csv file was declared as variable ‘djia’.

The stock data is represented with Dow Jones Industrial Average (DJIA) which is downloaded from Yahoo Finance. From this dataset, the column of uttermost importance is column 6, ‘Adj Close’ column which is found on the CombinedNewsDJIA.csv file and this will be used to calculate the class label.

Notice the date ranges for Dow Jones are different from RedditNews.csv as the starting range for date is later (2008-08-08).

Here are some financial background information be better understand this table:

**Dow Jones Industrial Average (DJIA)** is a stock market index that measures the stock performance of 30 largest bluechip companies in the United States from the New York Stock Exchange (NYSE) and NASDAQ.

**Open**: Price at which stock first traded on a trading day.

**High**: Highest price at which stock traded during a trading day.

**Low**: Lowest price at which stock traded during a trading day.

**Close**: Last price at which stock traded during a trading day.

**Volume**: Number that stock changes hands over trading day.

**Adj Close**: Adjusted Closing Price. Amendment of closing price to accurately reflect the stock’s value after accounting for any corporate actions such as: stock splits, dividends, distributions, rights offerings and etc. Considered to be “true price” of a stock and is often used in analyzing historical returns.

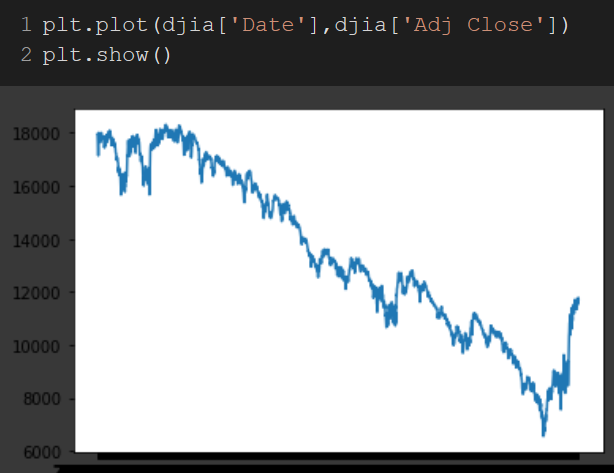
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Column Name | Count | Minimum | Maximum | Data Type |
| 0 | Date | 1989 | 2008-08-08 | 2016-07-01 | object |
| 1 | Open | 1989 |  |  | float64 |
| 2 | High | 1989 |  |  | float64 |
| 3 | Low | 1989 |  |  | float64 |
| 4 | Close | 1989 |  |  | float64 |
| 5 | Volume | 1989 |  |  | int64 |
| 6 | Adj Close | 1989 |  |  | float64 |

|  | **Open** | **High** | **Low** | **Close** | **Volume** | **Adj Close** |
| --- | --- | --- | --- | --- | --- | --- |
| **count** | 1989.000000 | 1989.000000 | 1989.000000 | 1989.000000 | 1.989000e+03 | 1989.000000 |
| **mean** | 13459.116048 | 13541.303173 | 13372.931728 | 13463.032255 | 1.628110e+08 | 13463.032255 |
| **std** | 3143.281634 | 3136.271725 | 3150.420934 | 3144.006996 | 9.392343e+07 | 3144.006996 |
| **min** | 6547.009766 | 6709.609863 | 6469.950195 | 6547.049805 | 8.410000e+06 | 6547.049805 |
| **25%** | 10907.339844 | 11000.980469 | 10824.759766 | 10913.379883 | 1.000000e+08 | 10913.379883 |
| **50%** | 13022.049805 | 13088.110352 | 12953.129883 | 13025.580078 | 1.351700e+08 | 13025.580078 |
| **75%** | 16477.699219 | 16550.070312 | 16392.769531 | 16478.410156 | 1.926000e+08 | 16478.410156 |
| **max** | 18315.060547 | 18351.359375 | 18272.560547 | 18312.390625 | 6.749200e+08 | 18312.390625 |

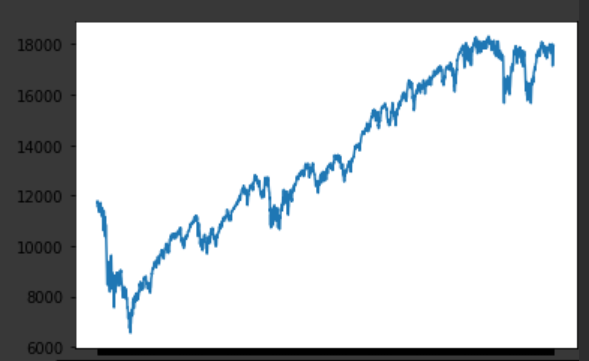
This figure can be obtained using **djia.describe()** command

The column of interest for the project is **Adj Close** from this dataset.

To visualize a trend, a line graph was plotted using Date as the x-axis and Adj Close as the y-axis



Based on the timeframe that was gathered, we can see that the Dow Jones was on a downward trajectory with a bounce back near the end. However, upon closer look, it was discovered that the dates were not in ascending order. Therefore, the dates were correctly sorted and the indexes were reset for correct formatting. Also, this step was necessary since the articles and the combined data, were organized by dates in ascending order.



This is the correct plotting of the DJIA. The stock index was at an rising trajectory.

Unlike RedditNews.csv, there were no missing values that are unaccounted for. This was good news considered when merging, the dates from DJIA, not Reddit, will be used to merge the files Reddit and DJIA. DJIA had shorter date ranges which meant that Reddit had more than adequate data present to create a joint database.

The values of DJIA database were integer or float values except for date (object).

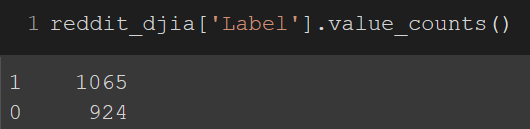
**CombinedNewsDJIA.csv**

This dataset was created by the user to make things easier for his students which is a combined dataset of RedditNews.csv and DJIA\_table.csv with 27 columns. The first column is the Date and second column is the Label. The following ones are news headlines ranging from “Top1” to “Top25”.

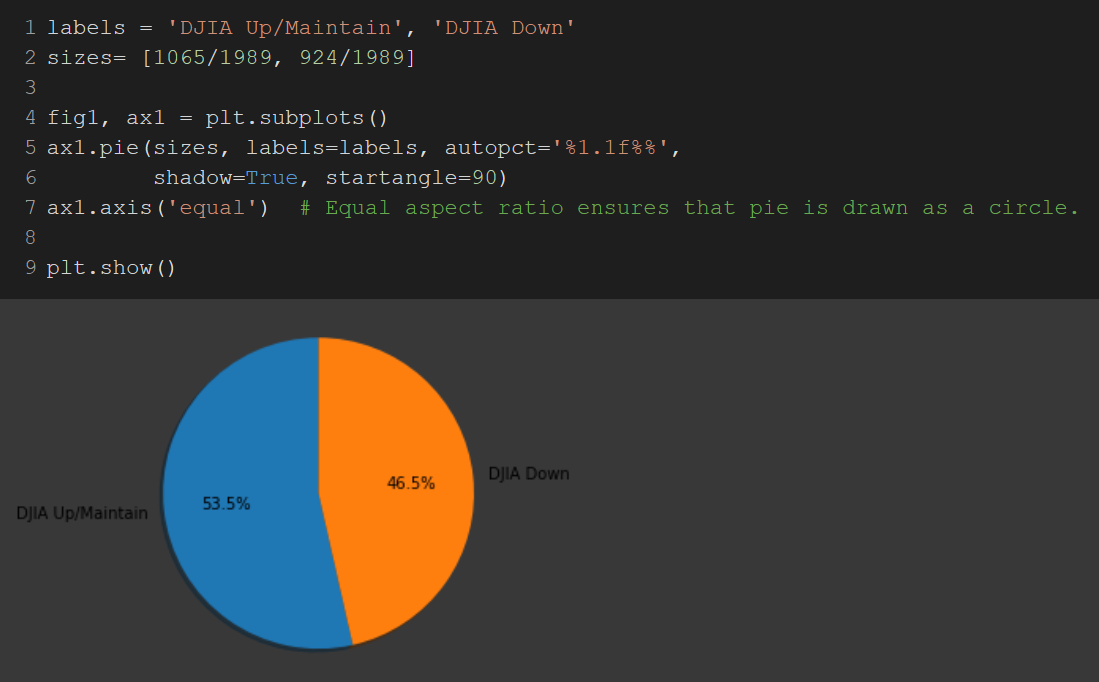
This is the result that we would want if we were to merge RedditNews and DJIA databases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Column Name | Count | Minimum | Maximum | Data Type |
| 0 | Date | 1989 | 2008-08-08 | 2016-07-01 | object |
| 1 | Label | 1989 |  |  | int64 |
| 2 | Top1 | 1989 |  |  | object |
| 3 | Top2 | 1989 |  |  | object |
| 4 | Top3 | 1989 |  |  | object |
| 5 | Top4 | 1989 |  |  | object |
| 6 | Top5 | 1989 |  |  | object |
| 7 | Top6 | 1989 |  |  | object |
| 8 | Top7 | 1989 |  |  | object |
| 9 | Top8 | 1989 |  |  | object |
| 10 | Top9 | 1989 |  |  | object |
| 11 | Top10 | 1989 |  |  | object |
| 12 | Top11 | 1989 |  |  | object |
| 13 | Top12 | 1989 |  |  | object |
| 14 | Top13 | 1989 |  |  | object |
| 15 | Top14 | 1989 |  |  | object |
| 16 | Top15 | 1989 |  |  | object |
| 17 | Top16 | 1989 |  |  | object |
| 18 | Top17 | 1989 |  |  | object |
| 19 | Top18 | 1989 |  |  | object |
| 20 | Top19 | 1989 |  |  | object |
| 21 | Top20 | 1989 |  |  | object |
| 22 | Top21 | 1989 |  |  | object |
| 23 | Top22 | 1989 |  |  | object |
| 24 | Top23 | 1988 |  |  | object |
| 25 | Top24 | 1986 |  |  | object |
| 26 | Top25 | 1986 |  |  | object |

Using the ‘describe’ function, we were able to see that there are more 1s than 0s. This is evident since mean is > 0.5.



This was evident when counting the number of 1s and 0s present in the Label column. There were 1065 1s (up or remain) and 924 0s (down).



Each corresponding date will possess 25 news headlines and label will provide detail as to what the stock movement was for the day.

The creator has made this to be a binary classification task There are two labels for the label column:

“1” when DJIA Adj Close value rose or stayed as the same.

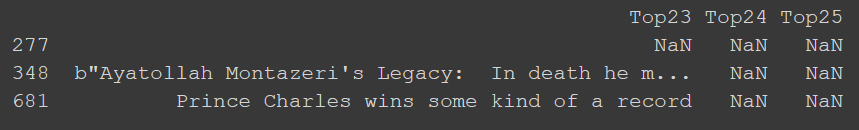
“0” when DJIA Adj Close value has decreased.

Also, it was advised to make a 80%/20% split from the data creator for training (2008-08-08 to 2014-12-31) and test sets (2015-01-02 to 2016-07-01)

Upon close inspection, it was found that there were missing articles in this dataset.

|  |  |
| --- | --- |
| Top23 | 1988 |
| Top24 | 1986 |
| Top25 | 1986 |

All the values within the table should read 1989 but some columns have a lower number. With coding, it was indeed found that even the ‘polished’ dataset had some missing values to be dealt with.



277, 348, 681 were the indices (locations) where the missing articles are found.

Initially, there was an expectation that the cleaned dataset would have no missing values and each date would have 25 articles. This has caused some problems in compiling towards the final cleaned dataset since certain Pandas commands would not work since the shape was not perfect. There were work arounds needed and this was done but compiling the articles in to list of lists and outputted to its dates.

However, it was later found that this was not a major deal breaker in further exploring the data.

The reason was that when coding for machine learning, each dates’ articles will be compiled into a paragraph or corpus. Therefore, the dates with less than 25 articles, will have slightly shorter paragraphs compared to other dates, but, this will still allow for NLP and machine learning to be possible.

Probably there would be a method where each article could be independently assessed. If this was the case, then, there would be issues in doing further processing.

# Approach

## **Data Collection**

This is the process of gathering the data in raw format to enable the collector of data to conduct tests.

The data was acquired from a public source (Kaggle.com) with Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) which allows the collector to share and adapt. The source of data was scraped DJIA data from Yahoo Finance and retrieve the news headlines from Reddit.

### Creating the class label using DJIA dataframe

For an additional challenge, the RedditNews.csv and DJIA\_table.csv files were combined and manipulated to become CombinedNewsDJIA.csv. This allowed for a learning opportunity and also allowed for greater capacity for tweaks in the class label if required to do so.

The was done by reading the .csv files and declaring them as variables (dataframes). The DJIA dataframe was the focus in creating the class label. First off, a copy was created. Initally, the code used was djia2 = djia. Though djia2 was created, it was later discovered that when changes made to djia2 also affected djia. Eventually, a solution was found. When a copy is made, .copy() command was crucial in creating an independent dataframe where changes made would not impact the former database. Therefore, the code would be djia2 = djia.copy(). This as of result allowed manipulation/changes to the new dataset while retaining the old one; which can be recalled.

Once an independent copy was made, the dates needed to be sorted and index restarted. This was because when the dates were sorted, the unchanged indexes would also appear.

With the sorted data, a new column was created and differences were calculated to record the changes made during the day. For instance, in Day 0 a metric was at 1000 and Day 1 was at 1100; we want to record +100. This was done using the .diff() method.

The result was a column with value changes from the previous day. There were an array with positive and negative floats.

The next step was to organize the numbers into 1 or 0 categories (>= 0 meant 1 and <0 meant 0). This was done with a conditional statement. This conditional statement had left the first class label to be 1. However, this was supposed to be 0. Therefore, the first class label was the exception and was arbitrarily changed to 0.

Upon inspection of the class label, it was found that the label datatype was ‘object’. This was incorrect and was casted to integer format using code ‘.astype(‘int).’

### Reorganizing Reddit News Dataset

Next step was working with the Reddit News dataframe.

Initially, the dataframe had two columns: Date, News

This had to be transformed into a format where the Date is compiled into 1 unique date with the corresponding articles on the horizontal axis.

|  |  |
| --- | --- |
| Date | News |
|  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Top 25 Reddit News Articles | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

This transformation was difficult. Initially, the plan was to combine the data using Pandas commands exclusively. The road block was that as explained in EDA section, the data had null values and it appeared that the shape had to be ‘rectangular’ without null values for the code to work in Pandas. Therefore, the work around was to use loops and lists. From the initial exploration, it was found that none of the dates exceeded 25 and just few dates had less than 25.

The loop was designed so that it would iterate over all the articles. It would store it in a list. When the loop notices a change in the date, the list would be inputted towards a list. Therefore, becoming list of list. The dates were also stored in a list as well. Once the necessary lists were created, a dataframe was created with the columns and the lists were outputted.

### Merging Reddit News and DJIA Datasets

Then the modified news and modified djia dataframes were inner merged with djia being the initial dataframe, merging on dates column.

After that, the unnecessary columns were dropped to be pruned with the required columns.

Unfortunately, even when the data was merged correctly, there were some missing articles (data). However, as explained prior, the amount of data missing was negligible and were able to continue on with the process without issues.

## **Data Exploration**

This is the step where users explore the dataset to uncover initial patterns, characteristics and points of interest. This process helps to create bigger picture of important trends and major points to conduct analysis.

When the data is split into training and test datasets, these will be the same for all the testing methods so that they can be standardized and compared.

The dataset provider mentioned about splitting the dataset:

"For task evaluation, please use data from 2008-08-08 to 2014-12-31 as Training Set, and Test Set is then the following two years data (from 2015-01-02 to 2016-07-01). This is roughly a 80%/20% split."

Once the data were split, the counts and shape were looked into. The training dataset had 1611 rows with 864 “1” and 727 “0”. The testing dataset had 378 rows with 193 “1” and 185 “0”. The ratios for 1s and 0s for training and testing datasets respectively were 1.19 and 1.05. There weren’t too much deviations in the ratios, therefore, rebalancing was thought not to be necessary.

This portion for DJIA, Reddit and combined datasets are more comprehensively described in Section 2. “Data Description and EDA (Exploratory Data Analysis”

## **Feature Selection and Preprocessing**

Feature selection is the process of reducing the number of input variables when developing a predictive model. The reasons for feature selection can be: to reduce computational cost of modelling, improve performance of the model. In NLP context, this would be the process of selecting what we think is worthwhile in our documents.

### Attribute Selection

In the DJIA, only the adjusted close price was retained as this is the class label that will be used for analysis. The changes of the class were denoted with ‘1’ and ‘0’ for simplicity for classification model.

In the Reddit data, the dates and news headlines were pivoted and collapsed for simplicity. Therefore, each row corresponds to a date and is followed by 25 columns of headlines.

The final cleaned dataset is where there are 27 columns. Date, Label, and 25 news headlines.

The combined file had the necessary cleaned data to have the data put into next steps. For training and testing purposes, essentially, there will be 2 attributes that will be of concern: the articles for a date would be compiled into a paragraph, and the class label.

## Text Preprocessing

Removing b’ ‘ and b” “ encasing.

When looking at texts, there were encasing on each sentences. The reason is still to be discovered as to why the sentences were encased with a ‘b’ character with punctuation encasing but this provided no benefit in text analysis but just additional noise/data. Therefore, these were removed using regex and .replace() method. This regex methods did not impact the text within the encasing and the sentences still retained its original structure in words and punctuations. This was then saved into ‘result’ database since all future attempts will not be requiring the encasing.

result.replace("b'|b\"", "", regex = True, inplace = True)

result.replace("\"|'$","", regex = True, inplace = True)

Removing punctuation

The punctuation is removed because classifiers would count punctuation as a separate word. This is not beneficial since punctuations might convey meaning in text but not necessarily in computational textual analysis. Therefore, these can be removed.

However, there needs be some thought in removing punctuations. Replacing them with space turns words like U.S. and U.N. into ‘U’,”S’ and ‘U’, “N’. Where these individual characters would be considered a word and would be lost in analysis. Also, hyphenated word like ‘cross-examination’ are of concern. They did exist in the documents but removing the punctuation out weighted the costs.

Lowercase/Uppercase

This is important since the computer can not differentiate between ‘Car’ and ‘car’. Instead of classifying them as a same word, they will be registered as separate words. An easy solution to this problem is to convert the text in the corpus lowercase or uppercase. This step should be done prior to removing stop words since when removing, you would want all instances to be removed (this would not be possible if same stop words are in different cases). With coding, you can select the dataframe and place a .lower() method to the text. It was also later discovered that some classifiers automatically convert to lowercase by default, so coding was not entirely necessary for this portion.

Stop Words

Stop words are common words such as ‘a’, ‘and’, ‘but’, and etc. In NLP, these are words which are ignored/removed. NLTK (Natural Language Toolkit) and Scikitlearn has packages which enables users to remove stop words. Also, classifiers such as CountVectorizor() also has parameters which enables to remove stop words but the list seems limited compared to other methods. Numerous methods have been attempted but was not able to verify which stop removal strategy is optimal.

Stemming / Lemmatization

Stemming is the processing of variants of a word into its root form. For instance, converting ‘likes’,’liked’,’likely’ to ‘like’. However, this process is not perfected at this point since over and under stemming exists. To simply the program, it was decided not to use stemming but if desired, PorterStemmer from NLTK library can be used.

## **Feature Engineering and Model Building**

Once the text data has been cleaned and preprocessed, The features for the data structures and vectorizes the data which then can be understood by the machine learning algorithm. Some of these techniques can include Bag of Word, TF-IDF using tools like Word2Vec, GloVe and etc.

BOW (Bag of Words) Model

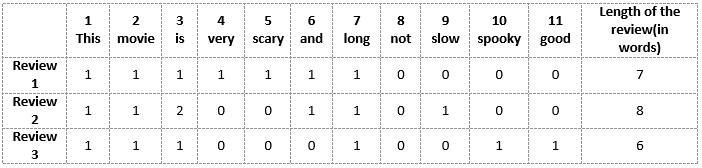
Unlike humans, where language is intuitive, for machines, the text needs to be converted to numbers for a computer to understand. This model is simple in that the order of words do not matter and it rather focuses on the occurrence of words.

First, each word is assigned a unique number. Then the text is encoded in a fixed length vector and the value in each position in the vector could be filled with a frequency of a word in the encoded document.

A visualization is represented at the bottom:

https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/

* Review 1: This movie is very scary and long
* Review 2: This movie is not scary and is slow
* Review 3: This movie is spooky and good

[](https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/BoWBag-of-Words-model-2.png)

Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]

Vector of Review 2: [1 1 2 0 0 1 1 0 1 0 0]

Vector of Review 3: [1 1 1 0 0 0 1 0 0 1 1]

CountVectorizer() was commonly used in my NLP for BOW model.

The drawbacks of this model is that if new sentences contain new words, then the vocabulary size would also increase and as of result also increase the size of the vectors. Many times, vectors would contain many 0s causing a sparse matrix to occur. Also, as mentioned before, no data is retained regarding the ordering of words or grammar of the sentences. This was the case weith my project since some vectors reached over 300,000 words.

TF-IDF Model

Another technique to convert text into numeric vectors (tokenization). This method allows to find how important a word is to a document in a corpus. First, a term frequency matrix like BOW model is determined. Then, the IDF value is incorporated with formula IDF = log(total # of documents/# of documents containing that particular word). The TF and IDF values are multiplied together to get a value; the higher the score, the more important the word would become. TF-IDF gives higher score for less occurring words and also when both IDF and TF are high (ex. word is rare in all of the documents combined but frequent in a single document).

N-gram

n =1 is unigram, n=2 is bigram and so on. In unigram, the machine will assume that the occurrence of each word is independent of the previous word. In bigram, we assume that each occurrence of each words depends only on its previous word.

CountVectorizer() and TfidfVectorizer() found in SciKitLearn package enables to convert the text into a matrix and also has parameters to control for N-grams.

Classifiers to Test the Model

Here are the list of classifiers that were used to test the machine learning models:

Logistic Regression, K Nearest Neghbors, SVC (Support Vector Machines), Decision Tree, Random Forest, and Gaussian Naïve Bayes.

The codes and results can be found at Github:

<https://github.com/QOneK/Ryerson-Data-Analytics-Final-Project-for-Kyuhwan-Kim>

# Results

In a python environment, the confusion matrix does not indication as to where the TP,TN, FP, FN are.

Therefore, upon searching, a discovery was made on how python organizes a confusion matrix.

<https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781838555078/6/ch06lvl1sec34/confusion-matrix>

|  |  |  |
| --- | --- | --- |
|  | Predicted | |
| Actual | **0** | **1** |
| **0** | TN | FP |
| **1** | FN | TP |

Above is a chart called **confusion matrix** which describes the performance of the classification model to summarize the classifier performance.

**Accuracy** = (TP + TN) / (TP + TN + FP + FN)

**Precision** = TP / (TP + FP)

Out of all predictive positive classes, how many we predicted correctly? High precision desired.

**Recall** (Sensitivity / TP Rate) = TP / (TP + FN)

Out of all actual positive classes, how many have we predicted correctly? High recall desired.

**F – Score** = (2 \* Recall \* Precision) / (Recall + Precision)

Difficult to compare two models with different precision and recall. Therefore, to compare, we use F Score.

Perfect precision and recall =1 , worst F Score would be 0.

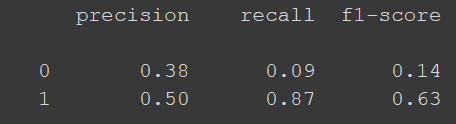
Prior to each of the attempts, the text was preprocessed and prepared for machine learning.

For the first attempt, I have used CountVectorizer() with unigram settings to build the model and RandomForestClassifier to test the model. The predicted model resulted in 42 guesses for ‘0’ and 336 guesses for ‘1’.

**Test 1**

|  |  |  |
| --- | --- | --- |
|  | Predicted | |
| Actual | **0** | **1** |
| **0** | 16 | 169 |
| **1** | 26 | 167 |

Accuracy = 48.4%



For test 1,

Precision was 0.38, Recall was 0.09 and F Score 0.14 for guessing 0

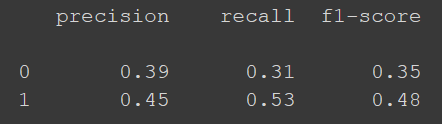
Precision was 0.50, Recall was 0.87 and F Score 0.63 for guessing 1

**Test 2**

For the second attempt, I have used CountVectorizer() to tokenize and train. I let the tool to remove the punctuations by itself and used LogisticRegression to test the model. The predicted model resulted in 149 guesses for ‘0’ and 229 guesses for ‘1’.

|  |  |  |
| --- | --- | --- |
|  | Predicted | |
| Actual | **0** | **1** |
| **0** | 58 | 127 |
| **1** | 91 | 102 |

Accuracy = 42.3%



For test 2,

Precision was 0.39, Recall was 0.31 and F Score 0.35 for guessing 0

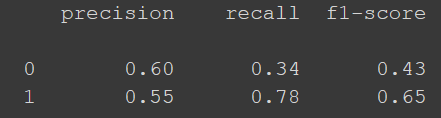
Precision was 0.45, Recall was 0.53 and F Score 0.48 for guessing 1

**Test 3**

For the third attempt, I have used CountVectorizer() to tokenize and train. Used ngrams=2 within the parameters of the tool. LogisticRegression to test the model. The predicted model resulted in 62 guesses for ‘0’ and 316 guesses for ‘1’.

|  |  |  |
| --- | --- | --- |
|  | Predicted | |
| Actual | **0** | **1** |
| **0** | 62 | 123 |
| **1** | 42 | 151 |

Accuracy = 56.3%



For test 3,

Precision was 0.60, Recall was 0.34 and F Score 0.43 for guessing 0

Precision was 0.55, Recall was 0.78 and F Score 0.65 for guessing 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Analysis of 3 Tests**

At first glance, it can be tempting just to pick on the test with the highest accuracy. But, definitely, more inspection will be required to compare these tests.

When comparing test1 and test2, test1 has a higher accuracy and higher F Score for predicting 1. However, upon more scrutiny, test1 has very low score for predicting 0 (at 0.14). This result was the case because test 1 was predominately guessing 1. From previous observations, we have observed that there are more 1s ( and that the DJIA is on an upward trajectory “The testing dataset had 378 rows with 193 “1” and 185 “0”.

Therefore, if the computer was to be ‘lazy’ and guessed each and every label as ‘1’, this would result in an accuracy score higher than 50% with high F Score for predicting ‘1’. However, this would be at the cost of sacrificing the scores for predicting ‘0’s; which was the case for test 1 with a very low F Score for ‘0’. Therefore, when comparing test1 and test2, though test1 has higher overall accuracy, test 2 is the better choice with more ‘balanced’ F Scores.

Test3 is when the model gets much more accurate and better. Using ngrams and LogisticRegression (compared to RandomForest), the model was able to predict 0 and 1 with the highest accuracy and F Score compared to the other models. Therefore, we can conclude that through the progressions, the model has been improving. However, with accuracy of 56%, though it is not an optimal percentage, this is a much better figure than blatantly guessing 50/50 whether the stock will go up/down.

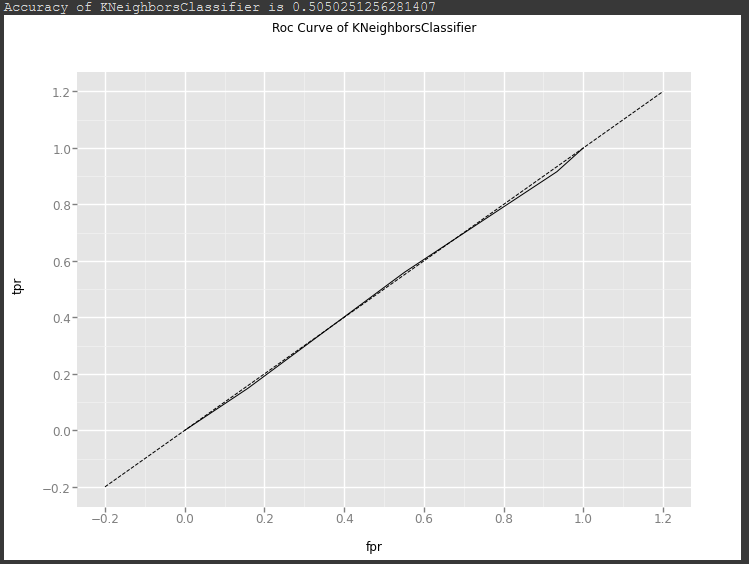
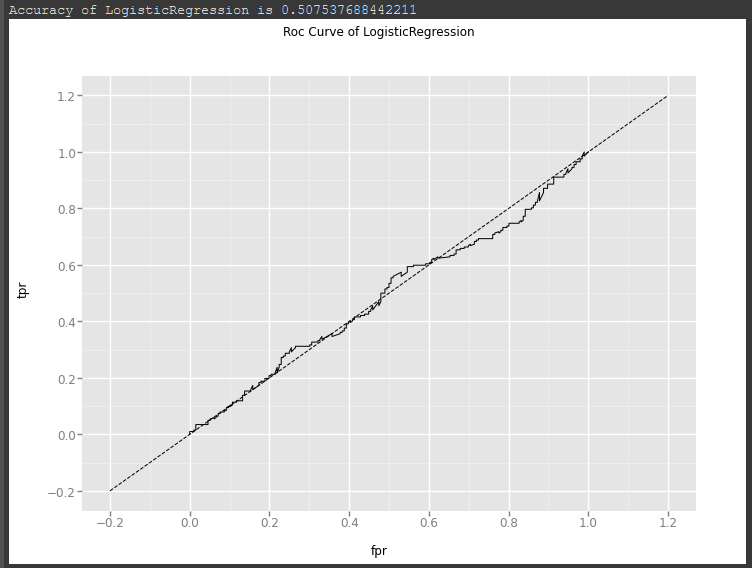
**Analysis of Test4**

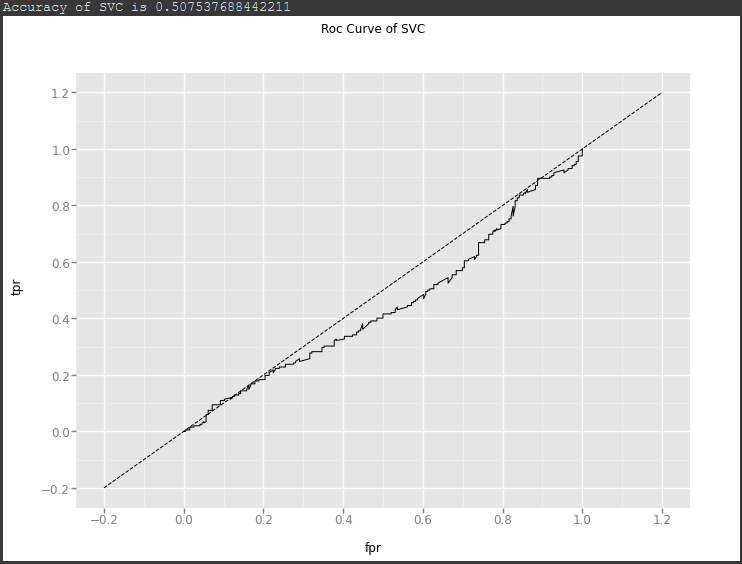
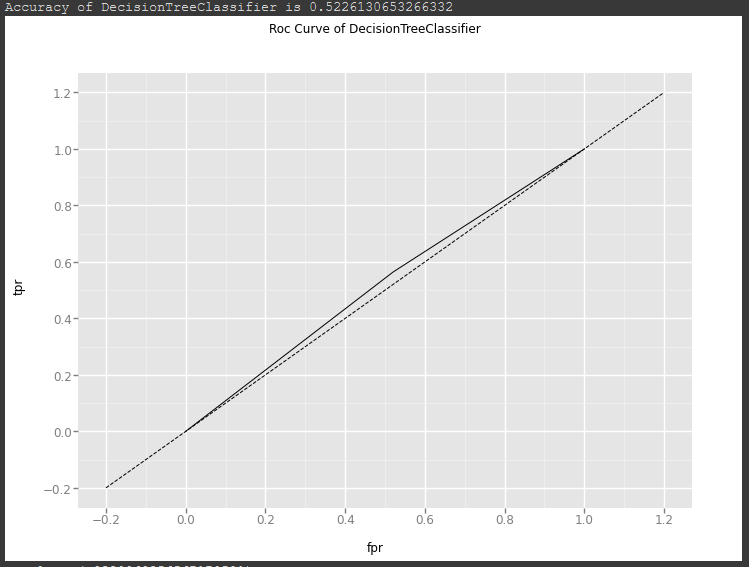
Another attempt was made with the code. This time, to observe the ROC curves. This used an unconventional package ggplot. Therefore, for Test4, the current version of Pandas is uninstalled with an earlier version of Pandas and other tools are installed as normal (see the code for more details on section test4).

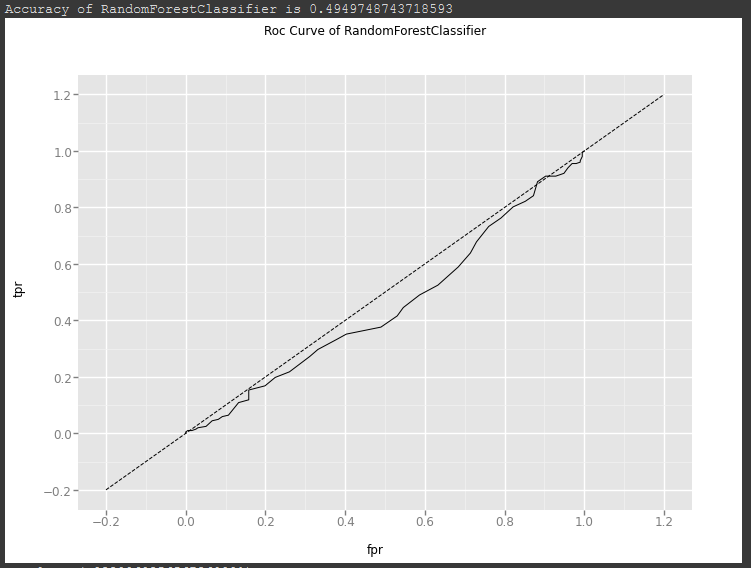
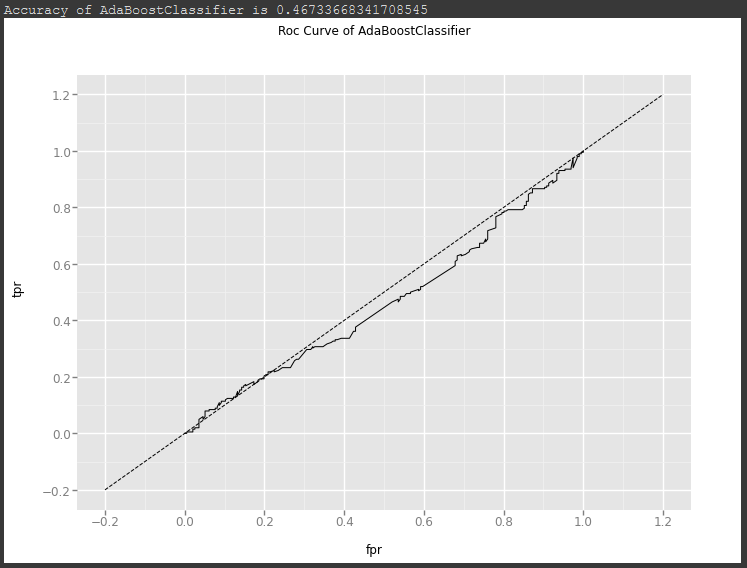
What is AUC – ROC Curve? It stands for Area Under the Curve Receiver Operating Characteristics Curve. This is a visual performance measurement on a classification problem. The enables how much the model is capable of distinguishing between classes. Higher the AUC, better the model is predicting 0s and 1s. The y-axis is TPR (True Positive Rate) or Recall/Sensitivity and FPR (False Postive Rate) or 1 – Specificity.

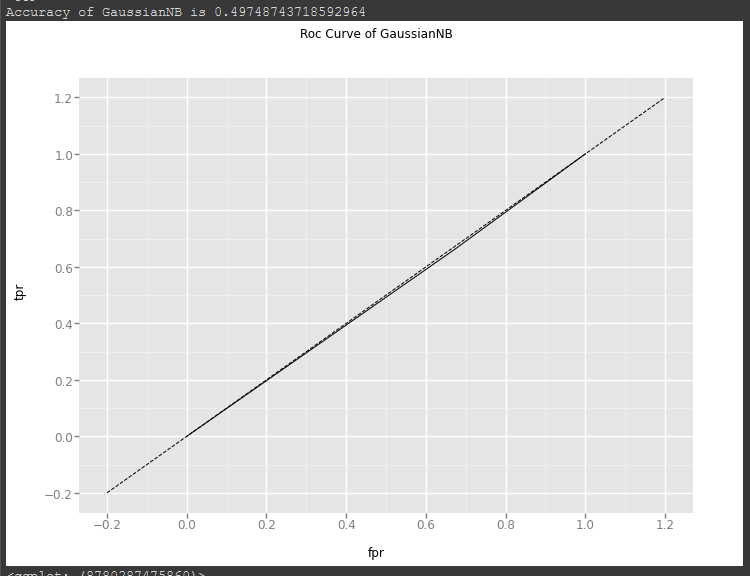
An excellent model has AUC closer to 1, poor model has near to 0. When AUC = 0.5, it means the model has no class separation capacity.

This coding component was something that was beyond my understanding of Python coding. I would like to credit Jason Liu for coding to have ROC outputs https://www.kaggle.com/jiashenliu/presenting-seven-classifiers





In our models, we observe a lot of models where AUC is approximately 0.5 which means that the model has no discrimination capacity to distinguish between positive class and negative class.

In fact, some curves are actually getting reciprocating results which means that the model is predicting negative (0) class as a positive class (1) and vice versa.

This result makes sense because in test1, we have observed that a lot of 0s has been incorrectly labelled as ‘1’. Therefore, it appears that no matter which model we choose to go with, we would not be achieving a high ROC curve.

In conclusion, we can assume from the results seen so far that the models (or any model experimented so far) would not have good predicative ability.

From visualizations, it appears that Logistic Regression and Decision Tree would be the best classifier to use for this dataset since this was the only curve that has attained small positive results. Of course, to achieve this result, some tweaking within the tools would be required and that seems to have been achieved with test3 (which coincidently used Logistic Regression).

The codes and results can be found at Github:

<https://github.com/QOneK/Ryerson-Data-Analytics-Final-Project-for-Kyuhwan-Kim>

# Conclusions

This project has explored the topic of NLP (Natural Language Processing) in Python programming environment using Google Colabs using various dependencies (NLTK, SciKitLearn), classifiers (Linear Regression, Random Forest, Logistic Regression and etc) and methods like N-grams.

There were challenges such as errors in coding, noticing small details such as the way DJIA data was order asc or desc, methods in implementing and assessing NLP results. Though not an absolute conclusion can be provided at this time, much data analytics and NLP techniques were achieved through this project. Critical thinking was definitely required towards data analytics.

Each characteristics appears to have an impact in the final result. Sometimes, the result can not be taken quantitatively at face value such as test1 vs test2 on their accuracy and F scores. Even though test1 had higher accuracy and F scores for predicting ‘1’; that was not conclusive to decide that test1 was indeed better than test2. In this case, both ‘0’ and ‘1’ scores for accuracy and F Scores are needed to be taken into account. Another good measurement to visualize the effectiveness of tools would be ROC curves. This enabled visually to determine which tool would be an optimal choice and have an outlook as to what confusion matrix results we can expect.

In conclusion, with the experimentations that were done so far, we can assume that the Reddit News had no relation towards the movement of the DJIA.

For future revisions on this study, perhaps experimenting with shift in dates and the news might be of help. If the stock market was to react to the after math of a news, then, the news would logically have to be distributed and understood by the people for there to be market reaction. Therefore, shift in the distribution of news and stock market would have to be experimented to 1 or 3 days to see if this hypothesis would hold.

Also, another condition to experiment would be the type of news. The news dataset in this case was on Reddit users’ ranked top 25 news articles for the day. DJIA (Dow Jones Industrial Average) is an American stock exchange index. Therefore, some news in a different part of the world might not necessarily have an impact on the DJIA. Hence, financial news about a company about a company compared to the described company’s stock price would be a much more direct comparison. Another hypothesis is that a more related financial news compared to a more related stock index would have a much more ROC curve and predictive ‘power’.

# References

Anthony (Tony) Cox, L., Jr. (2017), Misbehaving: The Making of Behavioral Economics by Thaler, Richard. Risk Analysis, 37: 1796-1798. Doi:[10.1111/risa.12871](https://doi.org/10.1111/risa.12871)

G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller. 2015. Scikit-learn: Machine Learning Without Learning the Machinery. *GetMobile: Mobile Comp. and Comm.* 19, 1 (January 2015), 29–33. DOI:https://doi.org/10.1145/2786984.2786995

Kavšek, B. (2017). Using words from daily news headlines to predict the movement of stock market indices. *Managing Global Transitions, 15*(2), 109-121. Doi:10.26493/1854-6935.15.109-121

Madnani, N. (2007). Getting started on natural language processing with Python. *XRDS: Crossroads, The ACM Magazine for Students*, *13*(4), 5-5.

Nguyen, T. H., Shirai, K., & Velcin, J. (2015). Sentiment analysis on social media for stock movement prediction. *Expert Systems With Applications*, *42*(24), 9603–9611. <https://doi.org/10.1016/j.eswa.2015.07.052>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *The Journal of machine Learning research*, *12*, 2825-2830.

Thorat, S., Deshpande, P., & Shaga, V. (2017). REVIEW OF SENTIMENT ANALYSIS ON TWITTER DATA USING PYTHON. *International Journal of Advanced Research in Computer Science, 8*(9) Retrieved from <http://ezproxy.lib.ryerson.ca/login?url=https://search-proquest-com.ezproxy.lib.ryerson.ca/docview/1980479225?accountid=13631>

Yamashita, Y., Joutaki, H., & Takahashi, H. (2013). Analysing the influence of headline news on the stock market in japan. *International Journal of Intelligent Systems Technologies and Applications, 12*(3-4), 328-342. Doi:10.1504/IJISTA.2013.056539

# Common NLP Terminology

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| Sentiment Analysis | Text based supervised machine learning classification task with a phrase and the classifier tells if the sentiment(view/attitude) behind that is positive, negative or neutral. Sometimes, third attribute is not taken to keep it a binary classification problem. Model is tested on unlabeled phrases.  Presence > Sequence of words. |
| Word | Atomic entity within a *document* |
| Document | Large sequence of words [tweets: short size, article: larger size] |
| Feature Selection | Finding out most relevant features that relate to the *class label* |
| Bag of Words | First step of doing any text classification problem. Frequency of occurrence of relevant words.  Representation not only of specific words but *unique* words and their frequencies of occurrence. |
| Corpus | Feature set found from bag of words. Rows are feature vectors |
| Classifiers | Programs that automatically analyze text and then assign a set of predefined tags/categories based on its content |
| NLTK | Natural Language Toolkit. Open sourced. |
| Tokenization | Text -> individual words |
| Stemming | Cutting prefix/suffix ex. [booing, booed] -> boo |
| Lemmatization | Cut word into root word ex. [am,are] -> be |
| Stopwords | Common words that are taken out without changing mean to a phrase |
| Parsing | Stage of NLP concerned with segmenting text based on syntax |