Stock Market prediction using news headlines

by

Adithya Job

Shravan Chintha

Table of Contents

Abstract:

It has always been an alluring dream for the quants in Wall street to predict the stock market by looking at the historical records. Active research has been going in this field for many years but because of the erratic nature of the market a mature solution has not been formulated yet. The reason for failure in this domain is because the researchers have pointed out that some un-quantifiable forces play a major role in the trend of the stock market. The un-quantifiable factors could be a political development, policy changes or some time related event. Surprisingly the news tabloids are an interesting collection of records of such event which are also ordered chronologically and indexed to a great extent. Our project is an attempt to analyse these news top headlines to find if there is a co-relation between the contents of such top headlines to the performance of the stock. We will be analysing word patterns and sentiment of the top headlines to predict the performance of the stock market.

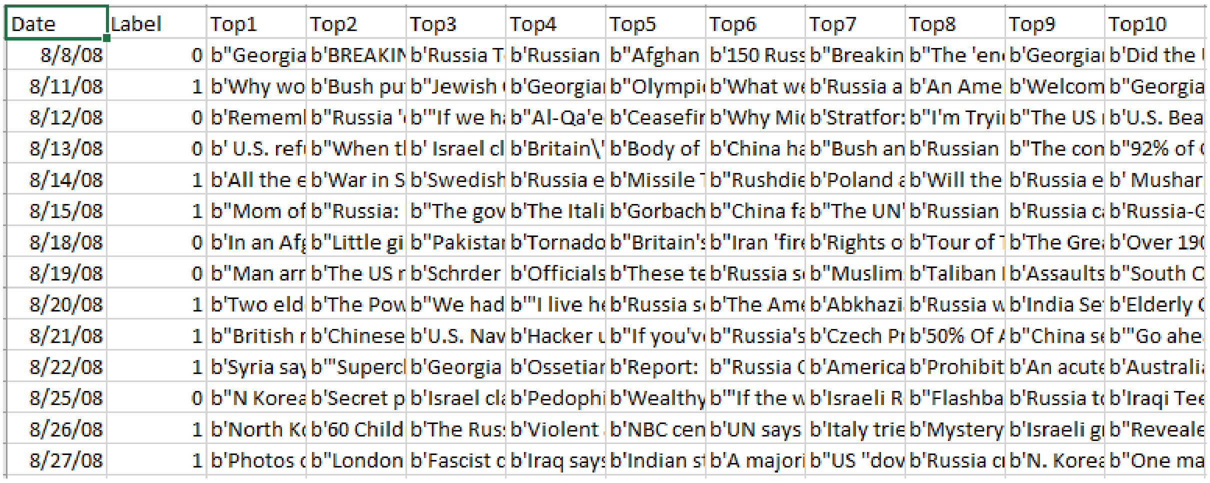
Introduction:

The project is about predicting the stock market movement based on the news headlines that published on a particular day. The news data is collected from Reddit news and top 25 headlines, ranked based on reddit user votes, are taken on each day. The stock market data, DJIA (Dow Jones Industrial Average) of each day is collected from Yahoo finance. Combined both datasets to process and apply modeling techniques further to get desired results.

Natural language techniques such as word clouds, bag of words, ngrams, sentiment analysis etc., are used to process the data. Also, machine learning techniques such as logistic regression, random forest, Naïve Bayes model, gradient boosting, xgboost are applied to predict the outcome variable. A baseline model, logistic regression with bag of words is performed to check the accuracy and use it as a baseline for our rest analysis. Then, computed accuracies of various model to know the best performing model among the models we applied. We got the best accuracy for sentiment analysis with xgboost algorithm, able to improve the accuracy to 62.7% compared to the baseline 46.07%.

Dataset:

The news data and stock market DJIA data are combined to form a single dataset that contains the date the news published on, label i.e., performance indicator of DJIA and Top 1 to Top 25 headlines corresponding to that day. Headlines are ranked from 1 to 25 based on the number of votes they receive from Reddit users. The dataset contains 1989 records with dates ranging from 08-08-2008 to 07-01-2016. Label is defined as “0” if the closing value of DJIA has decreased from previous day’s close, and as “1” if the closing value of DJIA has increased or stayed same as previous day’s close. The data is balanced with a data split of 47/53, with 925 records labeled as “0” and 1064 records labeled as “1” in the total 1989 records present in the data.



Date	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10
8/8/08	0	b"Georgia	b'BREAKIN	b'Russia T	b'Russian	b"Afghan	b'150 Russ	b"Breakin	b"The 'en	b'Georgia	b'Did the
8/11/08	1	b'Why wo	b'Bush pu	b"Jewish	b'Georgia	b"Olympi	b'What we	b'Russia a	b'An Ame	b'Welcom	b"Georgia
8/12/08	0	b'Remem	b"Russia '	b'"If we h	b"Al-Qa'e	b'Ceasefir	b'Why Mi	b'Stratfor:	b"I'm Tryi	b"The US	b'U.S. Bea
8/13/08	0	b' U.S. ref	b"When t	b' Israel c	b'Britain\'	b'Body of	b'China ha	b"Bush an	b'Russian	b"The con	b"92% of
8/14/08	1	b'All the e	b'War in S	b'Swedish	b'Russia e	b'Missile	b"Rushdie	b'Poland a	b'Will the	b'Russia e	b' Mushar
8/15/08	1	b"Mom of	b"Russia:	b"The gov	b'The Itali	b'Gorbach	b"China fa	b"The UN'	b'Russian	b'Russia c	b'Russia-G
8/18/08	0	b'In an Af	b"Little gi	b"Pakista	b'Tornado	b"Britain's	b"Iran 'fir	b'Rights o	b'Tour of	b'The Gre	b'Over 19
8/19/08	0	b"Man arr	b'The US r	b'Schrder	b'Officials	b'These te	b'Russia s	b"Muslim	b'Taliban	b'Assaults	b"South C
8/20/08	1	b'Two eld	b'The Pow	b"We had	b'"I live h	b'Russia s	b'The Am	b'Abkhazi	b'Russia w	b'India Se	b'Elderly
8/21/08	1	b"British	b'Chinese	b'U.S. Nav	b'Hacker	b"If you'v	b"Russia's	b'Czech Pr	b'50% Of	b"China se	b'"Go ahe
8/22/08	1	b'Syria say	b'"Superc	b'Georgia	b'Ossetiar	b'Report:	b"Russia (	b'America	b'Prohibit	b'An acute	b'Australi
8/25/08	0	b"N Korea	b'Secret p	b'Israel cl	b'Pedophi	b'Wealthy	b'"If the w	b'Israeli R	b"Flashba	b'Russia t	b'Iraqi Tee
8/26/08	1	b'North K	b'60 Child	b'The Rus	b'Violent	b'NBC cen	b'UN says	b'Italy trie	b'Mystery	b'Israeli g	b"Reveale
8/27/08	1	b'Photos	b"London	b'Fascist c	b'Iraq says	b'Indian s	b'A majori	b"US "dov	b'Russia c	b'N. Korea	b"One ma

Fig 1. Combined news and DJIA dataset

Data Pre-processing:

Sample news headline from the dataset looks like this:

b"Georgia 'downs two Russian warplanes' as countries move to brink of war"

The news headline contains some extra information which we do not need or not desirable for text processing. For example, in the above news, letter b, double quotes, single quote, upper case letters are all present which must be removed in order to further process the text and apply our text classification models. There are many such headlines present in the data, which have similar issues as the headline show.

To address these issues in the data, before applying models we pre-processed the data by combining all the sentences or headlines in a row i.e., top 1 to 25 news and the used a function called CountVectorizer from SciKit learn package of python. This function does the following:

- Removes extra characters and meaningless words. Such as quotes, extra letters coming between (‘b).
- Converted the words into lower case letters.
- Divides the sentences into words and created a table with count of those words.

Exploratory Data Analysis:

To identify different types of words that are present in the data, we performed exploratory data analysis. To know the words that are frequently repeated or appeared the most in headlines, constructing their word clouds are one of the best methods. So, word clouds are made to know the most important words that appear in the headlines in terms of their frequency.

To construct word clouds, separated the data into two parts based on the label 1 or 0. All the words in the news with label 1 are made into a positive word cloud and all the words in the news with label 0 are made into a negative word cloud. The word clouds are as below:

Positive word cloud:

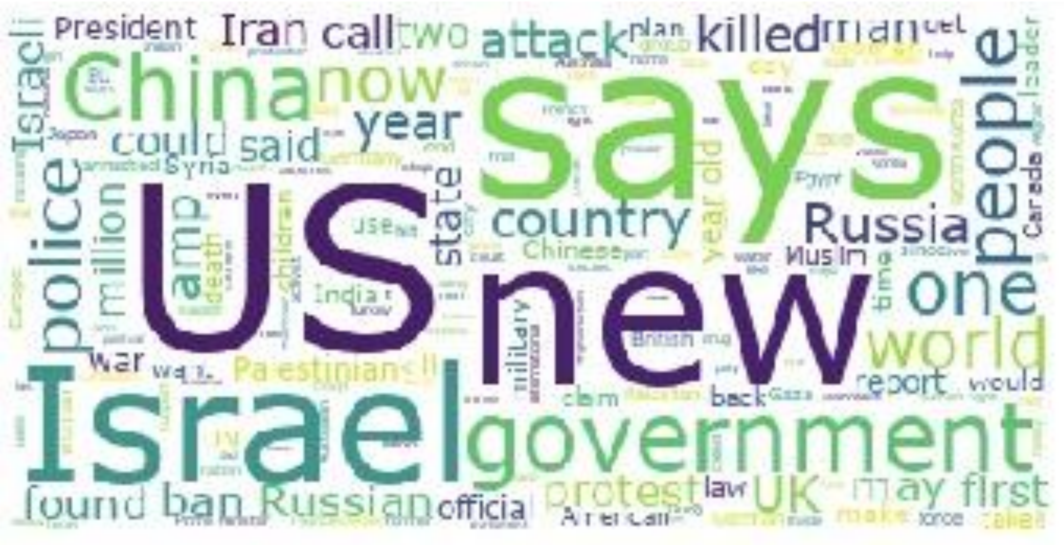


Fig 2: Positive word cloud

The word cloud shows the importance or highly used words as positive words. Words like US, says, new, Israel are some of the most frequent positive words.

Negative word cloud:

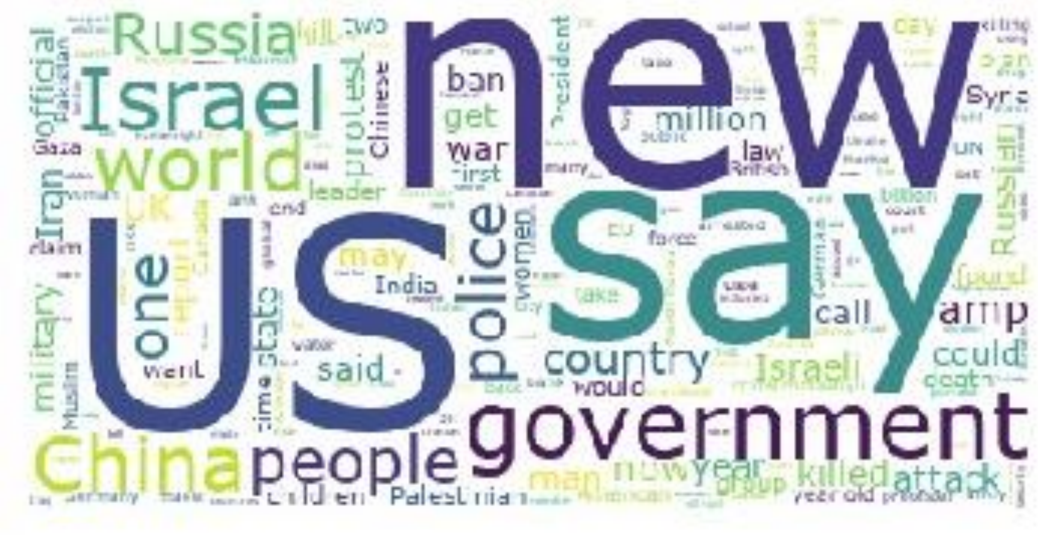


Fig 3. Negative word cloud

The word cloud shows the importance of highly used words as negative words. Words like US, government, police, China are some of the most frequent negative words.

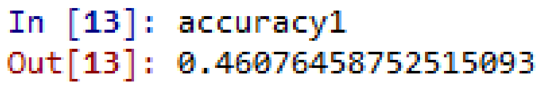
Baseline Model:

To compute a baseline model, we used logistic regression model with bag of words technique. Before applying this model, we divided the data into training and testing sets with 75/25 split. Since the data is based on dates from 2008 to 2016, data is divided into train/test sets based in date. Training set is defined from dates 08/08/2008 to 07/14/2014, test set is defined from dates 07/15/2014 to 07/01/2016. That is 1492 records in train set and 497 records in test set.

Logistic Regression with bag of words:

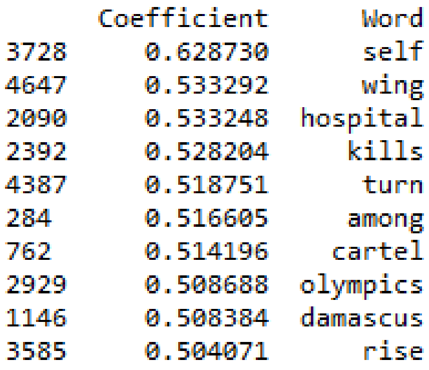
Logistic regression is a statistical method that analyzes the data, in which there are one or more independent variables determines the outcome where the outcome has only two possible values. For example, classifying a boy or girl, binary digits 0 or 1 etc.

We used CountVectorizer function to divide the sentences into bag of words, and applying logistic regression model gave the below results:

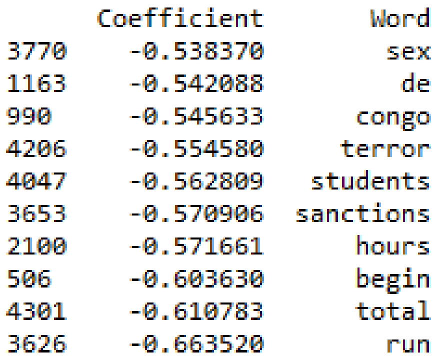


```
In [13]: accuracy1
Out[13]: 0.46076458752515093
```

An accuracy of 46.07%, which is very less. And based on the model coefficients, top 10 negative words and top 10 positive words are identified.



```
      Coefficient      Word
3728     0.628730      self
4647     0.533292      wing
2090     0.533248  hospital
2392     0.528204     kills
4387     0.518751      turn
284      0.516605     among
762      0.514196    cartel
2929     0.508688  olympics
1146     0.508384  damascus
3585     0.504071      rise
```



```
      Coefficient       Word
3770    -0.538370        sex
1163    -0.542088         de
990     -0.545633      congo
4206    -0.554580     terror
4047    -0.562809   students
3653    -0.570906  sanctions
2100    -0.571661      hours
506     -0.603630      begin
4301    -0.610783      total
3626    -0.663520        run
```

Fig 4. Top 10 positive words

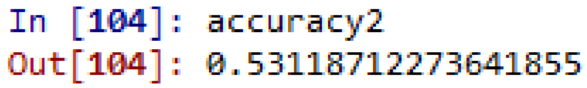
Fig 5. Top 10 negative words

Then it is understood that, accuracy can be increased if better methods are used. Then we tried using logistic regression on bigrams using TfidfVectorizer.

Logistic Regression with bigrams:

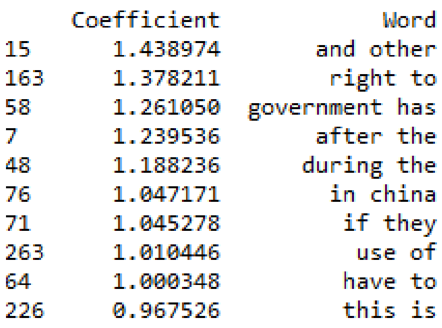
Converted the sentences into bigrams using TfidfVectorizer function. It is Term frequency times inverse document frequency. It is used to offset the frequency generated by the most repeated words yet very less meaning (such as ‘a’, ‘the’, ’is’ etc.,) in a large corpus by re-weighting the count features to values suitable for using in the classifier.

The results with bigrams and tfidf approach are:



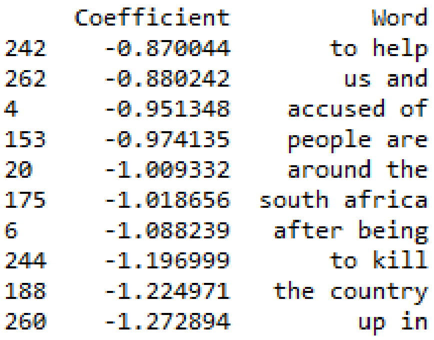
```
In [104]: accuracy2
Out[104]: 0.5311871227364185
```

Accuracy has been increased from 46% to 53%, about 7% improvement in the accuracy is a good sign. And based on the model coefficients, top 10 negative words and top 10 positive words are identified.



```
     Coefficient              Word
15      1.438974         and other
163     1.378211          right to
58      1.261050    government has
7       1.239536         after the
48      1.188236        during the
76      1.047171          in china
71      1.045278           if they
263     1.010446            use of
64      1.000348           have to
226     0.967526           this is
```

Fig 6. Top 10 positive words in bigrams



```
     Coefficient           Word
242    -0.870044        to help
262    -0.880242         us and
4      -0.951348     accused of
153    -0.974135     people are
20     -1.009332     around the
175    -1.018656   south africa
6      -1.088239    after being
244    -1.196999        to kill
188    -1.224971    the country
260    -1.272894          up in
```

Fig 7. Top 10 negative words in bigrams

Bigrams are better performing than bag of words with accuracy as metric. So, trying other machine learning models on bigrams would be a better idea to check if the accuracy can be improved further. Implemented Random forest, Naïve Bayes and gradient boosting algorithms to check if there is an increase in accuracy.

Random forest with bigrams:

A Random Forest n-gram model is a collection of randomly constructed decision tree n-gram models. Random Forest models in language modeling are used to deal with the data sparseness problem.

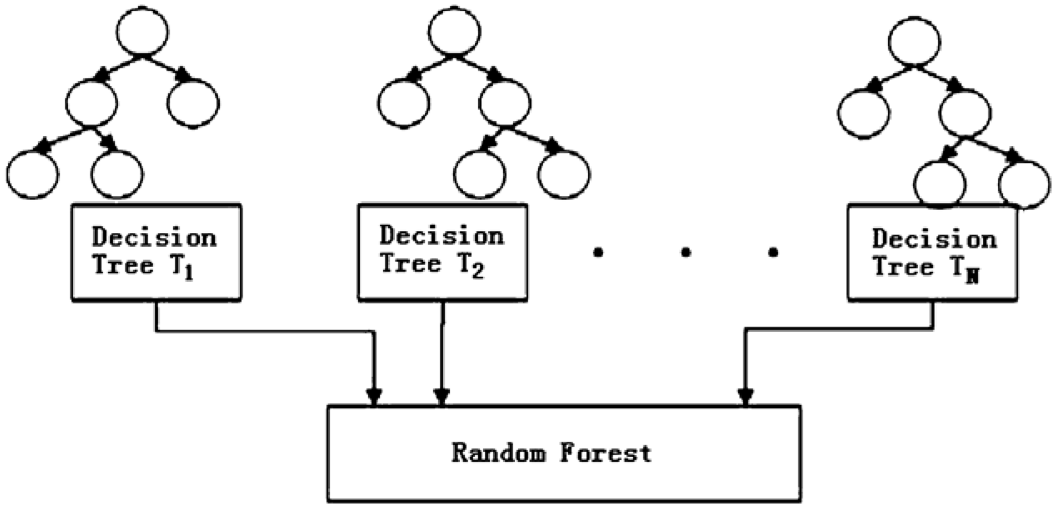
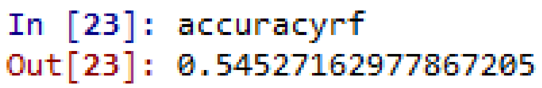


Fig 8. Random forest algorithm

Decision trees are formed by building a number of trees and taking a point as node as shown above.

Random forest algorithm is applied to the bigrams with a similar kind of approach i.e., using Tfidf vectorizer. The results are as below:

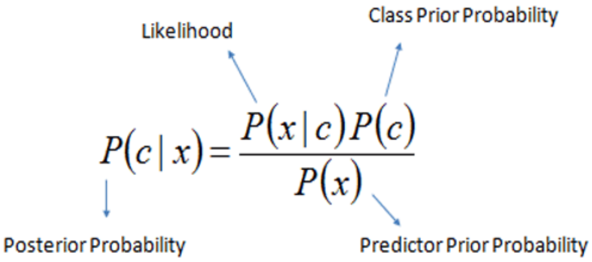


```
In [23]: accuracyrf
Out[23]: 0.54527162977867205
```

Accuracy is 54.52 now, accuracy has further improved over logistic regression using same kind of bigram approach. Applied few other generally best performing machine learning algorithms using similar approach to see if the accuracy can further be improved.

Naïve Bayes model with bigrams:

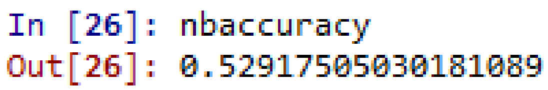
The probabilistic model of naive Bayes classifiers is based on Bayes' theorem. Naïve Bayes classifiers are linear classifiers which are known to be simple and efficient models. Naïve Bayes is one of the better models when text classification is concerned.



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

When similar approach is used on the bigrams, yielded the below results:



```
In [26]: nbaccuracy
Out[26]: 0.5291750503018109
```

Accuracy is 52.91%, dropped a bit when compared to the other two models on bigrams. Applied gradient boosting machines algorithm to check if it was a better model.

Gradient Boosting:

The gradient Boosting is a data modelling algorithm used for regression and classification task, this produces predictions with the help of models in the form of ensemble of weak learners, which are typically simple decision trees. The algorithm proceeds in a stage wise fashion where an arbitrary loss function is optimized. The weak learner in this algorithm are combined in an additive manner where the above-mentioned function is optimized. As the explanation suggests the model in this approach provides the best results. The TFID method prevents the bias of common words (stop words). The ngram in consideration is of the parameter (2,2), that is it is of type Bi-gram. The gradient boosting algorithm is taking two words at a time to see if there is relation with the DIJA.

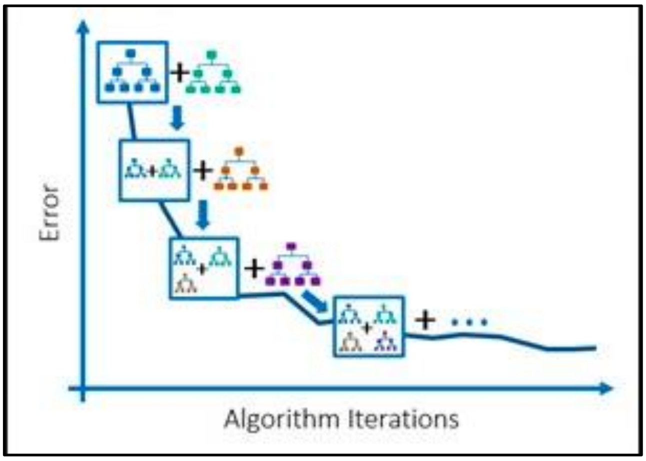


Fig 10. Gradient boosting algorithm

The accuracy we obtained in this approach was 55.33%.

Confusion Matrix:

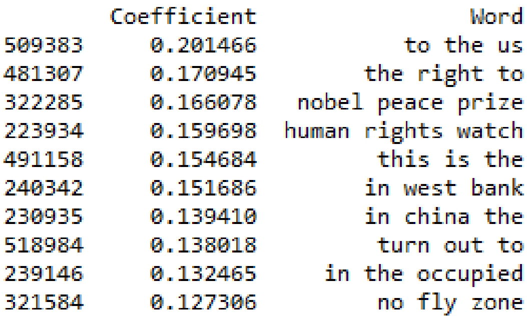
	Predicted	
observed	**0**	**1**
0	**92**	**148**
1	**74**	**183**

Table 1: Confusion matrix of Gradient boosting machines

The evaluation of model suggests that the performance of the same is distributed across the class types. The model can predict better when the stock market will perform rather than in the cases when the market will not perform. As in the beginning we have recorded our observation that this research is to find a signal but not for accurate prediction because it is impossible to model the high rate of variation of the stock market data. This model can perform better if the prediction power across all classes is improved.

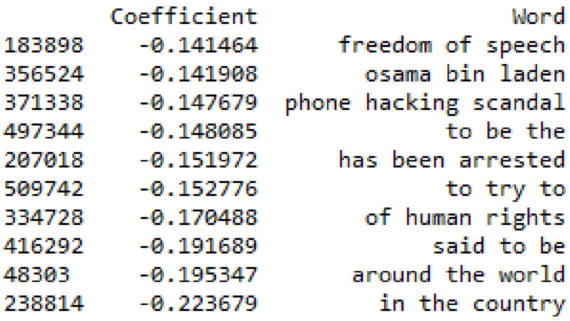
Logistics Regression with Trigram:

The logistics regression with trigram is an approach to see if the increasing range of ngrams' parameter is having a positive impact in increasing the accuracy of the model. This approach looks into a combination of three words at a time with the relationship with the stock market. A TFID vectorization approach was employed to mitigate the influence of the stop word. This approach for sure increased our performance on the training dataset but the same could not achieve in the testing data. This underperformance in the testing data is because of over fitting of the model. When the combination of three words is considered, the model variation is compromised and bias is increased. The coefficients of the model are reduced and the word vector is furthermore sparsely populated. The accuracy of this model was reduced to fifty-two percentage which discouraged us to not to explore in this direction further.



```
        Coefficient                 Word
509383     0.201466            to the us
481307     0.170945         the right to
322285     0.166078    nobel peace prize
223934     0.159698  human rights watch
491158     0.154684          this is the
240342     0.151686        in west bank
230935     0.139410         in china the
518984     0.138018          turn out to
239146     0.132465      in the occupied
321584     0.127306          no fly zone
```

Fig 11. Top 10 positive trigrams

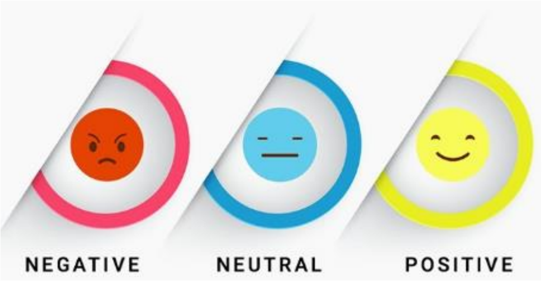


```
        Coefficient                   Word
183898    -0.141464      freedom of speech
356524    -0.141908        osama bin laden
371338    -0.147679  phone hacking scandal
497344    -0.148085              to be the
207018    -0.151972       has been arrested
509742    -0.152776              to try to
334728    -0.170488        of human rights
416292    -0.191689             said to be
48303     -0.195347       around the world
238814    -0.223679         in the country
```

Fig 12: Top 10 negative trigrams

Sentiment Analysis:

The sentiment analysis is the approach which is gaining traction in the research community. The hypothesis of this approach is that we can predict the stock market by analysis the sentiment of the top headlines published on that day. The top headlines of each day were analysed and their sentiments were extracted. To measure the sentiments of the article we got the polarity scores of the top headlines. To compute the polarity of the sentences, we assumed that the words in consideration are either positive and negative. Each time a positive word is mentioned the polarity score is incremented and if a negative score is mentioned the same score is decremented. At the end for each headline a polarity score ranging from -1 to 1 is computed and the data frame constituting non-quantifiable content is converted to numbers. This data frame is used for machine learning purposes to learn the relationship between the performance of the stock with sentiments of the top headlines of the day. The polarity score ranges from minus one to positive one.



A major problem in this approach is there are many headlines which are neutral in nature and need to mitigate this problem of bias. The accuracy we obtained was in this approach is 62.7 percentage.

Model Comparison:

All the above models are compared using accuracy as the common metric. Below is the table showing the accuracies that are achieved using various models:

Model	Accuracy
Baseline model (Logistic regression with bag of words)	46.07%
Logistic Regression with bigrams	53.11%
Random Forest with bigrams	54.52%
Naïve Bayes with bigrams	52.91%
Gradient Boosting machines with bigrams	55.33%
Logistic Regression with trigrams	51.71%
Sentiment analysis	62.77%

Table 2: Model accuracy comparison

Conclusion:

The initial baseline accuracy was at 46%, able to improve the accuracy to a considerably high percent is achieved by applying Gradient boosting machines algorithm on bigrams. The accuracy is 55.33%.

Then, performing sentiment analysis on the same data by computing sentiments of different words, accuracy achieved is 62.7%, which is a good improvement considering a baseline of 46%. Hence, sentiment analysis with xgboost algorithm was the best performing model in our stock prediction using news headlines problem.

References:

1. Aaron7sun. Daily news for stock market prediction. Retrieved from Kaggle datasets: https://www.kaggle.com/aaron7sun/stocknews
2. Zygmunt Z. June 8th, 2015. Classifying text with bag-of-words: a tutorial. Retrieved from web article: http://fastml.com/classifying-text-with-bag-of-words-a-tutorial/
3. Pramod Chandrayan. Aug 26, 2015. Machine Learning part 3: Logistic Regression. Retrieved from web article: https://towardsdatascience.com/machine-learning-part-3-logistics-regression-9d890928680f
4. Cambridge University Press. 2008. Retrieved from web article: https://towardsdatascience.com/machine-learning-part-3-logistics-regression-9d890928680f
5. Front Neurorobot. 2013. Gradient boosting machines, a tutorial. Retrieved from web article: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/
6. Jason Brownlee. Sep 9th, 2016. A gentle introduction to Gradient boosting algorithm for machine learning. Retrieved from web article: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/
7. Miguel González-Fierro. Jan 31, 2017. A Gentle Introduction To Text Classification And Sentiment Analysis. Retrieved from publication: https://miguelgfierro.com/blog/2017/a-gentle-introduction-to-text-classification-and-sentiment-analysis/
8. Vivian Rajkumar. Jul 18th, 2017. Sentiment analysis for Yelp review classification. Retrieved from web article: https://medium.com/tensorist/classifying-yelp-reviews-using-nltk-and-scikit-learn-c58e71e962d9
9. Steven Loria. 2018. TextBlob: Simplified text processing. Retrieved from web: https://textblob.readthedocs.io/en/dev/