ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse108952020.pdf

https://doi.org/10.30534/ijatcse/2020/108952020

News Article to Uncover Media Bias

M.Ranjani[1], G.Divya[2], D.Saisandhiya[3]

[1]SRM Institute of Science and Technology, India, ranjanim1@srmist.edu.in
[2]SRM Institute of Science and Technology, India, divyag2@srmist.edu.in
[3]SRM Institute of Science and Technology, India, saisantd@srmist.edu.in

ABSTRACT

Media bias towards political issues is on the rise in the Indian subcontinent and South Asian nations. Media plays an extremely important role in the functioning of any democratic system. A bias media can lead to the formation of fallacious perceptions among the public. In recent years, there have been numerous instances where the ruling parties allegedly used media to manipulate people’s opinions in their favor. In this paper aims to study the news articles of major media houses, over the past few months and to find the degree of political bias. The data will be scraped from the published articles by various media houses and people's reactions to it from social media. The concept of clustering and topic modelling will be used to obtain the articles related to relevant topics from the data acquired. Finally, various classification techniques Naïve Bayes’ classifier, Bernoulli Bayes classifier, MNB classifier, stochastic gradient descent, NuSVC and LinearSVC will be used to determine the positive or negative sentiment for a particular topic. The result will draw a comparison between the media reaction and public sentiments towards a certain political party or issue. It will help us in determining the most trustworthy media houses in the country. It will also serve as a tool to put a check on the prejudiced news shown to the public and protect them from being manipulated.

Key words: Media bias, Naïve Bayes, MNB classifier, positive, negative

1. INTRODUCTION

Given the increasing number of these outlets, relying on a credible source of news seems to be a constant problem faced by the masses. Thus it becomes highly important for the common citizen to have access to credible sources to ensure that they are not manipulated by the extreme prejudices prevalent in the media. The news we read through digital sources is sought to be highly credible and trust in its senses but there are various filters through which each issue passes through before it comes out as an article. Text mining can play a very important role in avoiding this biasing. Sentiment Analysis from online news sources can be used to identify the topic of the article and determine the views being portrayed by the news outlet. Also, the recent boom of news aggregators also poses a similar kind of problem by putting forward articles from multiple sources and leaving it up to the audience to decide which source to believe in which may sometimes lead to reliance on the specious news outlets. To solve this issue in the best way possible, we developed a sentiment analysis.

It consists of a classifier that classifies text as positive or negative with the help of several algorithms namely Logistic Regression, Stochastic Gradient Descent and different implementations of the Naive Bayes Theorem. The paper consists of a User Interface that prompts the user to provide the keywords related to the news topic to be analyzed. On receiving the input news articles from major media houses are fetched using the Google news API. Similarly, tweets related to the topic are acquired using the twitter API. Now Sentiment analysis is performed separately on the data collected from both the sources. The analysis is done using different data sets for accurate results. The results are used to compare the reaction of the general masses represented by the twitter data to that of different media houses. Furthermore, a graphical representation is used to depict the results more effectively.

India is a diverse country with numerous factions of people coming from various back- grounds and cultures. With such a diverse mixture of population comes an equally clustered opinion for any given issue addressing the people of the country. These opinions may get offensive or hurtful to some factions of the country. The people in power have long been using and fixating upon these prejudices to fulfil their needs of power or acceptance. Considering this, it becomes a necessity for a credible source to portray the issues of the country in a neutral and a rightful and a rightful manner which may not create social unrest in the country.

Audiences are highly dependent on the current digital news platform boom and counting on it for daily updates on the various happenings across the world. It is very evident that not every new outlet shall be neutral in portraying the news and issues addressing the country. This may be due to their own political interests, financial backing by a political entity or maybe sheer fear of being dejected by the masses. This poses a great threat on the very fabric of democracy as people's right to information is falsified by malicious news. Our country has been recently in wake of numerous fake news or manipulated news via text messaging platforms and social media. Not everyone is equipped with the ability to filter out the false news or false agendas and derive the critical information from these articles. Thus we intend to eradicate this problem with our comprehensive system and help the audience get access to credible and most accurate news.

2. RELATED WORK

[1] One such endeavor was carried out by BRAC University, Dhaka. The main aim of the framework was to detect fake news circulated through tweets during the election campaign. The authors of the paper constructed a dataset of 200 tweets on 'Hillary Clinton'. Then normalization process was performed where pre-processing was done and articles were classified into categories with the help NER (Name Entity Recognition). The data of tweets was divided on the basis of their sources that were credible sources and malicious sites. Then with the help of NLTK library the words of importance were performed. So, 100 tweets were from credible dataset and the rest 100 were from malicious dataset.

On performing sentimental analysis on both the types of datasets it was found that credible tweets on 'Hilary' were positive and neutral while the malicious dataset showed negativity. Biasness could be observed since both the datasets results were totally opposite. K nearest Neighbors algorithm was used. The both datasets were assigned a polarity from -1 to +1. The sentimental analysis of random tweets of Hillary was -0.3. The value of K was 3, which means three nearest neighbors. With the help of Euclidean equation the three neighbor's distance was calculated and out of the three nearest neighbors 1 was labeled genuine and 2 were labelled fake. So the tweet was fake with an accuracy of 66.66%.

[2] Yet another research was conducted by fellow students of Dr. B. A. M. University. The main aim of

manner which may not create social unrest in the country

the framework was to determine sentiment of BBC news articles. In this research work 2225 documents of five different categories such as sports, technology, politics, entertainment and business from BBC news were chosen. The first stage performed was preprocessing, the main function was to clean the data and perform sentiment analysis on only relevant data. So, numbers, full stops, punctuation, white spaces and URL were removed with the help function 'removenumbers' in R studios. Then, all the data was converted to lowercase form and stemming was performed. In the final section technology out of 401 articles, 136 articles were positive and 24 articles were negative while the rest others were neutral

[3] In this research the author proposes to use lexicon based approach to find the sentimental analysis of the news articles. BBC news articles of five different categories such as sports, business, technology, politics and entertainment were used as data set. A total of 2225 articles of the year 2004 to 2005 were used. After the collection of articles, preprocessing was performed on the dataset to clean the data. Tokenizing of text was carried by the operator 'Tokenize'. Punctuation marks, stop words, white spaces were discarded and the entire text was converted to lower form. At the end of this stage, stemming was performed for better efficiency. After preprocessing, the polarity of the data was calculated using TFIDF, a mathematical method in which the words frequency was counted and weightage was given accordingly. In all categories of news articles it was observed that majorly their sentiments were either positive or neutral. In Technology and entertainment category articles most articles were negative while in business category most articles were positive. It was observed that in politics and sports category the positive and negative articles were equal in numbers.

[4] Another research was conducted by fellow students of University of Macau, China. The major mechanism used in the framework was MALLET (Machine Learning for Language Tool- kit) and other common methods such as TFIDF and chi square were not used because of less accuracy. Each algorithm's final score was calculated by taking the average of the 10 results calculated. The results showed Naive Bayes was the best algorithm as it was the only algorithm to classify all six articles correctly. Decision tree and Maximum entropy classifier were second best as four out of six articles were classified correctly. Both the winnow algorithms were not suitable for such specific tasks as their performance was the poorest as both of them had predicted random guess. The results were computed in the form of ROC curves (Receiver Operating

Characteristic). The greater the area between the random line and ROC curve the better the performance. Naïve Bayes has the greatest area while winnow has the least area.

[5] In this study by a University in Chennai, sentimental analysis was performed on the real time twitter dataset. A domain was selected and the live twitter data related to domain are stored in the database. The sentimental analysis was performed on the stored data. In this research paper, the topic 'IPL' was chosen and sentimental analysis was done using the tweets regarding IPL as twitter can also be considered as a medium for public's opinion. Several components of big data like apache flume, hive warehouse and hadoop distributed file system were used in the entire framework. Streaming API was used to download tweets which were then transformed to JSON and this transformed dataset was forwarded to flume sink. The main use of flume agent was to fetch live tweets which are real time and with the help of hadoop cluster these are stored in the database. Hive server was integral part of the hadoop cluster which was used in the operation. The sentimental analysis was performed on the stored data and three outcomes can be predicted which are positive, negative and neutral.

[6] This research by College of Engineering, Trivandrum performed Sentiment Analysis in Twitter using Machine Learning Techniques. The proposed solution was broadly divided into three parts, firstly a dataset was created using the tweets of a particular domain. In the framework the domain chosen was 'Electronic Products'. In the second stage, preprocessing was done on the dataset which includes removal of stop words and URL. Hashtag and emoji are used as relevant parts of the framework as emotions are expressed in it. So, weightage was given to such kind of occurrences. Emotions which were positive were given '+1' and the negative emotions were given '-1' weightage. These weightages were helpful in calculating the final result and improving the accuracy. Slang words were not avoided as they also contribute emotions. The last stage included performing sentimental classification by various classifiers. The results showed that if the pre-processing was done with highest quality then the results would be similar despite various classifiers.

[7] The research on sentiment analysis of Online News Articles was done by four different Universities of Bangladesh. In the framework, sentence level analysis played an important role in the polarity of the overall articles. With the help of a full stop sign, the sentence was detected. The sentences were classified into four categories which are simple sentence, complex sentence, compound sentence and compound complex sentence. If a simple sentence occurs, the polarity was calculated and added to the final result. For compound sentences, the sentences are divided into segments and then each segment's polarity was calculated from the polarity of words. The sentence polarity was calculated from all segment polarity and added to the final result. In case of complex sentences, it was divided into clauses which are dependent and independent. The polarity of both clauses are added to the final result. If the sentences are compound complex, the polarity was calculated by clauses separated by comma and added to the final result.

[8] This research by MSIT New Delhi, India mainly focuses on finding sentimental analysis of public review. The public reviews are gathered from various forums, e-commerce websites and portals. In this research paper, smartphone reviews were only considered as dataset and sentimental analysis was performed on it. Various preprocessing techniques were not performed in this research as dependencies of words were found and weightage was given accordingly and sentimental analysis was performed. So, Parsing was performed. The main goal of parsing was to find relation between words in the sentence. Then, sentences are tagged in a linguistic manner. Only relevant dependencies were taken into account while the rest others were ignored. Lexicon based approach and semi supervised based approach were used and SentiWordNet was used to calculate the score of the dataset. All the corresponding scores were added to get a final score. Multimap data structures were used.

[9] In the study conducted by fellow students of Sahyadri College of Engineering & Management, Mangalore. The authors dwell deeper into the aspects of Sentimental Analysis using Text Mining. In the proposed work, sentiments on different tweets were performed or the feedback which were published on the twitter were identified and then polarity was evaluated. The sentiments of the tweets were evaluated with respect to each word selected. For the testing and training of each word NBC was used for better efficiency. The classifier was also used to check the sentiment of each tweet. Different parameters based on performance were considered like time, precision plus accuracy and were compared with three classifiers based on machine learning which were SVM, Random Forest and NBC.

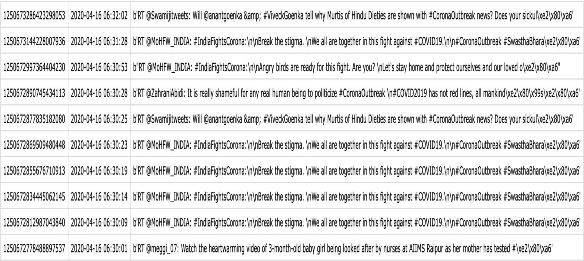
[10] This research by Giani Zeal Singh Campus College of Engineering and Technology, Bhatinda performed sentimental analysis using Hadoop and R language for the execution of different operations. Rhadoop tools were used as it could process large data (size of TB) in an efficient manner with the help of R language and R Hadoop Connector. The sentimental

analysis of small data was performed using the R algorithm. The tweets were extracted with the help of Twitter API. The dataset was then compared with loaded positive and negative words and sentimental analysis was performed on the dataset. Sentimental analysis was performed on 2000 tweets. The output could be positive, negative and neutral. For larger dataset Rhadoop was used. All the required files from the HDFS file system are fetched and functions are called. After the use of both the algorithms the scores are used for the comparison between R algorithm and Hadoop.

3. PROPOSED WORK

3.1 DATASETS

This area incorporates the dataset's depiction also as the preprocessing ventures before our investigation. We likewise expound on the usage of the citation extraction calculation we convey as Data gathering. Our analysis focus on gather the data which are tweets and the news articles relevant to a topic entered into the system. The tweets are obtained by the Twitter Api for Python using the tweepy library. The implementation is such that the user is asked to input the keyword or hashtag for searching the tweets and the news. The number of tweets that the system fetches has been set to a high value in order to get maximum tweets for optimum data analysis. The tweets are then stored onto a csv file in the local directory along with the date, a unique id for each tweet and the username.



1250673286423298053	2020-04-16 06:32:02	b'RT @Swamijitweets: Will @anantgoenka &amp; #ViveckGoenka tell why Murtis of Hindu Dieties are shown with #CoronaOutbreak news? Does your sickul\xe2\x80\xa6'
1250673144228007936	2020-04-16 06:31:28	b'RT @MoHFW_INDIA: #IndiaFightsCorona:\n\nBreak the stigma. \nWe all are together in this fight against #COVID19.\n\n#CoronaOutbreak #SwasthaBhara\xe2\x80\xa6'
1250672997364404230	2020-04-16 06:30:53	b"RT @MoHFW_INDIA: #IndiaFightsCorona:\n\nAngry birds are ready for this fight. Are you? \nLet's stay home and protect ourselves and our loved o\xe2\x80\xa6"
1250672890745434113	2020-04-16 06:30:28	b'RT @ZahraniAbidi: It is really shameful for any real human being to politicize #CoronaOutbreak \n#COVID2019 has not red lines, all mankind\xe2\x80\x99s\xe2\x80\xa6'
1250672877835182080	2020-04-16 06:30:25	b'RT @Swamijitweets: Will @anantgoenka &amp; #ViveckGoenka tell why Murtis of Hindu Dieties are shown with #CoronaOutbreak news? Does your sickul\xe2\x80\xa6'
1250672869509480448	2020-04-16 06:30:23	b'RT @MoHFW_INDIA: #IndiaFightsCorona:\n\nBreak the stigma. \nWe all are together in this fight against #COVID19.\n\n#CoronaOutbreak #SwasthaBhara\xe2\x80\xa6'
1250672855676710913	2020-04-16 06:30:19	b'RT @MoHFW_INDIA: #IndiaFightsCorona:\n\nBreak the stigma. \nWe all are together in this fight against #COVID19.\n\n#CoronaOutbreak #SwasthaBhara\xe2\x80\xa6'
1250672834445062145	2020-04-16 06:30:14	b'RT @MoHFW_INDIA: #IndiaFightsCorona:\n\nBreak the stigma. \nWe all are together in this fight against #COVID19.\n\n#CoronaOutbreak #SwasthaBhara\xe2\x80\xa6'
1250672812987043840	2020-04-16 06:30:09	b'RT @MoHFW_INDIA: #IndiaFightsCorona:\n\nBreak the stigma. \nWe all are together in this fight against #COVID19.\n\n#CoronaOutbreak #SwasthaBhara\xe2\x80\xa6'
1250672778488897537	2020-04-16 06:30:01	b'RT @meggi_07: Watch the heartwarming video of 3-month-old baby girl being looked after by nurses at AIIMS Raipur as her mother has tested #\xe2\x80\xa6'

Figure 1: CSV File with 10 tweets on the topic Corona Outbreak

Now using the csv library the driver code accesses the file and reads each tweet along with the relevant information that it carries such as the date, username of the person etc. This happens to be of no value in the analysis and hence through the gensim library and the nltk, the system extracts only the essential text pieces in the tweets. A tweet may comprise of images or html links which are now meaningless once stored locally. All this incoherent data is eradicated through stemming, lemmatization and removal of stop words.

Example of one such operation is as follows:

1225475879171653632

2020-02-06 17:46:32

b"RT @ANI: BJP MP Tejasvi Surya in Lok Sabha yesterday: What is happening today in Delhi's Shaheen Bagh is a stark reminder that if the major\xe2\x80\xa6".

This is one row Fig 1) in the csv file containing the tweet, its ID, date and the username. This particular tweet also has some pictures attached to it which are converted into the relevant format. However the pictures do not provide insights in the data hence they should not be accounted for.

The end result of the preprocessing of this tweet is as follow:

[‘happen’, 'today', 'delhi', 'shaheen', 'bagh', 'stark', 'remind', ‘major']

As evident, a substantial amount of incoherence in the data has been removed and what's left is the data worth performing analysis on. Similarly, the system also fetches the news articles on the input topic through the news API by Google. The API fetches the various links of articles on the input topic in a JSON file. Next, where the data gathered shall be subjected to different algorithms using the scikit library. This module of the system is to determine the polarity of the text in concern is dependent on various classifiers. The polarity of the text is going to be either positive (in support of the topic) or negative (against the topic). The reason behind choosing several classifiers is the varying accuracy of each classifier at every instance of analysis.

Thus it would lead to inefficient results if only one classifier is singled out. Following are the classifiers that have been put into use in the system. Naive Bayes' Classifier, MNB Classifier, Bernoulli NB Classifier, SGD Classifier, Logistic Regression Classifier, NuSVC Classifier, LinearSVC Classifier

The training of the classifiers has been done using two sets of training data gathered from the internet. One set contains a large corpus of movie reviews divided into positive and negative reviews. Twitter users often tend to use a satirical or an informal manner of vocabulary to voice their concerns which is also the case with movie reviews. Hence using this corpus for testing twitter tweets would yield the most accurate results.

The second data set contains a large corpus of various news articles. This was obtained from yet another online source wherein the author classified several links to online news articles as positive or negative. The articles in those links were further extracted using the Newspaper API by Google. Thus the two data sets can be used accordingly to train and test different kinds of Texts. The movie reviews data shall be relevant for Twitter data and the news article data would be for the news articles we intend to analyses.

These two data sets were then used to generate features of the text.

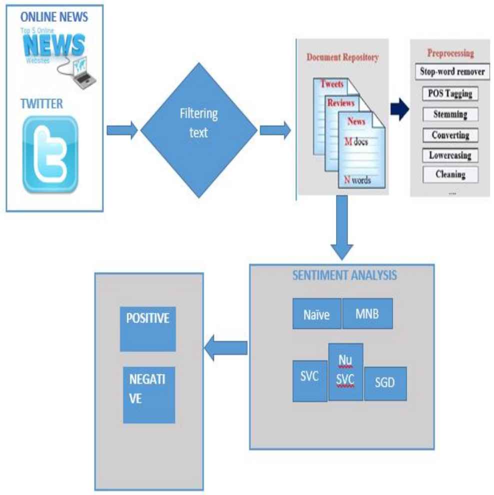


Figure 2: Architecture diagram

The data set containing the movie reviews was stored in two different file namely positive.txt and negative.txt. Both these files were accessed using the file reading techniques in python to extract out all the words in the files. All the words will be stored in a list of tuple with their polarity. The words from files will be read and placed in a tuple with ‘pos’ or ‘neg’ accordingly. All these tuples will be stored in a list to create a list of all words from both positive and negative reviews. Now, out of all these words, only adjectives and verbs are further taken because of their importance in text polarity.

Tokenization is the first step in Natural Language Processing. It breaks a sentence into words. The part-of-speech of each word is then obtained by pos_tagging in Python NLP. This allows the words to be tagged in the form of lists of two elements. The first being the word itself and the second being the tag. Now tags would be used to allow only verbs and adjectives to be considered further and stored into another list. Further, out of all these adjectives and verbs, only the most occurring ones are taken forward for defining the features of the text. This is done by using the Frequency Distribution function of the NLTK to get the most occurring words and then taking the first 7000 words. These pos_tagged words are then checked to see if they exist in the all files of both the positive and negative files and accordingly giving those Boolean results i.e. True or False. This generates a list of features with corresponding Boolean values checking their presence in both the text files. These features will then be used to train and test the classifiers for accuracy. The classifiers that are trained and tested against these features as follows.

3.2 Naive Bayes’ Classifier

The process is accomplished by measuring each label's prior probability by evaluating the frequency of each label within the training set. The frequency contribution of the overall event by every single feature combined with the initial probability gives the final probability for every label. The level with greatest probability is the final result given by the Classifier.

The formula for the classifier in mathematical terms would be:

$P(E|y) = P(y|E)*P(E)$

$P(y)$ Here, E is an event and y is the assumption.

Thus P (E|y) denotes the probability of an event E when the assumption y is true which is obtained by dividing the product of prior probability of the event given the assumption y and the probability of the event E by the probability of the assumption y classifier would come into play in text analysis because the system generates a list of features from the given text. These features would be used to predict whether the text is positive or negative by considering the possibility of each feature having equal importance in determining whether the text is positive or negative. In this case, analogous to the Event E in above example would be the text being Positive/Negative and the assumption x would be each feature of the text.

3.3 MNB Classifier

The Multinomial Naive Bayes (MNB) algorithm implements the original NB algorithm for data that is distributed in a multinomial manner. This classifier differs from the original NB in the sense that it takes into account the weightage of each word that occurs in the text by using Word Vector Counts. Unlike the original NB which does not take into consideration the importance of words as features, the MNB does use tf-idf to estimate the probability of features belonging to a class.

The distribution is parameterized by vectors Py = (Py1… Pyn), where y= class and n= no. of features of the text corpus and Pyi is P (i | y), the likelihood of the ‘i’ occurring in the class y.The parameter Py is approximated by a smoothed version of maximal likelihood by incorporate a constant α>=0 which is used to account for features absent from the learning samples.

$\hat{}Pyi = \quad Myi + \alpha$

$My + \alpha m$

Where Myi = no. of times feature ‘i’ appears in a sample of class y

My = total count of all features for class y

Assigning the smoothing constant α=1 is known as Laplace Smoothing and any value smaller than one is known as Lidstone Smoothing. An instance of using this classifier would be while classifying a text into a wider category. Consider a news aggregator system which has to display news based on the various categories such as Sports, Finance, Defence, and Crime etc. Thus, given a text the program analyses the importance of each word (or feature) in the text and determines whether or not the text belongs to a category.

3.4 Bernoulli NB Classifier

This implements the original NB classification algorithm for the distributed data according to a multivariate Bernoulli distribution. It is assumed that multiple features possess a binary value. Thus, samples have to be represented as feature vectors with binary value. If given any other kind of data, the classifier will try to binaries the value according to a binary parameter before proceeding.

The Bernoulli NB decision rule is determined on the following basis:

P (wi | z) =P (i | z) wi + (1-P (i | z)) (1-wi)

The differing factor from the Multinomial NB is the fact that it penalizes the absence of a feature i, an indicator for class z, whereas the multinomial variant simply overlooks an absent feature. During our study it was found that the Bernoulli NB might out-perform the other classifiers on some datasets. However it was best to employ a system which takes into consideration, all the classifiers and opts for the best one on its own.

3.5 Stochastic Gradient Descent

Different variations of Gradient descent algorithms are used for optimization of various data in the form of clusters, squiggle, line etc. Gradient means the slope of a surface, and gradient descent algorithms are all about finding the minima of a surface. Gradient descent is a repute algorithm. It starts at an arbitrary position on the function curve and traverses down its gradient in a stepwise manner. The iteration continues till the minima of the curve is reached. The gradient descent however fails when the data size is too large. Stochastic Gradient Descent helps us to overcome this problem. SGD selects one random data point after each iteration which immensely reduces the calculations. It is also very efficient when working with data which have redundancies, as it selects only one point from a cluster or ‘mini-batch’ of data which significantly brings down the number of data points.

3.5.1 Logistic regression

The following procedure is followed to calculate the likelihood.

1. The original data points are projected onto a candidate line.
2. The y coordinate of the projections gives the values of log (odds). Then we transform the candidate log(odds) into candidate probabilities using the formula:P = elog (odds) / 1+ elog(odds)
3. We plot a squiggle using the obtained probabilities for each point as y coordinates.
4. In this case, the likelihood for all the data points is the same as the probability i.e. the value corresponding to the y-axis. The likelihood of the squiggle can be computed by finding the product of all the individual likelihoods.
5. The process is repeated with different candidate lines, and the squiggle with the maximum likelihood is selected.

3.6 NuSVC and LinearSVC

In scikit learn NuSVC and LinearSVC are implemented under the SVM library. State Vector Machine (SVM) is an algorithm very useful in classification. In simple words it divides different classes using a hyperplane drawn in a multidimensional space. The SVM algorithm finds an optimum threshold value for the construction of hyperplane, in order to reduce chances of error. The distance from the hyperplane to the nearest elements of the classes is known as margin. The data points more closely to the hyperplane are known as support vectors and are more relevant in constructing the classifier.

3.6.1 SVM Kernels:

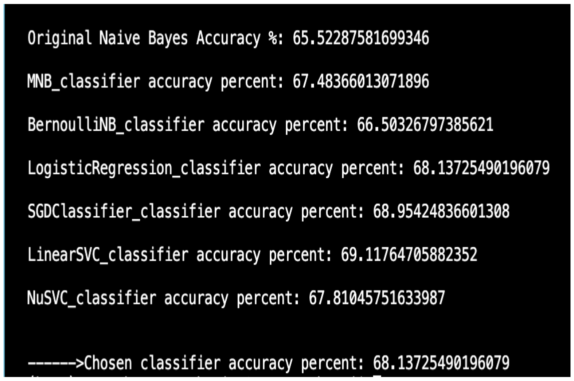
In certain problems the data points are inseparable and cannot be classified using a two dimensional hyperplanes. For such problems the SVM algorithm uses a kernel trick to convert the data into higher dimension space so that it can be classified using a hyperplane. Linear SVC is uses linear kernel and possess greater flexibility in regard to loss functions and penalties. It is especially helpful in scaling large numbers of samples. The implementation is based on lib linear. NuSVC is based on libsvm implementation. It makes use of parameters in order to control support vectors. The kernel parameter is set to RBF by default, but can be changed if required. Now choosing the best out of all these classifiers is going to be through a voter class in Python.

An object of the class takes all these classifiers as arguments. The classifiers are then put into a list by the constructor of the voter class. Then, by iterating through all the classifiers in the list one by one, the system analyses the text. The output of the analysis is going to be either ‘pos’ or ‘neg’. This is because, each every classifier has been trained with a feature set and the set contains nothing but key-value pairs of features of text and the polarity they belong to. The polarity during training was 'pos' or 'neg' and hence the testing shall also pro- duce the same result.

4. EXPERIMENT AND RESULT

Now after every classifier has been used to analyses the text given, the results are stored in a list. The final result of the analysis shall be the mode of all the results. That is, checking how the majority of classifiers classify the text. If there are more number of 'pos' results in the list than 'neg' then the text is noted to be in Positive polarity and vice versa. The class also possesses the functionality responsible for measuring the result's confidence%. This is calculated by taking into account the mode count of the results dividing it by 7 (no. of classifiers). This is then multiplied by 100 to get a percentage.

It is clear that more number of classifiers find the text to be positive. Hence the final outcome shall be



```
Original Naive Bayes Accuracy %: 65.52287581699346

MNB_classifier accuracy percent: 67.48366013071896

BernoulliNB_classifier accuracy percent: 66.50326797385621

LogisticRegression_classifier accuracy percent: 68.13725490196079

SGDClassifier_classifier accuracy percent: 68.95424836601308

LinearSVC_classifier accuracy percent: 69.11764705882352

NuSVC_classifier accuracy percent: 67.81045751633987


------>Chosen classifier accuracy percent: 68.13725490196079
```

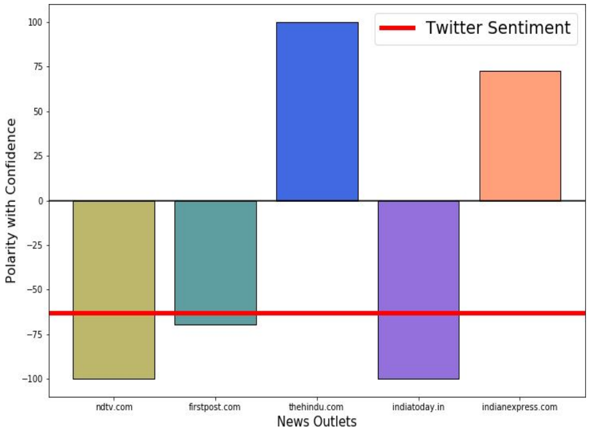
Figure. 3: The output of driver code of analysis module. These percentages are subject to change, which is why the chosen classifier shows an accuracy which is not the highest of the above accuracies because the numbers altered when the second time run.

'pos'. For confidence, the number of classifiers that voted as positive is 4.

Thus the confidence shall be:

C0=4*100 7 C0=57.14%

Hence, the tweet shall be declared as POSITIVE with a confidence of 57.14%

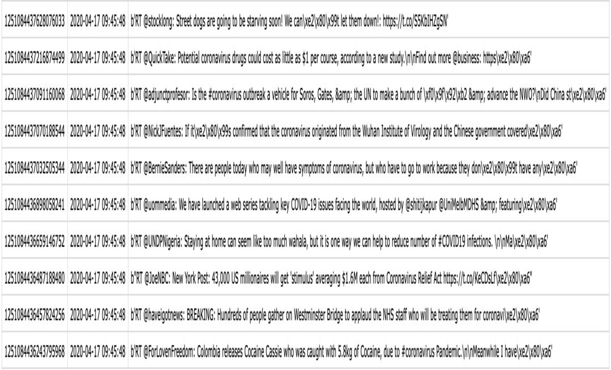


This confidence acts as a utility in determining the cumulative sentiment of all tweets or all articles. The confidence is taken as a weightage factor in determining the cumulative. The final result of analysis of the tweet shall be a tuple ('pos', 57.14) with the sentiment and its confidence. Similar results of all tweets will then be used in assigning a cumulative for Twitter Sentiment by taking the sum of confidence of mode of the results of all tweets. If the confidence of negatively rated results is greater than the twitter sentiment on the topic will be chosen as Negative and vice versa.

The confidence percentage of this result will be calculated by:

C=Sum of confidence of all chosen results*100 Total confidence of all results

For better understanding take the following example, considering the analysis of 10 tweets on Coronavirus



1251084437628076033	2020-04-17 09:45:48	b'RT @stocklong: Street dogs are going to be starving soon! We can\xe2\x80\x99t let them down!: https://t.co/S5KbIHZgSN'
1251084437216874499	2020-04-17 09:45:48	b'RT @QuickTake: Potential coronavirus drugs could cost as little as $1 per course, according to a new study.\n\nFind out more @business: https\xe2\x80\xa6'
1251084437091160068	2020-04-17 09:45:48	b'RT @adjunctprofesor: Is the #coronavirus outbreak a vehicle for Soros, Gates, &amp; the UN to make a bunch of \xf0\x9f\x92\xb2 &amp; advance the NWO?\nDid China st\xe2\x80\xa6'
1251084437070188544	2020-04-17 09:45:48	b'RT @NickJFuentes: If it\xe2\x80\x99s confirmed that the coronavirus originated from the Wuhan Institute of Virology and the Chinese government covered\xe2\x80\xa6'
1251084437032505344	2020-04-17 09:45:48	b'RT @BernieSanders: There are people today who may well have symptoms of coronavirus, but who have to go to work because they don\xe2\x80\x99t have any\xe2\x80\xa6'
1251084436898058241	2020-04-17 09:45:48	b'RT @uommedia: We have launched a web series tackling key COVID-19 issues facing the world, hosted by @shitijkapur @UniMelbMDHS &amp; featuring\xe2\x80\xa6'
1251084436659146752	2020-04-17 09:45:48	b'RT @UNDPNigeria: Staying at home can seem like too much wahala, but it is one way we can help to reduce number of #COVID19 infections. \n\nMa\xe2\x80\xa6'
1251084436487188480	2020-04-17 09:45:48	b"RT @JoeNBC: New York Post: 43,000 US millionaires will get 'stimulus' averaging $1.6M each from Coronavirus Relief Act https://t.co/KeCDsLf\xe2\x80\xa6"
1251084436457824256	2020-04-17 09:45:48	b'RT @haveigotnews: BREAKING: Hundreds of people gather on Westminster Bridge to applaud the NHS staff who will be treating them for coronavi\xe2\x80\xa6'
1251084436243795968	2020-04-17 09:45:48	b'RT @ForLovenFreedom: Colombia releases Cocaine Cassie who was caught with 5.8kg of Cocaine, due to #coronavirus Pandemic.\n\nMeanwhile I have\xe2\x80\xa6'

Figure 4: Tweets on the topic Coronavirus Stored in a local CSV file

Now these tweets are retrieved, extracted and cleansed before proceeding to analyse using the above algorithm. The CSV file accesses each tweet and extracts the relevant information to finally give us the text portion in each tweet. The text is then tokenized and made devoid of any unimportant elements such as image formats, emoticons or hashtags. The pre-processed tweets are now ready to be analysed and the results of the analysis of each tweet is noted.

Here it is evident that the confidence of positive results is less than negative results, thus the overall sentiment of Twitter can be considered to be Negative.

For overall confidence

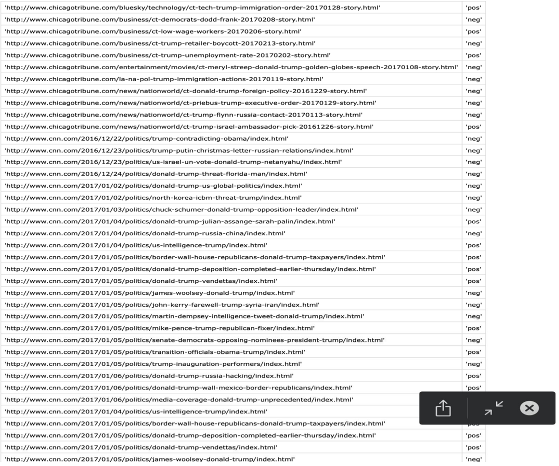
C=Sum of confidence of all negatively rated results*100 Total confidence of all results

A similar approach is also utilized in analyzing News Articles Text. The dataset for training is stored as a CSV file in the local directory. This file contains web links to various news articles and their ratings as positive or negative as shown in Fig 5.

Figure 5: Graph showing comparison of all news outlets in contrast with the public opinion

These links are then accessed to retrieve the article text using the Newspaper library in Python and then stored in two different text files namely positive_news.txt and negative_news.txt accordingly. The newspaper library uses the link and extracts the information about the article. Doing this, the text in the article can be extracted and stored into one of the two txt files. After all the links have been accessed and article texts have been stored locally, they can be used as the data for training and testing.

Now the process for training and testing the data will be through the same procedures and classifiers as they were for Twitter data. The sentiment of news articles will be used to check how reputed media houses are reacting to the topic in concern.



'http://www.chicagotribune.com/bluesky/technology/ct-tech-trump-immigration-order-20170128-story.html'	'pos'
'http://www.chicagotribune.com/business/ct-democrats-dodd-frank-20170208-story.html'	'neg'
'http://www.chicagotribune.com/business/ct-low-wage-workers-20170206-story.html'	'pos'
'http://www.chicagotribune.com/business/ct-trump-retailer-boycott-20170213-story.html'	'neg'
'http://www.chicagotribune.com/business/ct-trump-unemployment-rate-20170202-story.html'	'pos'
'http://www.chicagotribune.com/entertainment/movies/ct-meryl-streep-donald-trump-golden-globes-speech-20170108-story.html'	'neg'
'http://www.chicagotribune.com/la-na-pol-trump-immigration-actions-20170119-story.html'	'neg'
'http://www.chicagotribune.com/news/nationworld/ct-donald-trump-foreign-policy-20161229-story.html'	'neg'
'http://www.chicagotribune.com/news/nationworld/ct-priebus-trump-executive-order-20170129-story.html'	'neg'
'http://www.chicagotribune.com/news/nationworld/ct-trump-flynn-russia-contact-20170113-story.html'	'neg'
'http://www.chicagotribune.com/news/nationworld/ct-trump-israel-ambassador-pick-20161226-story.html'	'pos'
'http://www.cnn.com/2016/12/22/politics/trump-contradicting-obama/index.html'	'neg'
'http://www.cnn.com/2016/12/23/politics/trump-putin-christmas-letter-russian-relations/index.html'	'neg'
'http://www.cnn.com/2016/12/23/politics/us-israel-un-vote-donald-trump-netanyahu/index.html'	'neg'
'http://www.cnn.com/2016/12/24/politics/donald-trump-threat-florida-man/index.html'	'neg'
'http://www.cnn.com/2017/01/02/politics/donald-trump-us-global-politics/index.html'	'neg'
'http://www.cnn.com/2017/01/02/politics/north-korea-icbm-threat-trump/index.html'	'neg'
'http://www.cnn.com/2017/01/03/politics/chuck-schumer-donald-trump-opposition-leader/index.html'	'neg'
'http://www.cnn.com/2017/01/04/politics/donald-trump-julian-assange-sarah-palin/index.html'	'pos'
'http://www.cnn.com/2017/01/04/politics/donald-trump-russia-china/index.html'	'neg'
'http://www.cnn.com/2017/01/04/politics/us-intelligence-trump/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/border-wall-house-republicans-donald-trump-taxpayers/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/donald-trump-deposition-completed-earlier-thursday/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/donald-trump-vendettas/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/james-woolsey-donald-trump/index.html'	'neg'
'http://www.cnn.com/2017/01/05/politics/john-kerry-farewell-trump-syria-iran/index.html'	'neg'
'http://www.cnn.com/2017/01/05/politics/martin-dempsey-intelligence-tweet-donald-trump/index.html'	'neg'
'http://www.cnn.com/2017/01/05/politics/mike-pence-trump-republican-fixer/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/senate-democrats-opposing-nominees-president-trump/index.html'	'neg'
'http://www.cnn.com/2017/01/05/politics/transition-officials-obama-trump/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/trump-inauguration-performers/index.html'	'neg'
'http://www.cnn.com/2017/01/06/politics/donald-trump-russia-hacking/index.html'	'pos'
'http://www.cnn.com/2017/01/06/politics/donald-trump-wall-mexico-border-republicans/index.html'	'pos'
'http://www.cnn.com/2017/01/06/politics/media-coverage-donald-trump-unprecedented/index.html'	[illegible]
'http://www.cnn.com/2017/01/04/politics/us-intelligence-trump/index.html'	[illegible]
'http://www.cnn.com/2017/01/05/politics/border-wall-house-republicans-donald-trump-taxpayers/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/donald-trump-deposition-completed-earlier-thursday/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/donald-trump-vendettas/index.html'	'pos'
'http://www.cnn.com/2017/01/05/politics/james-woolsey-donald-trump/index.html'	'neg'

Figure 6: Data set consisting web links to articles

This is done by accessing the web links to news articles posted by a few media houses selected which are present in the code. We chose NDTV, India Today, First Post, The Hindu and Indian Express as the media houses to be compared. The system uses the News API by Google to access web links of articles by each of these news outlets.

The News API returns a JSON object which is then used to extract out the link of the article. Now that the link is obtained, the newspaper library successfully extracts the text as well. This text is then subjected to the same algorithm to assign a sentiment and corresponding confidence percentage. This is done for all articles by each of the news outlets. The cumulative for each news outlet shall be the cumulative of sentiment for each article by the outlet. After doing this procedure for each of the five media houses, a comparison can be drawn to denote which houses are in motion with the public sentiment and which are not. We use the sentiment of the twitter data as the public sentiment on the topic of concern and set it as the bar for comparison. Any news outlet which opposes the public sentiment can be considered as trying to cloud the opinion of the public through their articles.

Both of these modules and their intricacies are also masked by a comprehensive UI which allows the user to have a single access point to enter the topic of concern which invokes all the necessary functions to retrieve twitter data and news articles and perform the analysis to show the results in a comprehensive graph format as depicted above. The UI has been developed using the Python Tkinter module which provides all the necessary widgets and containers for them.

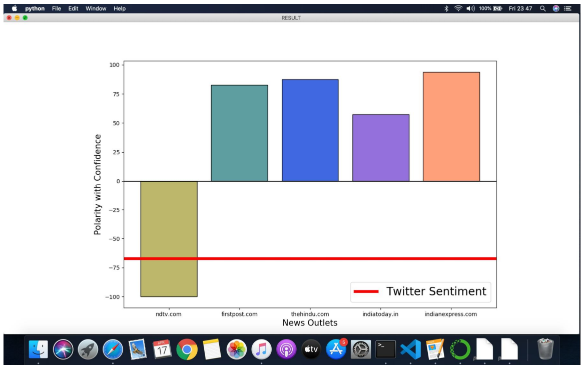


Figure 7: The final graph shown after clicking the Display Result button on the home screen

5.CONCLUSION

This paper built using the latest tools and techniques and within the given timeframe brings about a lot conclusions with it. It is not overstating to say that the Media is one of the most powerful entities after the government itself and it naturally becomes a necessity to put a check on that power. We intended to build a simple yet highly comprehensive system for this purpose. The UI built for the system has been tried to be simplistic. As explained earlier, the system has differing data sets and a wide range of classifiers to choose from for the analysis. The news articles are also extracted using a robust tool provided by Google and so are the tweets using the API by Twitter. All these tools and the algorithms used have helped us create the best possible system within schedule. The processing of the text ensures that there is no dirty data which may lead to fallacy. Each of the seven classifiers are tested at every instance of execution and only the best one is chosen for further analysis. This not only gives a higher accuracy but also reduces risk of flaws in the final result. The final result is portrayed as a bar chart and allows the user to determine how the news outlets are faring against the public opinion. The study during this project also helped us gain vital knowledge in the field of Machine Learning and the plethora of its application to solve real world problems.

6.FUTURE ENHANCEMENT

Given the increasing number of news providers in the world it gets devious for the common man to be sure whether the news he reads is presented in the right form or not. Governments also play a major role in this scenario by shunning a particular media house or promoting the other in the form of funding, ratings or firsthand information. These cases can be avoided by employing this system to make sure that what the public perceives as news is in turn matching with their philosophy too. If a particular media house intends to cloud the public opinion by their articles for the purpose of supporting a particular cause or for funding from a source then this system shall be able detect the difference in opinion and leave it at the choice of the public to rely on the information or not. The algorithm currently takes about 2 full minutes to go from start to finish. With further efforts on the system, the algorithmic efficiency can be scaled up to yield better results within less processing time and space. With such a system being open source, users can harness the utility and also contribute in making the system faster and better.

REFERENCES

1. Amitabha Dey, Rafsan Zani Rafi, Shahriar Hasan Parash, Sauvik Kundu Arko and Amitabha Chakrabarty, **“Fake News Pattern Recognition using Linguistic Analysis”,** 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV)
2. Vishal S. Shirsat. Rajkumar S. Jagdale and S. N. Deshmukh, **“Document Level Sentiment Analysis from News Articles”,**2017 International conference on computing, communication, Control and Automation(ICCUBEA)
3. Soonh Taj, Baby Bakhtawer Shaikh, Areej Fatemah Meghji, **“Sentiment Analysis of News Articles: A Lexicon based Approach”,** 2019 International Conference on Computing, Mathematics and Engineering Technologies.
4. Simon Fong, Yan Zhuang, Jinyan Li, Richard Khoury **“Sentiment Analysis of Online News using MALLET”,** 2013 International Symposium on Computational and Business Intelligence
5. G.Kavitha, B.Saveen, Nomaan Imtiaz **“Discovering Public Opinions by Performing Sentimental Analysis on Real Time Twitter Data”** 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)
6. Neethu M S, Rajasree R, **“Sentiment Analysis in Twitter using Machine Learning Techniques”,** 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)
7. Muhammad Usama Islam, Faisal Bin, Ali Imam Abir, M.A. Mottalib**, “Polarity Detection of Online News Articles Based On Sentence Structure and Dynamic Dictionary”,** 2017 20th International Conference of Computer and Information Technology (ICCIT)
8. Pooja Kherwa, Prashast Kumar Singh, Arjit Sachdeva, Dhruv Mahajan, Nishtha Pande**, “An approach towards comprehensive sentimental data analysis and opinion mining”,** 2014 IEEE International Advance Computing Conference (IACC)
9. Shamantha Rai B, Sweekriti M Shetty **“Sentiment Analysis Using Machine Learning Classifiers: Evaluation of Performance”,** 2019 IEEE 4th International Conference on Computer and Communication Systems
10. Sunny Kumar, Paramjeet Singh, Shaveta Rani **“Sentimental Analysis of Social Media Using R Language and Hadoop: Rhadoop”,** 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
11. M.Trupthi1, Suresh Pabboju, G.Narasimha, **“Sentiment Analysis on twitter using streaming API”,** 2017 IEEE 7th International Advance Computing Conference (IACC)
12. Lipika Dey, Anuj Mahajan and SK. Mirajul Haque, **“Document Clustering for Event Identification and Trend Analysis in Market News”,** 2009 Seventh International Conference on Advances in Pattern Recognition.