International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Rating Prediction Based on Textual Review: Machine Learning Approach, Lexicon Approach and the Combined Approach

Rukhsar Haji[1], Daanyaal Kapadia[2], Deval Ghevariya[3], Rushikesh Gajmal[4]

[1]*Assistant Professor, Department of Computer Engineering, Rizvi College of Engineering, Mumbai, India*
[2, 3, 4] *Student, Department of Computer Engineering, Rizvi College of Engineering, Mumbai, India*

Abstract -*Reviews and Ratings of any restaurant are must to know before going to visit it. This paper presents Rating Prediction based on user Review. A step-by-step methodology for User-based review sentiment analysis. The first, machine learning approach, tackles the problem as a text classification task employing supervised classifier like Naive Bayes algorithm as it is most suited for text classification. The Second, a lexicon-based method, uses a dictionary of words with assigned scores to the text-based review to calculate a polarity of reviews and decide if the review is positive or negative. Here in this paper, we show the combination of lexicon and machine learning approaches which improves the accuracy of Naïve Bayes classification by 5% to 10% based on review length and the size of the dataset.*

Key Words: Tokenisation, Stopwords, Stemming, Corpus, Bag of words, TF-IDF, Naive Bayes, Lexicon, VADER, reviews, rating.

1. INTRODUCTION

The rise in online shopping has brought a significant rise in the importance of textual customer reviews. There are thousands of review sites online and massive amounts of reviews for each and every product. Nowadays customers have changed their way of shopping. 60-70 percent of customers say that they use rating filters to filter out low rated items in their searches.

The ability to successfully decide whether a review will be helpful to other customers and thus give the product more exposure is important to companies that support these reviews, companies like Flipkart, Amazon and Yelp!.

The customer will make a decision to buy a product if he or she sees valuable reviews posted by others, especially the user's trusted friend. We believe reviews and reviewers will do help to the rating prediction based on the idea that 5-star ratings may greatly be attached with extremely good reviews. It's also agreed that different people may have different sentimental expression preferences. For example, some users prefer to use "good" to describe an "excellent" product, while others may prefer to use "good" to describe a "just so so" product[4] [5]. User's rating information is not always available on many review websites.

Here we will predict the rating of the textual reviews based on two different approaches, the Machine Learning approach, and the Lexicon approach. This problem can also be referred to as a text classification problem if solved by machine learning approach. The Machine Learning approach makes use of the Naïve Bayes algorithm to classify the reviews into classes or rating. If the machine learning approach is applied data pre-processing should to done in order to make the data suitable for the classifier to understand. The Lexicon approach can't classify the review into five different class but it can return the sentiment of the document whether it is positive or negative. With the help of Machine Learning approach and Lexicon approach, a new combined approach is created which will increase the existing accuracy of the Naïve Bayes algorithm by approximately 5 percent. The accuracy of prediction of the rating is low because it depends on the user to user, for example, the same review can get 4 star from a user and can also get 5 star from another user also.

2. Machine Learning Approach

In Machine Learning based approach we will cover supervised machine learning techniques used for text classification. Supervised machine learning requires a labelled training dataset on which the classifier will be trained [3]. Each example in the training dataset consists of an input attribute and a class. After training the classifier, the classifier can predict the class of the data. In our case, there is a class with 5 values associated with it. The supervised algorithm analyses labeled training data, extracts features that model's the differences between different classes and infers a function, which can be used for classifying new reviews entered by the user or from test dataset. The steps for building the Machine learning model using a supervised machine learning algorithm is as follows:

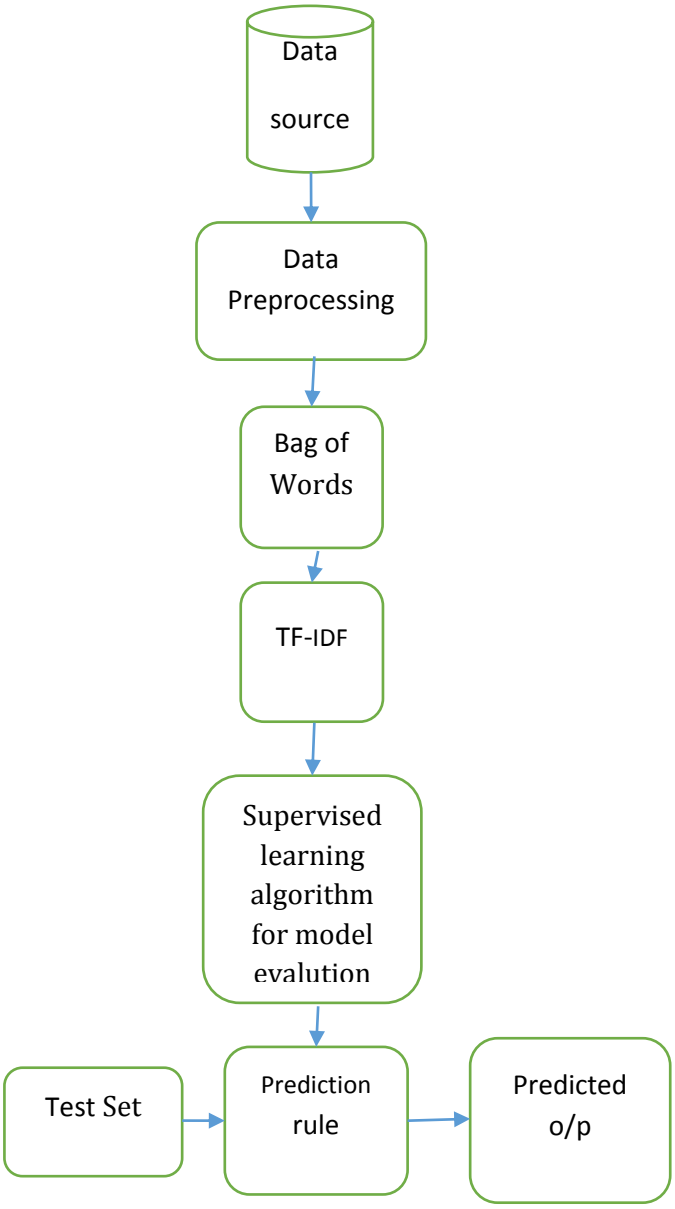


Fig 1. Work flow of ML Approach

2.1 Data Pre-processing

Data Pre-processing is a must needed step in order to build a classification model. In this step, the raw review will go through different pre-processing steps in order to get the data which is required for the classifier. The pre-processing steps are as follows:

2.1.1 Removal of unwanted characters and punctuation

In this step, all the unwanted character which are not required by the classifier or which do not contribute in making a review positive or negative will be removed and the only alphabet will be left over, which will be in both upper case as well as of lower case.

Example: This is not a good product!

O/P: This is not a good product

Here in the above example the '! ' will be removed.

2.1.2 Text Case Conversion

All the letters in the textual review will be converted in a single case that is the lower case which will help the classifier to gain more accuracy. For Example, the word 'Review' and 'review' will be considered as two different words. If 'Review' is converted to 'review' then both will be considered as same words. So that is the reason for the case conversion in the textual review**.**

2.1.3 Removal of Stop words

Stop words are those words which do not contribute to making a particular sentence positive or negative. Example words like 'is', 'a', 'the', etc. are considered as stop words. Stop words can be in positive as well as in negative reviews and it does not affect the polarity of the review.

Example: It is a good product

O/P: good product

2.1.4 Tokenization

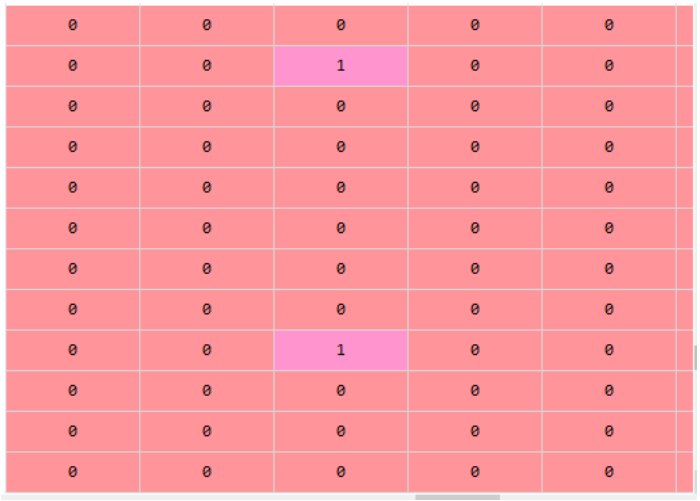
Tokenization is taking a sentence into consideration and breaking it up into its individual words or tokens and tokens are mostly a single word. The use of tokenization is in Bag of Words (BoW) model in which every column will represent different words.
Example: good product
O/P: ['good', 'product']
If there are five words excluding stopwords then five different tokens created for five different words.

2.1.5 Stemming

Stemming is the process of reducing a word to its word stem that affixes to the roots of words known as a lemma. Stemming is important in natural language processing (NLP) and text categorization.
Example: loving
O/P: love
After all these data pre-processing steps the pre-processed data is stored in a list which is called a corpus. Corpus means a collection of related written text.

2.2 Bag of Words (BoW)

A bag-of-words (BoW) model also known as binary BoW model is a way of extracting features from the text for use in training the machine learning model. The approach is very flexible and simple and can be used in a countless number of ways for extracting features from a text. It is called as '*BAG*' of words because any information or knowledge about the structure and order of words in the text is discarded. It is only concerned with whether known words occur in the sentence and how many time, not where in the sentence. The intuition is that text is similar if they have similar content. Further, from the content alone we can learn something about the meaning of the text.



0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Fig 2. BoW model example

2.2.1 Problems associated to Bag of Words model

BoW also called a binary BoW goes not give the importance of a word in a document. All the words have the same important. We can't distinguish which word is more important than other words. In the sentence like:
'You are an awesome guy.'
If here we replace the word awesome the complete meaning of the sentence will change. So here awesome have great meaning to it. But BoW goes not give any importance to it. Also, no semantic information preserved. For improving the BoW model we have other model called a TF-IDF model.

2.3 TF-IDF

In TF-IDF some semantic information is preserved as uncommon words are given more importance than common words
Here the word 'awesome' in the sentence 'You are an awesome guy' will get more important.
TF-IDF model give more importance to specific, uncommon and important words
Now considering 3 sentences.
Sentence 1 = 'This will be interesting' ....(i)
Sentence 2 = 'This movie is interesting' ....(ii)
Sentence 3 = 'This movie is bad' ....(iii)
For Creating a TF-IDF model, first, we have to create a BoW model. For constructing standard BoW model we have to follow the standard procedure. Firstly we should preprocess the data.

1. Remove the unwanted character
2. Conversion to lower case
3. Removal of stopwords
4. Tokenization
5. Stemming

This concludes the data preprocessing. After preprocessing the data BoW model can be created.
Creating a TF-IDF model:
TF is Term frequency. The term frequency of a particular word in a particular document can be calculated by a formula which is given as:

$$\frac{(Number\ of\ occurrences\ of\ a\ word\ in\ a\ document)}{(Number\ of\ words\ in\ that\ document)}$$

Fig 3.TF formula

With this formula, we can calculate the TF of all the words in all the documents.
IDF- Inverse Document frequency
The IDF value of a word is common in a whole corpus of the document. There will be only one IDF value for a given word in the whole corpus of the document. The IDF value can be calculated by the formula which is given as:

$$log\left(\frac{(Number\ of\ documents)}{(Number\ of\ documents\ containing\ word)}\right)$$

Fig 4.IDF formula

Now for getting the TF-IDF value, we need to multiply the TF and the IDF value for the given word.

$$TFIDF(Word) = TF(Document, Word) * IDF(Word)$$

Fig 5.IF-IDF formula

With the help of this TF-IDF model now we can find the importance of the word in the given document. The TF-IDF matrix of sentences (i), (ii) & (iii) is given below.

	0	1	2
0	0	1	0
1	0	0.707107	0.707107
2	0.795961	0	0.605349

Fig 6.TF-IDF model example

2.4 Choosing the right Classification Algorithm:

Classification can be performed on structured as well as on unstructured data. Classification is a technique where we categorize data items into a given n number of classes. The primary goal of a classification problem is to identify the class to which a new data will fall under. It is based on the training set of data containing observation. There are different types of classification algorithm with a different method to solve a given classification problem. There is two main classification algorithm for Natural Language Processing. First one is the Naïve Bayes has been used in the various problem like spam detection, and the other is Support Vector Machine has also been used to classify texts such as progress notes. But in our case, the Naïve Bayes algorithm is best suited for rating prediction problem.

2.4.1Naïve Bayes Algorithm

The Naïve Bayes is a Machine Learning algorithm for classification problems. It is primarily used for text classification problems. It is mainly used for text classification, which involves high-dimensional training data sets. This algorithm learns the probability of an object with certain features belonging to a particular group in class, in short, we can say that it's a probabilistic classifier. The Naive Bayes algorithm is called "naive" because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features [1]. As to the Bayes part, it refers to the Baye's theorem. The Basis of the Naïve Bayes algorithm is the Bayes theorem also known as Bayes rule or Bayes law.

2.4.1.1 Bayes Theorem

Bayes Theorem is stated as Probability of the event B given A is equal to the probability of the event A given B multiplied by the probability of A upon the probability of B.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Fig 7. Bayes Theorem Formula

P(A|B)(Posterior)= Probability (conditional probability) of occurrence of event A given the event B is true.

P(A)(prior) & P(B) (Evidence of prior probability)= Probabilities of the occurrence of event A and B respectively.

P(B|A)(likelihood)= Probability (conditional probability) of occurrence of event B given the event A is true.

Now the formula can be written as:

$$\text{Posterior} = \frac{\text{(Likelihood).(Proposition prior probability)}}{\text{Evidence prior probability}}$$

Fig 8. Modified formula for Bayes Theorem

2.4.1.2 Bayes Theorem for Naïve Bayes Algorithm:

In an ML classification problem, there are multiple features and classes, say, C1, C2,…, Ck. The main aim in the Naive Bayes algorithm is to calculate the conditional probability of an object with a feature vector x1, x2, …, xn belongs to a particular class Ci,

$P(Ci|x1,x2,\ldots,xn)=P(x1,x2,\ldots,xn|Ci).P(Ci)/P(x1,x2,\ldots,xn)$ for $1\le i\le k$

After calculations and the independence assumption, the Bayes theorem comes down to the following easy expression:

$P(Ci|x1,x2,\ldots,xn)=(\prod j=1 j=nP(xj|Ci)).P(Ci)/P(x1,x2,\ldots,xn)$ fo r $1\le i\le k$

The expression P(x1,x2,…,xn) is constant for all the classes, we can simply say that

$P(Ci|x1,x2,\ldots,xn)\propto(\prod j=1 j=nP(xj|Ci)).P(Ci)$ for $1\le i\le k$

4	1	0	7	3
3	1	12	4	5
11	15	21	16	13
34	26	54	52	56
112	81	163	165	168

Fig 9. Confusion Matrix of Naïve Bayes

According to the above formula, we have predicted the rating of the user reviews and have found that the accuracy of Naïve Bayes classifier is approximately 24% for a dataset of 10,000 entries and long review size. The confusion matrix of the Naïve Bayes classifier is as shown in the figure 9.

3. Lexicon Method

Lexicon methods choose a sentiment lexicon to describe the polarity (positive, negative and neutral), by weighing and counting sentiment-related words that have been evaluated and tagged. This lexicon-based approach is more understandable and can be easily implemented in contrast to machine learning based algorithms. To collect the opinion word list, three main opinion approaches have been studied: the manual approach, the dictionary-based approach, and the corpus-based approach. The manual approach is time-consuming and thus it is not usually used alone, but combined with automated approaches as the final check because automated methods make mistakes. Two automatic methods are discussed below.

i) Corpus-based methods use a seed set of sentiment words with known polarity and make use of syntactic patterns of co-occurrence patterns to identify new sentiment words and their polarity in a large corpus.

ii) Dictionary-based methods make use of available lexicographical resources like WordNet or HowNet. The main strategy in these methods is to collect an initial seed set of sentiment words and their orientation manually, and then searching in a dictionary to find their synonyms and antonyms to expand this set. The new seed set is then used iteratively to generate new sentiment words.

3.1 Vader Lexicon

VADER (Valence Aware Dictionary for sentiment Reasoning) is a model used for text-based sentiment analysis that is sensitive to both polarity (positive/negative) and intensity (strength) of emotion [2]. VADER text sentiment analysis uses a human-centric approach, combining qualitative analysis and empirical validation by using human ratters and the wisdom of the crowd [2].

VADER sentiment analysis depends on a dictionary which maps lexical features to emotion intensity called sentiment scores. The sentiment score of a text can be obtained by summing up all the intensity of word in the text-based document. By lexical feature, we mean everything that we use for textual communication. Think of a review as an example. In a normal user review, we can usually find only words, but also emoticons, acronyms and slang. The thing about VADER sentiment analysis is that these colloquialisms get mapped to intensity values as well. Emotion intensity or sentiment score is measured on a scale from -4 to +4, where -4 is extremely negative and +4 is the extremely positive. The midpoint 0 which represents a sentiment as neutral. Emotional intensity can be very unpredictable since it depends on person to person. Some words might not seem very positive to you, but they might be to other people. To counter this type of problem, the creators of VADER sentiment analysis enlisted the number of human ratters and averaged their ratings for each word. This depends on the concept of the wisdom of the crowd.

VADER returns a sentiment score in the range -1 to 1, from most negative to most positive. The sentiment score of a sentence is calculated by summing up all the sentiment scores of VADER-dictionary-listed word in the sentence. Individual words have a sentiment score between -4 to 4, but in returned sentiment score of a sentence is between -1 to 1. The score of a sentence in a review is the sum of sentiment score of each word present in99 that particular sentence. Normalization is applied to the total sentiment score to bring down the value in the range of 1 to -1.

The formula for normalization can be given as:

$$\frac{x}{\sqrt{x^2+\alpha}}$$

Where,

x is the sum of the sentiment scores of the words which are present in the sentence.

Alpha is a normalization parameter.

The normalization is graphed below.

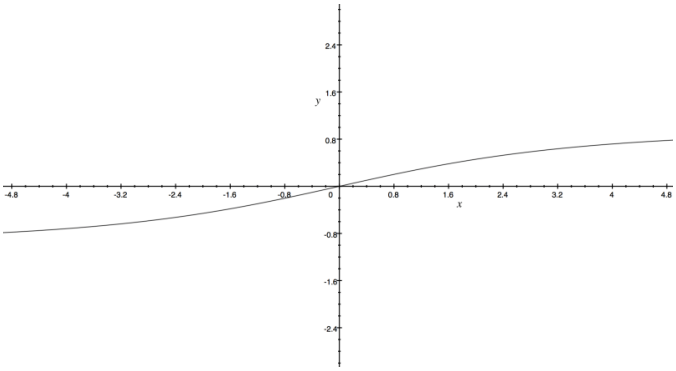


Chart -1: Normalization graph

Here x grows larger, it gets more and more close to 1 or -1. If there are a lot of words in the review score can be close to 1 or -1. VADER sentiment analysis works best on short documents, not on large documents. Thus, we will be using VADER lexicon in our combined approach for grouping the reviews.

4. The Combined Approach

In this combined approach we will predict the rating of the reviews on the basis of both the approaches in combination namely the machine learning approach and the lexicon-based approach. Firstly we will consider the prediction of 3, 4, 5-star rating from review as it is positive reviews and the same can be done for negative reviews. Firstly after importing our dataset of nearly 10,000 reviews, we selected nearly 800 reviews and classified it to positive and negative using the VADER semantic lexicon. This 800 reviews can be considered as the test set of this approach but here the negative reviews will be neglected. Then, excluding those 800 reviews, the other 9200 reviews are used as the training reviews. From those training set reviews, only those reviews are selected which have 3, 4, 5 stars labeled rating in the dataset because it comes under positive review. These positive reviews will then be pre-processed and then will be trained. Then on the basis of those reviews, the machine learning model will be prepared. After the model is prepared the classified review from the VADER lexicon will be further classified on the scale of 3, 4, 5 stars on this machine learning model. The reviews will be classified on the scale of 3, 4, 5 stars because here we are considering the positive reviews. And hence we can get our result. The same procedure applies to the negative reviews to classify the review on 1 and 2-star rating. The machine learning model for positive and negative reviews is different. The larger the dataset the more accuracy can be obtained. And hence the result can be obtained. The flow chart is given below for better understanding.

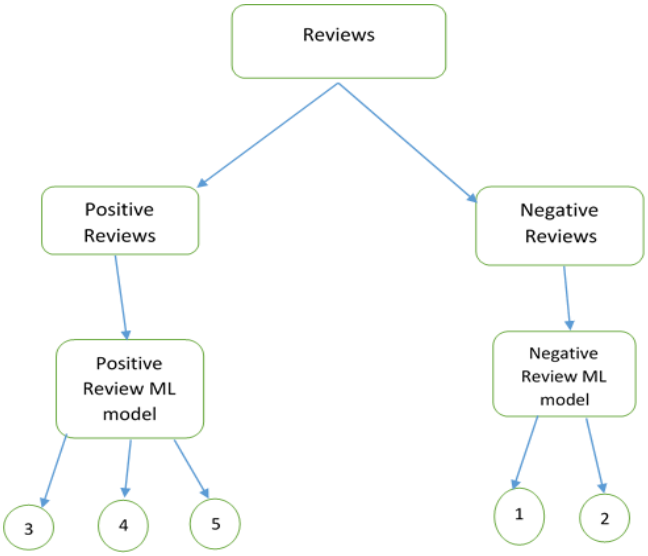


Fig -10: Flow chart of COMBINED APPROACH.

This approach is highly accurate if there is large dataset and the textual review are in small size. This is because large dataset will prepare accurate machine learning based model and short textual review will be more accurately classified by the VADER lexicon. Both go hand in hand. This combined approach method for a large dataset of 10,000 rows and large text size is approximately 30% accurate which beats the accuracy of Naïve Bayes model for the same dataset with the accuracy of 24% and also the other supervised machine learning classification algorithm.

0	3	2	2	4
2	1	2	4	7
4	2	8	10	20
23	22	26	61	51
66	71	99	122	149

Fig 11. Confusion Matrix

Here there are lots of sample miss-classified on the first stage itself because the text in the reviews is very large. The model in the screenshot gives an accuracy of 29%.
If the text size is small in the reviews and has large dataset the accuracy can be increased to approximately 50%.

3. CONCLUSIONS

In this paper, we have implemented the combined approach of machine learning and lexicon-based approach and we found that nearly 5% to 10% increase in accuracy if the review size is long and the dataset is large. If the review size is small and the dataset is large nearly 20% increase can be obtained using a combined approach.
The interest in the domain of sentiment analysis as a field of research is excelling rapidly. It can show that the transformation of the huge volume of textual data from the web into meaningful information can be very useful. However, the task of accurate opinion extraction from unstructured data will still remain challenging.

REFERENCES

[1] Rashmi Jain, "Introduction to Naive Bayes Classification Algorithm in Python and R" https://www.hackerearth.com/blog/machine-learning/introduction-naive-bayes-algorithm-codes-python-r/ . Accessed: 30-Mar-2019.

[2] Pio Calderon, "VADER Sentiment Analysis Explained" http://datameetsmedia.com/vader-sentiment-analysis-explained/. Accessed: 15-Mar-2019.

[3] Thársis T. P. Souza, Olga Kolchya, Tomaso Aste and Philip Treleaven, "Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination," Handbook of Sentiment Analysis in Finance. Mitra, G. and Yu, X. (Eds.).(2016). ISBN 1910571571

[4] Xiaojiang Lei, Xueming Qian, Member, IEEE, and Guoshuai Zhao," Rating Prediction based on Social Sentiment from Textual Reviews," in IEEE TRANSACTIONS ON MULTIMEDIA, MANUSCRIPT ID: MM-006446

[5] F. Li, N. Liu, H. Jin, K. Zhao, Q. Yang, X. Zhu, "Incorporating reviewer and product information for review rating prediction," in Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, 2011, pp. 1820-1825.

[6] Harpreet K., Veenu M., Nidhi., "A Survey of Sentiment Analysis techniques" International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017) Mining Workshops. IEEE, 2011.

[7] M.Trupthi, Suresh Pabboju, G.Narasimha, "Sentiment Analysis On Twitter Using Streaming API", 2017 IEEE 7th International Advance Computing Conference, January 2017, pg. 915-919.

[8] Huma Parveen, Prof. Shikha Pandey, "Sentiment Analysis on Twitter Data-set using Naïve Bayes Algorithm", 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), July 2016, pg. 416-419.