International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 06 | June-2018 www.irjet.net p-ISSN: 2395-0072

Sentiment analysis of Twitter data: A machine learning approach to analyse demonetization tweets

Brinda Hegde[1], Nagashree H S[2], Madhura Prakash[3]

[1,2]*VTU & Information Science and Engineering BNMIT*
[3]*Assistant Professor, Dept. of Information Science and Engineering, BNMIT, Karnataka, India*

Abstract - *With 500 Million Tweets sent each day, that is 6000 Tweets being generated every second, Twitter is the most popular micro-blogging site that allows users to express their views and opinions in 280 characters. As companies and political leaders take to the online social media platform to establish and develop their brand, one cannot ignore the amount to data being generated on Twitter. The proposed system aims to extract and analyse tweets, classify them as positive or negative with the help of machine learning techniques and algorithms, and finally subject to performance evaluation techniques. On November 8, 2016, in a television broadcast, Prime Minister Narendra Modi declared that all the 500 and 1000 rupee notes were illegal in an effort to curb black money and fake notes. Considering the demonetization dataset extracted from Twitter using Twitter API, pre-processing is performed using NLTK and Scikit-learn, which is then subjected to algorithmic executions such as Naive Bayes, Logistic Regression and Support Vector Machines. A comparison of this execution is considered to determine which algorithm works better for given dataset in terms of precision, recall, accuracy and F1 Score.*

***Key Words*: Twitter, demonetization tweets, sentiment analysis, feature extraction, supervised machine learning algorithm**

1. INTRODUCTION

Twitter has become one of the most-used micro blogging website with about 271 million active users generating in excess of 500 million tweets a day; it is an interesting source of information. Twitter has limited message size allowing only 280 characters for the users to make use of. Twitter is therefore challenging their users to express their view in one or two key sentences. Demonetization is an event that brought immense changes to India both economically and socially. The proposed system focuses on demonetization tweets. The demonetization tweets are to be expressed in a simple word: Positive or Negative by subjecting the dataset to different algorithmic executions in order to determine which algorithm is best suited for Sentiment Analysis based on the given dataset.

1.1 Sentiment Analysis

Sentiment analysis is the process of determining whether a piece of writing or text is positive, negative or neutral. Usually, it is used to arrive at a binary decision such as for/against, good/bad or like/dislike. It is also called 'Opinion Mining' or 'Emotion AI'. In the marketing field, companies use it to develop their strategies, to understand customers' feelings towards products or brand, how people respond to product launches and why consumers don't buy some products. In the political field, it is used to keep track of political views, to detect consistency and inconsistence between statements and actions at the government level, also to predict election results as well.

2. RELATED WORK

In the field of Sentiment Analysis, a lot of research work has been done in the past. Some of the papers studied are as follows:

Gupta *et al.* [1] found a novel approach to filter tweets location-wise and at the same time compare the performance in terms of accuracy and efficiency. The author implements Naïve Bayes and SVM classifier to classify the twitter dataset into positive and negative. Then the locations are classified and sentiment maps are made. This is useful in understanding the sentiments of every state in India separately. Some features were added to the code which included the addition of latitude, longitude, number of kilometer and number of tweets to classify the sentiments region wise. The author also demonstrates the effect of pre-processing of datasets. Though the author considered state-wise reaction to demonetization, due to population polarisation, the overall sentiment of the netizens could not be captured.

Gautam *et al.* [2] implemented supervised algorithms such as SVM, Naïve Bayes and maximum entropy to classify the twitter dataset based on sentiments. The results obtained from algorithms are compared based on their relative performances on three parameters namely: accuracy, precision and recall. After training and classification, a semantic analysis, derived from the WordNet database. A comparative measurement is taken on the classification using supervised learning algorithms and the semantic analysis.

Tsapatsoulis *et al.* [3] aimed to check whether tokens, manually indicated by humans during tweet annotation, can form an index of terms that can be used for training effective tweet classification models. The author compared the human-created index of terms with several automatically extracted features sets for tweet classification, under a machine learning framework, and by using three different classifiers to justify author's claim. The 3 approaches author identified

were lexicon based approach, machine learning approach and the social approach. Author observed that tokens identified explicitly by human showed best performance among all other feature extraction techniques. However, the result author observed did not consider feature set combination.

Parveen *et al.* [4] presented a HDFS architecture and MapReduce technique. For implementation, the dataset is firstly processed, and then a supervised learning algorithm, Naïve Bayes, is applied. To implement Naïve Bayes algorithm a trained SentiWordNet dictionary is needed which is available online. Two methodologies are used to implement Naïve Bayes. Method 1 works on maps phase which reads the content of the SentiwordNet dictionary from a file and transform into the Hash map for key-value based polarity retrieval of words for faster processing. Method 2 is a reduce phase that collects the overall polarity of each tweet and transform into 5 different categories as extreme positive, positive, extreme negative, negative and neural. Although the classification transformed tweets into 5 different categories, the implementation was limited to only one algorithm.

Abdelwahab *et al.* [5] compared the effect of training set size on SVM and Naive Bayes. The author studied the effect of varying the training set size on the learning curves of both SVM and Naïve Bayes when used in twitter sentiment classification. In addition, author also examined the impact of the training set on different ensemble fusion types. Ensemble 1 where result of SVM and Naïve Bayes classifier are AND fused found to perform better than ensemble 2 which had SVM and Naïve Bayes classifiers OR fused as the false output of Naïve Bayes would be nullified by AND fusion. But combining the results of classifiers resulted in ambiguous results for the comparison between Naive Bayes and SVM.

Neethu *et al.* [6] studied the two ways of extracting the sentiment from the dataset; symbolic technique and machine learning technique. The unsupervised sentiment classification using symbolic techniques makes use of available lexical resources. For machine learning technique, a number of techniques like Naive Bayes, Maximum Entropy and SVM are used to classify reviews. The features extracted using Term Presence, Term Frequency, negation, n-grams and Part-of-Speech are used to find out the semantic orientation of words, phrases, sentences and that of documents. The author observed that Machine Learning techniques are simpler and efficient compared to Symbolic techniques.

Sahni *et al.* [7] used the subjectivity of tweets to classify dataset. Usually a purely objective sentence does not convey any sentiment. Hence, to reduce the number of training datasets, only pure subjective sentences are considered. Before implementing classification algorithms, various pre-processing techniques such as TextBlob are used to filter the subjective sentences. The different feature extraction techniques considered are n-grams and POS.

Trupthi *et al.* [8] illustrated the importance of pre-processing the training set. Pre-processing will reduce the amount of data by removing unwanted data. NLTK is used to remove the words with POS tags which are not useful to build the classifier. Hadoop is used to extract information from it and MapReduce is used to easily extract several words with their positive and negative probabilities. The output of reducer is several numbers of words with their positive and negative scores.

3. PROPOSED SYSTEM

The proposed system performs the process of sentiment analysis on Twitter demonetization data. Also a performance comparison on different techniques is done. The total size of the dataset is 12000 tweets, this is varied and performance parameters are measured for 25%, 50%, 75% and 100% of the size of the dataset.

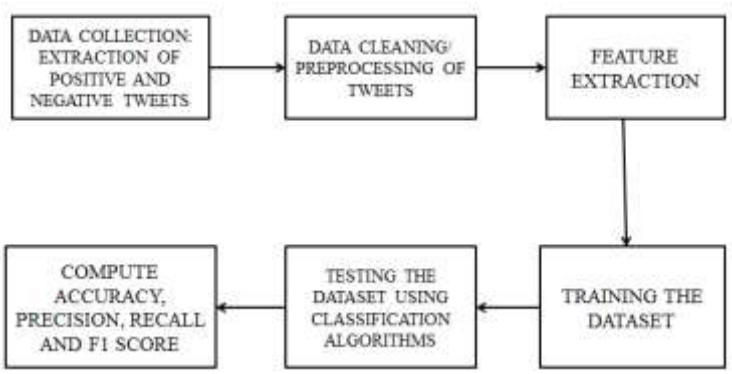


Figure 1: Overall flow of Sentiment Analysis

Figure 1 summarizes the complete flow of modules from data extraction to sentiment analysis and performance evaluation. The description for each of the modules and algorithms used are described in the following section.

3.1 Twitter data collection

The Twitter demonetization tweets are being collected using Tweepy package in python using the Twitter API.

3.2 Pre-processing the twitter data

Pre-processing involves removal of unimportant features from the data. In this phase, several techniques like Stemming and Stop word removal are applied to data set for noise reduction and facilitating feature extraction.

- **Stemming and Lemmatization:** Stemming and Lemmatization are two essential morphological processes of pre-processing module during feature extraction. Stemming removes word inflections only whereas lemmatization replaces words with their base form. For example, the words “caring” and “cars” are reduced to “car” in a stemming process whereas lemmatization reduces it to “care” and “car” respectively, hence lemmatization is considered to be more accurate. Unlike stemming, lemmatization needs additional dictionary support for searching and indexing, which enhances its accuracy in feature extraction applications.

- **Stop word removal:** Stop words are common and high frequency words like "a", "the", "of", "and", "an". Different methods available for stop-word elimination ultimately enhance performance of feature extraction algorithm. The stop words removal reduces dimensionality of the data sets. Words to be removed are taken from a commonly available list of stop words using NLTK.

Also, the tweets are searched for hyperlinks and URLs and are removed, along with punctuations and stop words.

3.3 Feature Extraction

Feature extraction is to extract features in a format supported by machine learning algorithms from datasets consisting of formats such as text and image. Feature extraction is carried out using:

- **Bag of Words:** This technique involves the following tasks:
  1. Tokenizing strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.
  2. Counting the occurrences of tokens in each document.
  3. Normalizing and weighting with diminishing importance tokens that occur in the majority of samples / documents.

- **TF-IDF (Term Frequency-Inverse Document Frequency):** The tf–idf transform is used in order to re-weight the count to shadow the frequencies of rarer yet more interesting features into floating point values suitable for usage by a classifier. Tf means term-frequency while tf–idf means term-frequency times inverse document-frequency. The equation 1 is the description for TF-IDF.

$$tf\text{-}idf(t,d) = tf(t,d) * idf(t) \quad \text{.....(1)}$$

- **N Grams**: An N Gram is a contiguous sequence of *n* terms from a given sequence of text. An n-gram of size 1 is referred to as a unigram; an n-gram of size 2 is a bigram; an n-gram of size 3 is a trigram and so on.

3.4 Training the dataset and applying algorithms

The dataset is divided into training and testing set using KFold cross validation technique with the value of k set to 10. The project implements 3 algorithms for preparing and training the model. The following are the 3 algorithms which are implemented.

- **Naïve Bayes:** Naïve Bayes Classifier is a probabilistic classifier based on applying Bayes' theorem with strong independence assumption that the presence of one feature in a class does not depend on the presence or absence of another feature. Naïve Bayes is a simple model which works well on text categorization. For tweets a multinomial Naïve Bayes model can be used.

$$c^* = arg_{max}\, c\, PNB\, c\, d \quad \text{.....(2)}$$

$$PNB\, c\, d := (c)\ (f|c)\ n_i(b) \quad \text{.....(3)}$$

Class c* is assigned to tweet d is represented in equation 2. In the equation 3, f represents a feature and n_i (d) represents the count of feature f_i found in tweet d. There are a total of m features. Parameters P(c) and P (f|c) are obtained through maximum likelihood estimates. P(c) is prior probability.

- **Support Vector Machine(SVM)**: Another algorithm for solving the text classification problem is Support Vector Machine (SVM). Support Vector Machine is a supervised machine learning algorithm which can be used for both classification and regression challenges. In this algorithm, each data item is plotted as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. It tries to find a hyper-plane which separates the data in two classes as optimally as possible.

- **Logistic Regression(LR):** Logistic regression is a statistical method for analysing a dataset in which there are one or more independent variables that determine an outcome. In logistic regression, the dependent variable is binary or dichotomous, i.e. it only contains data coded as 1 (TRUE, success, fraud etc.) or 0 (FALSE, failure, not-fraud, etc.). The goal of logistic regression is to find the best fitting model to describe the relationship between dependant variable and a set of independent variables. Logistic regression generates the coefficients (and its standard errors and significance levels) of a formula to predict a logit transformation of the probability of presence of the characteristic of interest. Equation 4 represents the formula for logit.

$$\text{logit(p)} = b_0 + b_1X_1 + b_2X_2 + \cdots + b_kX_k \quad \text{.....(4)}$$

where p is the probability of presence of the characteristic of interest. The logit transformation is defined as the logged odds. Equation 5 represents the logged odds and equation 6 defines the logit.

$$\text{odds} = \frac{p}{1-p} = \frac{probability\ of\ presence\ of\ characteristic}{probability\ of\ absence\ of\ characteristic} \quad \text{....(5)}$$

$$\text{logit(p)} = \ln\left(\frac{p}{1-p}\right) \quad \text{.....(6)}$$

Rather than choosing parameters that minimize the sum of squared errors estimation in logistic regression chooses parameters that maximize the likelihood of observing the sample values.

3.5 Performance Evaluation

The dataset is then subject to evaluation in the following criteria:

- **Accuracy** is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.
- **Precision** is the ratio of correctly predicted positive observations to the total predicted positive observations.
- **Recall** is the ratio of correctly predicted positive observations to the all observations in actual class.
- **F1 Score** is the weighted average of Precision and Recall. F1 score is more helpful than accuracy in uneven distribution.
-

The equations of accuracy, precision, recall and F1 Score is represented in equations 7, 8, 9 and 10 respectively.

$$\text{Accuracy} = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad \ldots\ldots\ldots (7)$$

$$\text{Precision} = \frac{t_p}{t_p + f_p} \quad \ldots\ldots\ldots (8)$$

$$\text{Recall} = \frac{t_p}{t_p + f_n} \quad \ldots\ldots\ldots (9)$$

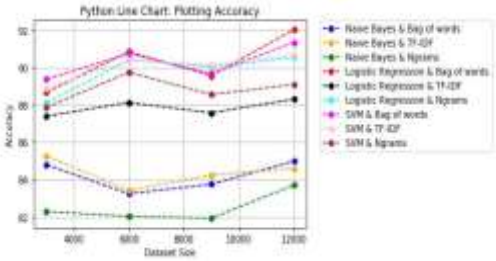
$$\text{F1 Score} = \frac{2 * recall * precision}{recall + precision} \quad \ldots\ldots\ldots (10)$$

where, t_p is true positive which correctly predicted positive values, t_n is true negative which correctly predicted negative values, f_p is false positive which is falsely predicted positive class and f_n is falsely predicted negative class.

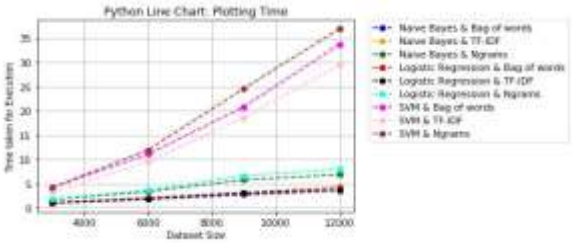
RESULTS

Accuracy, recall, precision, F1 score and time taken for execution are measured for each classification algorithm (Naive Bayes, Logistic Regression and Support Vector Machines) which is applied on all the 3 feature extraction techniques (Bag of Words, TF-IDF and N-Grams) being implemented, varying the dataset.

The figure 2 demonstrates pictorially, the variation of accuracy and time taken respectively with different dataset sizes. The figures depicts that Logistic Regression with bag of words has the highest accuracy but takes relatively more time compared to Naïve Bayes and less time compared to Support Vector Machine.



(a)Plot of Accuracy vs dataset size

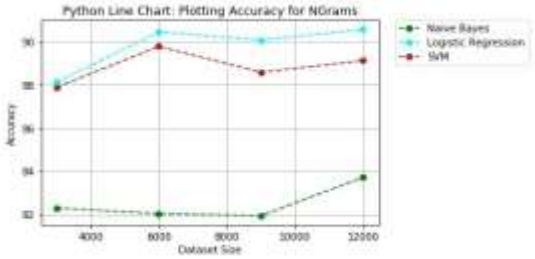


(b) Plot of Time Taken for vs dataset size

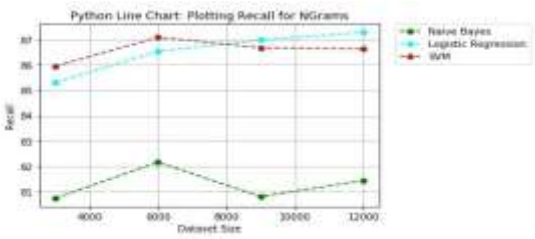
Figure 2: Plot of Accuracy and Time Taken for Training and Testing

The Support Vector Machine with N Grams for 12000 takes the highest time. Naïve Bayes with bag of words takes the least time for execution but the accuracy is not very good. Logistic Regression, compared to Support Vector Machine takes less time and provides high accuracy. The accuracy of algorithms increases with increase in size of data. But an anonymous behaviour is being exhibited by all algorithms. In Naïve Bayes, the accuracy initially drops and then increases whereas in case of Logistic Regression and Support Vector Machine, the accuracy drops for 9000 dataset, and increases for 12000 dataset.

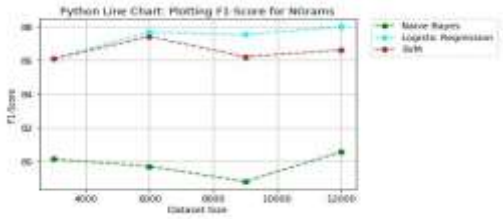
According to figure 2(a), as the size of the dataset increases to 12000 tweets, there is significant growth in the accuracy in all the algorithms—indicating that higher the size of the dataset, better is the accuracy. Naive Bayes accuracy stays below a limit of 86% whereas Support Vector Machines and Logistic Regression have higher accuracy that fall within the same range. Even though, Support Vector Machines is expected to perform better, Logistic Regression surprisingly outperforms Support Vector Machines in certain cases. According to the Figure 2(b), Naive Bayes and Logistic Regression do not take more than 10 seconds for training, testing and cross-validation. Though Support Vector Machines performs better than the others, it does take a huge amount of time for execution. Logistic Regression's accuracy is comparable with that of Support Vector Machines and takes less time.



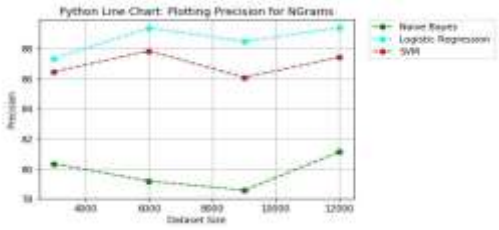
(a) Accuracy for N Grams



(b) Recall for N Grams



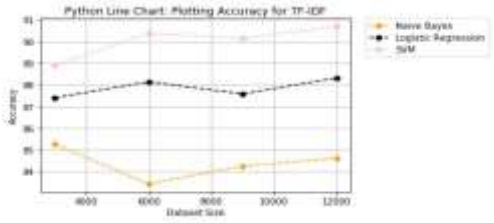
(c) F1 score for N Grams



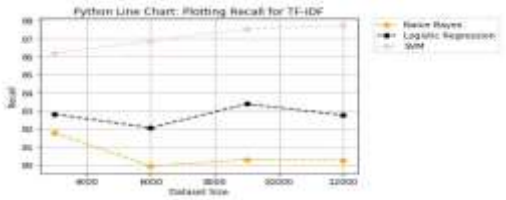
(d) Precision for N Grams

Figure 3: Plots for N-Grams (a) Accuracy, (b)Recall, (c)F1-score and (d) Precision

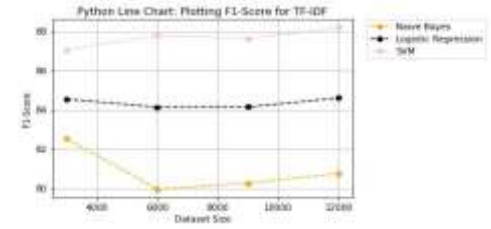
The Figures 3 depicts accuracy, recall, F1 score and precision on N Grams for Naïve Bayes, Logistic Regression and SVM varying the dataset from 3000 to 12000. Logistic Regression has the highest accuracy, precision and F1 Score. The recall value of SVM for 3000 and 6000 dataset is greater than recall value of Logistic Regression. But the recall value for 9000 and 12000 dataset of Logistic Regression is higher than SVM. The Naïve Bayes algorithm accuracy, precision and F1 Score is lesser compared to Logistic Regression and SVM and the value initially decreases with increase in dataset size. After 9000, the value increases whereas the recall value for Naïve Bayes has highest point at 6000 and then decreases with increase in dataset size.



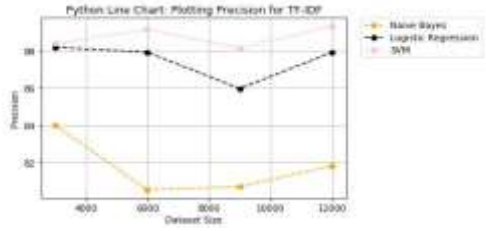
(a)Accuracy for TF-IDF



(b) Recall for TF-IDF



(c) F1 score for TF-IDF

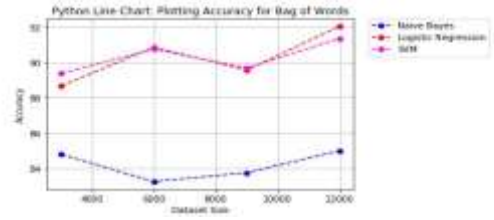


(d) Precision for TF-IDF

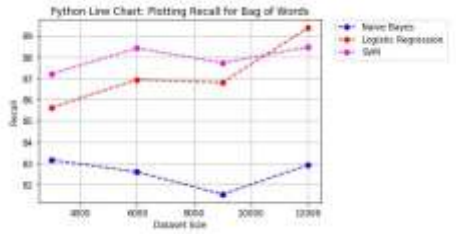
Figure 4: Plots for TF-IDF (a) Accuracy, (b)Recall, (c) F1-score and (d) Precision

The figure 4 depicts accuracy, recall, F1 score and precision on TF-IDF for Naïve Bayes, Logistic Regression and SVM varying the dataset from 3000 to 12000.

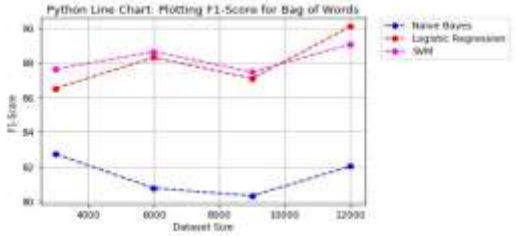
The algorithms show better accuracy, precision, recall and F1 Score values compared to N Grams. But with TF-IDF, SVM outperformed Logistic Regression. The behaviour shown by algorithms with change in dataset size is similar to N Grams.



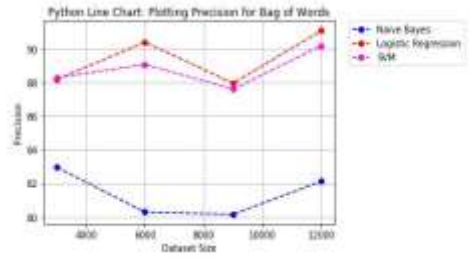
(a) Accuracy on Bag of Words



(b) Recall on Bag of Words



(c)F1 score on Bag of Words



(d)Precision on Bag of Words
Figure 5: Plots for Bag of words (a) Accuracy, (b)Recall, (c) F1-score and (d) Precision

The figure 5 depicts accuracy, recall, F1 score and precision on Bag of words for Naïve Bayes, Logistic Regression and SVM varying the dataset from 3000 to 12000.

Logistic Regression has highest accuracy, precision, F1 Score and recall for 12000 dataset. But the accuracy of SVM is greater than that of Logistic Regression for 3000 dataset. The Naïve Bayes algorithm accuracy, precision, F1 Score and recall initially decreases with increase in dataset till 9000 and then it increases.

Even though N-Grams was expected to perform better as it adds context to the tweets, the graphs prove otherwise. Surprisingly, Bag-of-Words performs better. It can be observed clearly that Logistic Regression for all feature extraction techniques has higher Precision values than Recall values. Higher precision indicates most of the true positives have been labelled correctly while low recall implies that it returns less true positives. Though F1-score is considered as a better scoring criterion than accuracy, it is observed that the F1-scores are almost equivalent to the Accuracies. This can be attributed to the fact that the dataset has binary classification and sentiment has only exactly one value.

4. CONCLUSIONS

Classifying the sentiment of Twitter data has become a common yet an interesting challenge not only for data scientists but also for growing businesses. Also, the feature extraction techniques must be taken into account along with the workings of different algorithms in order to determine which is better. As a part of this work, a software solution has been developed that compares the different feature extraction techniques such as Bag of Words, TF-IDF and N-Grams. The cleaned dataset's size has been varied and is subject to executions of Naive Bayes, Support Vector Machines and Logistic Regression.

Using a large dataset has showed improved accuracy and better outcomes. Naive Bayes performs satisfactorily but does not exceed expectations. Though Support Vector Machines gave better accuracy, its large execution time defeats the purpose of an efficient classifier. Logistic Regression performs as well as Support Vector Machines and takes as little time as Naive Bayes. As Sentiment Analysis is a vast domain, there can be much scope in the detection of sarcasm in the tweets and also, stream tweets in real-time and give real-time analysis and results.

REFERENCES

[1] Falguni Gupta, Swati Singhal, Amity University, "Sentiment Analysis of the Demonetization of Economy 2016 India, Regionwise", 2017 7th International Conference on Cloud Computing, Data Science & Engineering, January 2017, pg. 693-696.

[2] Geetika Gautam, Divakar Yadav, "Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis", 2014 Seventh International Conference on Contemporary Computing (IC3), August 2014, pg.437-442.

[3] Nicolas Tsapatsoulis, Constantinos Djouvas, "Feature Extraction for Tweet Classification: "Do the Humans Perform Better?" 2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), July 2017, pg. 53-58.

[4] 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.

[5] Omar Abdelwahab, Mohamed Bahgat, Christopher J. Lowrance, Adel Elmaghraby, "Effect of Training Set Size on SVM and Naïve Bayes for Twitter Sentiment Analysis", 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), December 2015, pg. 46-51.

[6] Neethu M S, Rajashri R, "Sentiment Analysis in Twitter using Machine Learning Techniques", 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), July 2013,pg. 1-5.

[7] Tapan Sahni, Chinmay Chandak, Naveen Reddy Chedeti, Manish Singh, "Efficient Twitter Sentiment Classification using Subjective Distant Supervision", 2017 9th International Conference on Communication Systems and Networks (COMSNETS), January 2017, pg. 548-553.

[8] 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.