

Automatic Sentiment Classification of tweets using Natural Language Processing

Reshma Gulwani^{1*} Rohit Singhal²

¹ Research Scholar, Sunrise University, Alwar Rajasthan, India

² Research Guide, Sunrise University, Alwar Rajasthan, India

Abstract – Text-based communication such as an email tweet and our statuses on a daily basis become one of the most common forms of expression. As a result, lot of unstructured data is generated. So analyze large quantities of text data is now a key way to understand what people is thinking. Sentiment Analysis is the keen area of research which concentrates on analyzing the opinions of users about any topic and classifies them into positive or negative reviews. Text like tweets on twitter helps us find trending topics in the world. Sentiment analysis is one of the important task of natural language processing Natural language processing techniques are used to analyze text, providing a way for computers to understand human language. Data is collected from twitter on railway services reviews. NLTK is used for building Python programs to work with human language data. Different machine learning approaches are used to build the classification model for training and testing the data. The performance of these models are evaluated and compared by using accuracy metric.

-----X-----

I. INTRODUCTION

Sentiment analysis is a natural language processing technique used for detecting positive, negative or neutral sentiments in a text. Sentiment analysis helps the businesses to monitor brand and product sentiment in customer feedback, and understand customer needs. Since customers are now more open in expressing their thoughts and feelings than before, sentiment analysis is becoming an important to monitor and understand that sentiment. Automatically analyzing customer feedback through survey responses and social media conversations, allows brands to learn how to make customers happy. For example, automatically analyse 5,000+ reviews about your product using sentiment analysis on social media could help to find out if customers are satisfied or not about your pricing plans and customer service. Twitter is one of the largest online social media platform used by many people nowadays. Twitter is a microblogging and social networking service founded in 2006 on which users post messages or tweets. Tweets are limited to 140 characters, leaving 20 characters for the username. People are expressing their opinions about the different products, services such as tourism, hotels, movies, and politics on twitter. A railway service is also one of the important topics because trains are one of the popular modes of transport. It is more convenient and comfortable for the long routes.

India is a country with second largest population in the world after china [1]. People mostly would like to travel in the trains for long distance journey because of convenience of comfort, low fare etc. Present world is equipped with smart devices, high speed internet, Wi-Fi facilities and social media platforms. Nowadays, people are using twitter as social media platform to express their emotions or sentiments about the various railway services such as food quality, cleanliness, quality of air conditioner and so on. Figure 1 illustrates some tweets



Fig.1: Positive tweet

II. LITERATURE SURVEY

Pak, A., &Paroubek [3] suggested creation of the twitter corpus which automatically collects tweets and annotates using emoticons. As the classification of the training set is purely based on the polarity of emotions, and this creates a chance of error in the classification of tweets. Abdelwahab, O., Bahgat, M., Lowrance, C. J., & Elmaghraby[4] provides ensemble framework for classification. Ensemble framework is done by combining several techniques and features. This ensemble has a Naïve Bayes classifier and an SVM classifier. AND gate is used to fuse the results produced by both classifiers. If the result of both classifiers is positive, then the output of ensemble classification will be positive. In all other cases, the ensemble classifies the tweet as negative. R. Parikh and M. Movassate [5] implement Naive Bayes bigram model and a Maximum Entropy models to classify tweets. Naive Bayes classifiers worked much better than the Maximum Entropy model. Go, R. Bhayani, L.Huang [6] proposed distant supervision is used to find the sentiment analysis for twitter data by in which training data composed tweets with emoticons which served as noisy labels. Naive Bayes, MaxEnt and Support Vector Machines (SVM) are used to bulid the model. Unigrams, bigrams and POS features are considered. They concluded that

SVM outperformed other models and that unigram were more effective as features. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau[7] developed 3-way model is for classifying sentiment into positive, negative and neutral classes. Models such as: unigram model, a feature based model and a tree kernel based model are used for experiments .Tweet is represented as tree tree kernel based model .The feature based model uses 100 features and the unigram model uses over 10,000 features. Features which combine prior polarity of words with their parts-of-speech (pos) tags are most important and play a major role in the classification task. The tree kernel based model outperformed the other two models. Dmitry Davidov, Ari Rappoport [8] suggested twitter user-defined hastags in tweets are used as a classification of sentiment type using punctuation, single words, n-grams and patterns as different feature types, which are then combined into a single feature vector for sentiment classification. K-Nearest Neighbor strategy is used to assign sentiment labels by constructing a feature vector for each example in the training and test set. Po-Wei Liang, Bi-Ru Dai [10] collected the data from Twitter API. Three different categories of data such as Camera movie and mobile data are used for training. The labels are used as positive, negative and non-opinions. Opinions were filtered in tweets. Unigram Naive Bayes model was implemented and the Naive Bayes simplifying independence assumption was employed. Useless features are eliminated by using the Mutual Information and Chi square feature extraction method. Finally, the orientation of a tweet is predicted. i.e. positive or negative. Pablo Gamallo, Marcos Garcia, "Citius [11] presented variations of Naive Bayes classifiers for detecting polarity of English tweets. Two different variants of Naive Bayes classifiers were built namely Baseline and Binary. Baseline is trained to classify tweets as positive, negative and neutral. Binary makes use of a polarity lexicon and classifies as positive and negative. Neutral tweets neglected. Features such as Lemmas (nouns, verbs, adjectives and adverbs), Polarity Lexicons, and Multiword from different sources and Valence Shifters were considered for classifier. Turney, P. [12] proposed Bag-of-words method is for sentiment analysis in which the relationships between words was not considered and a document is represented as just a collection of words. To determine the sentiment for the whole document, sentiments of every word was determined and those values are united with some aggregation functions. The lexical database WordNet is used to determine the emotional content of a word along different dimensions. Distance metric on WordNet is developed and semantic polarity of adjectives is determined. R. Xia, C. Zong, and S. Li, [13] used an ensemble framework for Sentiment Classification which is obtained by combining various feature sets and classification techniques. Part-of-speech information and Word relations are used as features and three base classifiers such as Naive

Bayes, Maximum Entropy and Support Vector Machine are used. They applied ensemble approaches like fixed combination, weighted combination and Meta-classifier combination for sentiment classification and obtained better accuracy. ZhunchenLuo, Miles Osborne, TingWang, [14] suggested the challenges and efficient techniques to mine opinions from Twitter tweets are highlighted. Spam and wildly varying language makes opinion retrieval within Twitter challenging task. Aisopos, Fotis, et al [15] presented microblog content, some serious challenges. Some of these are the applicability of sentiment analysis used in past and different classification methods caused by their inherent characteristics of content. To resolve them, a method that relies on two orthogonal and complementary sources of evidence: context-based method captured by polarity ratio and content based features acquired by n-gram graphs is introduced. Both the methods are language-neutral and tolerant to noise; guarantee high robustness and effectiveness. Jebaseeli, A. Nisha, and E. Kirubakaran [16] presented model which collects tweets from social networking sites and thus provide a view of business intelligence. There are two layers in the sentiment analysis tool, the data processing layer and sentiment analysis layer. Data processing layer deals with data collection and data mining. .

Manual tracking and extraction of the useful information from twitter is not possible, so sentiment analysis is needed. In this research sentiment analysis is used which automates the process of identifying and classifying subjective information in text data on twitter. This might be an opinion, a judgment, or a feeling about a particular topic or product feature. The most common type of sentiment analysis is 'polarity detection' and involves classifying statements as positive, negative or neutral. Sentiment analysis uses Natural Language Processing (NLP) to make sense of human language, and machine learning to automatically deliver accurate results.

III. PROPOSED METHODOLOGY

In this research, the work is initiated by started by collecting twitter data that would be pre processed further. Pre-processing is done so the data can be fit for feature extraction and polarity of the data has been fetched as positive or negative. Then Naive Bayes classifier is applied to check the accuracy of the model. Figure 2 shows the proposed architecture.

A. Tweet Collection

The tweets have been downloaded from the twitter by using tweepy package. The script to download the tweets is written in python. The hash tags are used to extract tweets. Twitter Application Programming Interface (API) [2]. To fetch tweets from the Twitter API an app needs to be registered through your

Twitter account. After creating an app the following steps have been performed:

- Open <https://apps.twitter.com/> and click the button
- 'Create New App'.
- Fill the details asked.
- When the App is created, the page will be automatically loaded.
- Open the 'Keys and Access Tokens' tab.
- Copy 'Consumer Key', 'Consumer Secret',

'Access token' and 'Access Token Secret'.

Once the dataset is ready for processing, then train a model on pre-classified tweets and use the model to classify the sample tweets into negative and positives sentiments. The tweets with no sentiments will be used to test your model.

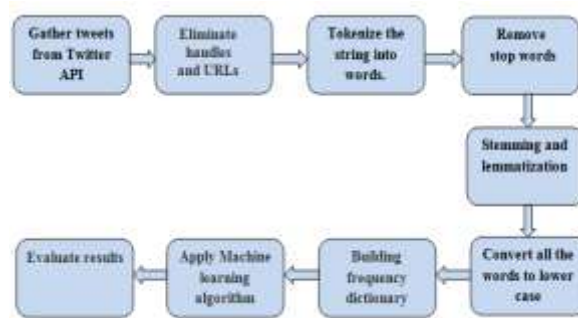


Fig. 2: Proposed Architecture

B. Tweet Pre-processing

Pre-processing is very important step in while sentiment analysis process. Natural Language processing (NLP) techniques are used to perform pre-processing of the tweets. NLTK package in python is used for all NLP tasks. The pre-processing is divided in to following phases

1) Tokenization

Language in its original form cannot be accurately processed by a machine, so you need to process the language to make it easier for the machine to understand. The first part of making sense of the data is through a process called tokenization, or splitting strings into smaller parts called tokens.

A token is a sequence of characters in text that serves as a unit. Based on how you create the tokens, they may consist of words, emoticons, hashtags, links, or even individual characters. A basic way of breaking language into tokens is by

splitting the text based on whitespace and punctuation. The punkt module is a pre-trained model that helps to tokenize the words and sentences. Tokenized () method is used for tokenizing the tweets. Tokenized () method returns special characters such as @ and _. These characters will be removed through regular expressions.

Table I: Algorithm for Word Tokenization

Input: Filtered Tweets
Output: Tokenize words
For all words in Processed Tweets
Tokenize the word passing to Tweet Tokenizer Method and append Tokenize Sentence
Return Tokenize Sentence

2) Remove Stop Words

- Noise is any part of the text that does not add meaning or information to data. Noise is specific to each system, so what represent noise in one system may not be in another system. The most common words in a language are called stop words. Some examples of stop words are “is”, “the”, and “a”. They are generally irrelevant when processing language, unless a specific use case warrants their inclusion.
- Regular expressions are used to search for stop words and remove these items:
- Hyperlinks**

All hyperlinks in Twitter are converted to the URL shortener t.co. So keeping them in the text processing would not add any value to the analysis. To remove hyperlinks, you need to first search for a substring that matches a URL starting with http:// or https://, followed by letters, numbers, or special characters.

Twitter handles in replies

These Twitter usernames are preceded by a @ symbol, which does not convey any meaning.

Punctuation and special characters

They provide context to textual data, this context is often difficult to process.

3) Normalizing the Data

Normalization in NLP is the process of converting a word to its canonical form. Normalization is a little more complex than tokenization. Words have different forms—for instance, “watched”, “watches”,

and “watching” are various forms of the same verb, “watch” can all be normalized into “watch.” There are two major techniques of Normalization: Stemming, Lemmatization. Stemming is the process of removing a part of a word, or reducing a word to its stem or root. Stemming, working with only simple verb forms, is a heuristic process that removes the ends of words. so it will fail to notice the relationship between “feel” and “felt.”. Lemmatization normalizes a word with the context of vocabulary and morphological analysis of words in text. In linguistics, morphology is the study of words, how they are formed, and their relationship to other words in the same language .It analyzes the structure of words and parts of words such as stems, root words, prefixes, and suffixes. The lemmatization algorithm analyzes the structure of the word and its context to convert it to a normalized form. Therefore, it comes at a cost of speed. Lemmatization is more complex and needs a very high degree of knowledge of a language. It is used to determine the base word. It is a lexical database for the English language.

Table II: Tokenization and Part-of-Speech Tag

Tokenization	Part-of-Speech Tag
['#FollowFriday',	[['#FollowFriday', 'JJ'],
@France_Inte',	['@France_Inte', 'NNP'],
@PKuchly57',	['@PKuchly57', 'NNP'],
@Milipol_Paris',	['@Milipol_Paris', 'NNP'],
'for',	['for', 'IN'],
'being',	['being', 'VBG'],
'top',	['top', 'JJ'],
'engaged',	['engaged', 'VBN'],
'members',	['members', 'NNS'],
'in',	['in', 'IN'],
'my',	['my', 'PRP\$'],
'community',	['community', 'NN'],
'this',	['this', 'DT'],
'week',	['week', 'NN'],
':)']	[':)', 'NN']

averaged_perceptron_tagger is used to determine the context of a word in a sentence.

pos_tag function is required to determine the context for each word in the text. The verb being changes to its root form, be, and the noun members changes to member.

Table III: Stemming and Stop word Removal

Stemming	Stopword Removal
['#FollowFriday',	#followfriday',
@France_Inte',	'
@PKuchly57',	
@Milipol_Paris',	
'for',	
'be',	
'top',	top',
'engage',	'engage',
'member',	'member',
'in',	
'my',	
'community',	'community',
'this',	
'week',	'week', ':)
':)']	

C. Building Frequency Dictionary and TF-IDF Feature Extraction

The most basic form of analysis on textual data is to find out the word frequency. A single tweet is too small of an entity to find out the distribution of words, hence, the analysis of the frequency of words would be done on all tweets. Freqdist is used to find the frequent count of the words in the tweets

TF-IDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf-idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf-idf. In the bag of words approach, each word has the same weight. The idea behind the TF-IDF approach is that the words that occur less in all the documents and more in individual document contribute more towards classification. TF-IDF is a combination of two terms. Term frequency (TF) and Inverse Document frequency (IDF). They can be calculated as:

Table IV: Calculation of TF-IDF

<p>• Example= "It is hot outside"</p> <p>TF = (Frequency of a word in the document)/(Total words in the document)</p> <p>TF for word outside=1/4=0.25</p> <p>IDF = Log((Total number of docs)/(Number of docs containing the word))</p> <p>=Log(3/2)</p> <p>=0.176</p> <p>The TF-IDF value will be 0.25 x 0.176 = 0.044.</p>

Sentiment analysis is a process of identifying an attitude of the author on a topic that is being written about. To train a model, training data set is created. Classifiers are trained on training data sets with the features obtained from the feature extraction techniques. It is a supervised learning machine learning process, which needs to associate each dataset with a "sentiment" for training. "Positive" and "Negative" sentiments are used. A model is a description of a system using rules and equations. It may be as simple as an equation which predicts the weight of a person, given their height. Build A sentiment analysis model would associate tweets with a positive or a negative sentiment. Split the dataset into two parts. The purpose of the first part is to build the model, whereas the next part tests the performance of the model.

In the data preparation step, prepare the data for sentiment analysis by converting tokens to the

dictionary form and then split the data for training and testing purposes.

D. Converting Tokens to a Dictionary

Prepare the data to be fed into the model. Naive Bayes machine learning classifier in NLTK to perform the modeling. The model requires not just a list of words in a tweet, but a Python dictionary with words as keys and True as values.

E. Machine Learning Algorithm

Naïve Bayes' algorithm is a probabilistic classifier. Which is used for classification of sentiments? This machine learning algorithm is based on Bayes' theorem. This classifier is very effective when size of data set is large and broadly used because of its simplicity. A Naïve Bayes classifier composed two components: quantitative and qualitative. The quantitative components of Naïve Bayes classifier can be shown in network parameters form called as conditional table while the qualitative components of Naïve Bayes classifier can be shown in network structure form [9]. The probability equation is given below

$$P(B_j|A) = \frac{P(B_j)P(A|B_j)}{P(A)} \quad (1)$$

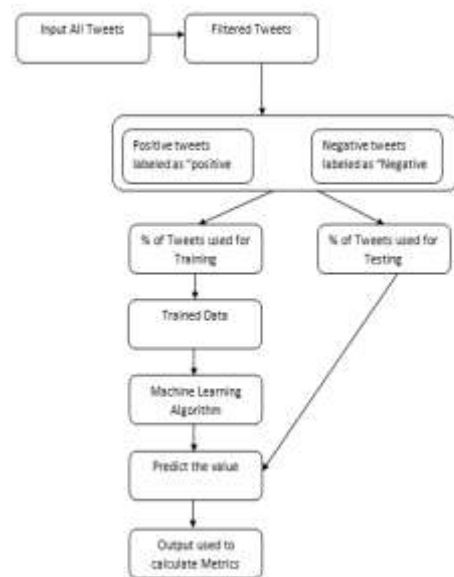


Fig.3: Classification Process

In this research, model of multinomial Naïve Bayes is used since it assumed the mutual independence of each word for all classes. The multinomial naive bayes equation is given below

$$C_{MAP} = \underset{c_i \in C}{\operatorname{argmax}} P(c_i) \prod_{j=1}^{j=n} P(x_j|c_i) \quad (2)$$

Based on the results of the pre-processing, the probability of occurrence of each class on the sample (prior probabilities) and the per-word weighting chances to be classified into positive classes and negative classes (conditional probabilities) is calculated.

The number of terms generated from the pre-process will be used on the NBC algorithms. Following are the words which occur most frequently in the twitter data set.

Table V: Most Frequent words in Twitter Data Set

Words	Frequency
:)	3691
:-)	701
:d	658
thanks	388
follow	357
love	333
good	283
get	263
thank	253

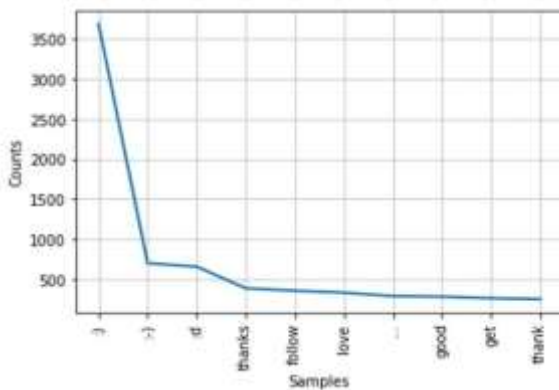


Figure 4: Graph representing most frequent words

Since it is a supervised learning task so it requires to associate each data set with a sentiment like "Positive and negative" or labeled with "1" or "0" and a test data set without labels. Split the data set in two parts i.e. Training set and Test Set. Now, the frequency of some sample words is calculated in positive and negative classes as shown in table VI.

Table VI: Word Frequency of each class

Word	Negative	Positive
:(2356.7	1.0
:)	1.0	1126.5
follower	1.0	41.1
bam	1.0	25.1
sad	22.7	1.0
arrive	1.0	21.1
x15	18.9	1.0
aw	15.6	1.0
glad	1.0	14.2
didnt	13.0	1.0

In the table that shows the most informative features, every row in the output shows the ratio of occurrence of a token in positive and negative tagged tweets in the training dataset. The first row in the data signifies that in all tweets containing the token :(, the ratio of negative to positives tweets was 2356.7 to 1. Interestingly, it seems that there was one token with :(in the positive datasets. The top two discriminating items in the text are the emoticons.

Further, words such as sad lead to negative sentiments, whereas welcome and glad are associated with positive sentiments. The class in which frequency of the word is more, then the word is to be categorized in that class only is given in table.

Input tweet is tested against the trained model. For input tweet 1, word like "nice" is having more frequency in positive category. So it is labelled as positive sentiment. For input tweet 2, word like "best" is having more frequency in positive category. So it is labelled as positive. For input tweet 3, word like "worst" is having more frequency in negative category. So it is labelled as negative

Table VII: Validation of Results (Test the Data)

Input Tweet 1: "I ordered just once from TerribleCo, they were nice, i will use again."
Output Label: Positive
Input Tweet 2: 'Congrats #SportStar on your 7th best goal from last sea son winning goal of the year :) #Baller #Topbin #oneof manyworldies'
Output Label: Positive
Input Tweet 3: Trains service was worst
Output Label: Negative

IV. RESULTS AND DISCUSSIONS

There are 10000 tweets used in the analysis. As we have already discussed tweets are collected via twitter API. Only English language tweets are considered. NLTK python library is used to pre-process and analyse the tweets. nltk.download ('twitter samples') package from python is used to download the samples. Corpus is used to include the samples. Then data have gone through the cleaning and pre-processing process. After pre-processing, data is divided into two parts, namely training data and testing data. The training data and testing data used are formulated using 70:30 rule. Therefore, by using the functionality of python to train the data set and analyse sentiment from data to measure its accuracy, building classifiers performing experiments and data visualization. Test parameters used for evaluation are accuracy, precision and recall whose calculations are obtained from the confusion

matrix table. Confusion matrix is a performance measurement for machine learning classification

where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values and gives us a matrix as output and describes the complete performance of the model. The calculation of accuracy, precision, and recall and F-measure is obtained through the formula [18].

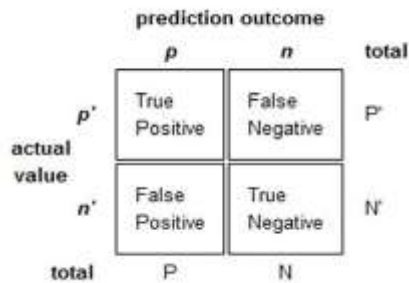


Figure 5: Confusion Matrix

There are 4 important terms in confusion metric:

- True Positive:** The cases in which we predicted YES, and the actual output was also YES.
- True Negatives:** The cases in which we predicted NO, and the actual output was NO.
- False Positives:** The cases in which we predicted YES, and the actual output was NO.
- False Negatives:** The cases in which we predicted NO, and the actual output was YES.

• **Accuracy**

Classification accuracy is the total number of correct predictions divided by the total number of predictions made for a dataset.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (3)$$

• **Precision**

Precision quantifies the number of positive class predictions that actually belong to the positive class.

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

• **Recall**

It quantifies the number of positive class predictions made out of all positive examples in the dataset.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

• **F-Measure**

It provides a single score that balances both the concerns of precision and recall in one number

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (6)$$

This data has been collected from Indian Railway Seva on twitter. During the analysis results obtained are really interesting. One sample of result analysis with multinomial naïve bayes is discussed here. In Multinomial Naïve Bayes 1159 tweets are categorized as true positive. 340 tweets are categorized as false negative. 408 tweets are in False positive category and 1093 tweets are in True Negative category. Apply above mentioned formula for accuracy, precision, recall and F measure on these results.

Table VIII: Analysis using Multinomial Naive Bayes Classifier

```

Multinomial Naive Bayes Accuracy Score ->
75.033

Confusion Matrix
[[1159 340]
 [ 408 1093]]

precision recall f1-score support
0 0.74 0.77 0.75 1499
1 0.76 0.73 0.74 1501

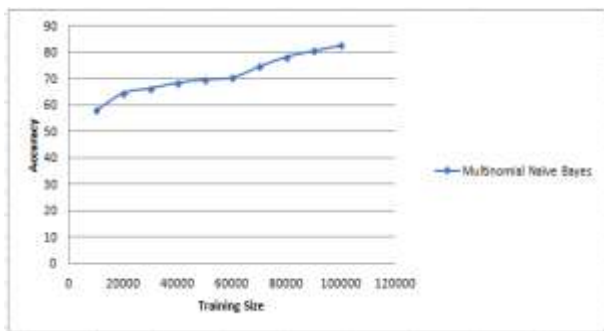
accuracy          0.75 3000
macro avg 0.75 0.75 0.75 3000
weighted avg 0.75 0.75 0.75 3000
    
```



Figure 6: Graph Representing Different results obtained for Multinomial Naïve Bayes Algorithm

Table IX: Accuracy, precision and recall value with Multinomial NBC Corpus Data set

Evaluation Parameters		Algorithms
		Multinomial Naïve Bayes
Accuracy		82.6%
Precision	Positive	81.57%
	Negative	83.57%
Recall	Positive	84.6%
	Negative	80.6%
F1 Score	Positive	83.05%
	Negative	82.05%

**Figure 7: Graph Representing Different results obtained for Multinomial Naïve Bayes Algorithm on Corpus Data set**

V. CONCLUSION

This paper makes contribution in data science and natural language processing fields. Experiments have been done on railway service by developing the model which performs sentiment analysis on twitter data using machine learning techniques. Natural Language Tool Kit (NLTK) is used to prepare the model on the dataset containing tweets. TF-IDF concept is used for feature extraction from preprocessed data. Multinomial Naïve Bayes Classifier is used for classification of twitter sentiments by calculating the probability of new input data and the tweet with the largest value is taken in to account as either positive or negative. Experiments have been done on two different kinds of the data set .The performance of the model is checked by calculating the accuracy as evaluation metric.

REFERENCES

- Population in India: <http://www.worldometers.info/world-population/population-by-country/> Accessed 23 July 2018
- M. B. Myneni., L. V. N. Prasad and J. S. Devi (2017). "A Framework for Semantic Level Social Sentiment Analysis Model", JATIT. vol. 96, no. 16, pp. 1992-8645
- Pak, A. & Paroubek, P. (2010). "Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In Proceedings of the Seventh Conference on International Language Resources and Evaluation, pp. 1320–1326.
- Abdelwahab, O., Bahgat, M., Lowrance, C. J., & Elmaghraby, A. (2015). Effect of Training Set Size on SVM and Naïve Bayes for Twitter Sentiment Analysis. IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 46–51
- R. Parikh and M. Movassate (2009). "Sentiment Analysis of User- Generated Twitter Updates using Various Classification Techniques", CS224N Final Report.
- Go, R. Bhayani, L. Huang (2009). "Twitter Sentiment Classification Using Distant Supervision". Stanford University, Technical Paper.
- Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau (2011). "Sentiment Analysis of Twitter Data", In Proceedings of the ACL 2011, Workshop on Languages in Social Media, pp. 30-38
- Dmitry Davidov (2010). Ari Rappoport." Enhanced Sentiment Learning Using Twitter Hashtags and Smileys". Coling 2010: Poster Volume pages 241{249, Beijing, August 2010
- Nazim Razali1, Aida Mustapha, Faiz Ahmad Yatim, Ruhaya Ab Aziz (2017). "Predicting Football Matches Results using Bayesian Networks for English Premier League (EPL)" International Research and Innovation Summit (IRIS 2017).
- Po-Wei Liang, Bi-Ru Dai (2013). "Opinion Mining on Social Media Data", IEEE 14th International Conference on Mobile Data Management, Milan, Italy, June 3 - 6, 2013, pp 91-96, ISBN: 978-1-494673-6068-5, <http://doi.ieeecomputersociety.org/10.1109/MDM.2013>.
- Pablo Gamallo, Marcos Garcia (2014). "Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets", 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland, Aug 23-24 2014, pp. 171-175.
- Turney, P. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In Proceedings of Annual Meeting of the

Association for Computational Linguistics (ACL2002).

13. R. Xia, C. Zong, and S. Li (2011). "Ensemble of feature sets and classification algorithms for sentiment classification," *Information Sciences: an International Journal*, vol. 181, no. 6, pp. 1138–1152.
14. Zhunchen Luo, Miles Osborne, Ting Wang (2013). "An effective approach to tweets opinion retrieval", *Springer Journal on World Wide Web*, DOI: 10.1007/s11280-013-0268-7.
15. Aisopos, Fotis, et. al. (2012). "Content vs. context for sentiment analysis: a comparative analysis over microblogs." *Proceedings of the 23rd ACM conference on Hypertext and social media*. ACM, 2012.
16. Jebaseeli, A. Nisha, and E. Kirubakaran (2012). "A Survey on Sentiment Analysis of (Product) Reviews." *International Journal of Computer Applications* 47.11.
17. Routray, P., Swain, C. K., & Mishra, S. P. (2013). A survey on sentiment analysis. *International Journal Of Computer Applications*, pp. 2-4.
18. Routray, P., Swain, C. K., & Mishra, S. P. (2013). A survey on sentiment analysis. *International Journal of Computer Applications*, 2-4.

Corresponding Author

Reshma Gulwani*

Research Scholar, Sunrise University, Alwar
Rajasthan, India