International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-10 Issue-8, June 2021

Evaluation of Supervised Classification Techniques on Twitter Data using R

Annie Syrien, M. Hanumanthappa, Ravi Kumar K

Abstract*: The phenomenal development of the World Wide Web has resulted in enormous social networking sites producing tremendous data on web 2.0. Social networking sites have widened to a higher degree of use, in which any field of information can be sort by researchers. Data obtained from social media has strategized from many new machine learning algorithms and natural language processing. The data is unstructured; mining the data leads to finding important sentiments about various entities via appropriate classification techniques. In this paper, tweets' opinions are analyzed through machine learning algorithms such as naive Bayes and support vector machines using R programming; results are computed and compared. The SVM model manifests the higher precision, and naïve Bayes provides higher accuracy for sentiment analysis on the Bengaluru traffic data.

***Keywords*: Machine Learning Algorithms, Sentiment Classification, Text Mining, Twitter.**

I. INTRODUCTION

Twitter is the most sought after microblogging site in which users post their messages, which produces enormous and enterprising datasets yielding useful information [1]. At present, we live in an era of information where information is appraised as wealth. The data is extracted and categorized; unsheathing this potentially useful information is a trend nowadays from large giants to amateur for better prospects and opulence. Text classification is the prime mechanism used for organizing texts for many years [2]. In the past few years, many approaches have been used to envision user opinions on different entities [3]. However, the accuracy of envisioning user opinions is dependent on the accuracy of sentiment classification algorithms. Thus evaluating and comparing these classification algorithms contribute to the useful result on text classification on tweets.

Sentiment analysis and opinion mining has been a broad study of interest these days [4]. In view of knowing the opinions and sentiments of the netizens due to easy access to information. It is one of the easiest ways of expressing opinions to the world, whereas everyone has a platform to say what they perceive and comprehend [5].

In Twitter sentiment analysis, tweets are perused for categorizing positive, negative, or neutral [6] to evaluate one's opinions toward a particular product or topic. Hence Twitter data analytics is the most sought research field. It also poses many challenges for the researchers, as there lacks accountability for the works carried out. Our objective is to work on three different machine learning algorithms on tweets and propose the algorithm which works better on Bengaluru traffic tweets.

This paper is structured and follows as related work, methodology, results, and discussion, conclusion. Related work has shown clear evidence on data mining algorithms yielding to the good results in Twitter data with discussions on various researchers proposed techniques and machine learning methodologies, methodology, this domain discusses the accuracy of approaches arrogated to prognosticate traffic emotions in Bengaluru through the implementation of naive Bayes and support vector machine classification techniques, results and discussion give the output of these results and comparison between the different classification methods on data set using r packages, finally concludes with the scope of the research work amalgamating the challenges the comparison experiment done

II. RELATED WORK

Related work has shown clear evidence on data mining algorithms yielding to the good results in Twitter data. Text mining is a mechanism of procuring qualitative text information. It uses statistical-based techniques. Text mining involved data collection, noise removal, feature selection, training, and testing the model. Few similar works are briefed here. Geetika Gautam at al [7] worked on sentiment analysis of tweets through machine learning approaches, they extracted customer review tweets, preprocessed, and used supervised techniques such as naive bayes, SVM, and maximum entropy along with the inclination of semantics from wordnet and extracted the synonyms and similarity for the processed tweets. They also measured the precision, recall, and accuracy of each classification algorithm. They concluded that naive bayes produce better results than SVM and maximum entropy when it is a unigram model.

Yun Wan et al. [8] developed an ensemble sentiment classification system, which excerpts the majority vote principle of conglomerate classification methods such as Bayesian Network, Naive Bayes, C4.5, SVM, Random Forest algorithm, and Decision Tree.

Manuscript received on June 20, 2021.
Revised Manuscript received on July 02, 2021.
Manuscript published on June 30, 2021.
* Correspondence Author

Annie Syrien*, Assistant Professor, Department of Computer Science and Applications, Bangalore University, (Bengaluru), India. Email: syrien01@gmail.com

Hanumanthappa M, Professor, Department of Computer Science and Applications, Bangalore University, (Bengaluru), India Email: hanu6572@gmail.com

Ravi Kumar K, Scholar, Kalinga Institute of Industrial Management (KIIT), Bhubaneswar, (Odisha), India.

Retrieval Number: 100.1/ijitee.H92660610821
DOI: 10.35940/ijitee.H9266.0610821
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication
© Copyright: All rights reserved.

They have used tenfold evaluation to endorse the classifiers and concluded that high accuracy results are yielded for the airline services domain with their proposed work.

Alexander Pak et al. [9] contributed on Twitter as a Corpus for Sentiment Analysis and Opinion Mining; they have presented a medium for mechanized corpus collection. They cast offed tree tagger for POS tagging for positive, negative, and neutral classifications. Along with multinomial naive bayes classifier that employed N gram and POS tags as features. Finally concluded evaluations that exhibit their recommended techniques are better and efficient.

Akshi Kumar et al. [10] expounded on the compound method by combining corpus and dictionary-based methods to evaluate the tweets' sentiments. In the corpus-based method, semantics based on adjectives were assessed, whereas in the dictionary-based method, verbs and adverbs were used for evaluation. Their research paper titled sentiment analysis on Twitter illustrated a case study on the compound method proposed and concluded it as a motivating technique.

Ali Hasan et al. [11] presented the work on machine learning based sentiment analysis for twitter accounts, they proposed a hybrid technique with sentiment analyzer, with comparison of political views via Naive Bayes and Support vector machines along with sentiment lexicons. They analyzed the sentiments through dictionary based methodology and compared the results with two popular machine learning algorithms, with conclusions such as sentiment lexicons yielded better results. Their future proposed work is to find the patterns based on Twitter reviews.

Grant Willians et al. [12] has worked on mining twitter data for a more responsive software engineering process, they have collected data from Twitter feeds of three different software sources such as Minecraft, Snapchat and WhatsApp. The data was collected for three months, 51,792 tweets were collected. 400 tweets were sampled arbitrarily, then manually classified into three main types such as feature requests, bug reports, and others. followed by automatic classification via NB and SVM text classifiers were carried out. The results manifest NB and SVM are very tacit in disclosing technical tweets

III. METHODOLOGY

In our research work, we have used naive bayes and support vector machine classification to determine the positive, negative, and neutral sentiments of the tweets on Bengaluru traffic. The analysis of traffic data was done on the real time Twitter retrieved data. In machine learning, supervised techniques classify tweets in a three-way classification of emotions such as positive, negative, and neutral. Classification is a machine learning supervised technique for predicting a model based on input attributes. Fig. 1 shows the stages of classification. Classification is carried out in two juncture, stage one is training data, and stage two is testing data. For sentiment analysis, two sets of data are required, training data and testing data [13].

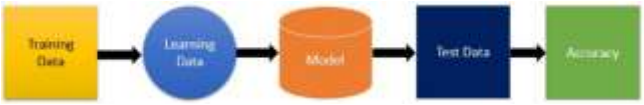


Fig. 1: Stages of Classification.

The classification process involves data retrieval, preprocessing data such as removing noise, detaching user mentions, special characters, digits, retweets, duplicates, hashtags, emoticons, and URL's, stop words, and so on. Sentiment identification involves assigning positive, negative, and neutral scores ranging from +5 to +1, -6 to -1, and 0. Fig. 3 illustrates the sentiment identification of tweets. Corpus is created, data is segregated as training and testing set, the feature is selected, feature selection is the pith of machine learning algorithm, what type of feature is selected will affect the model accuracy, then training the data through machine learning classification algorithms, functioning the model, testing, predicting the sentiment as shown in Fig. 2.

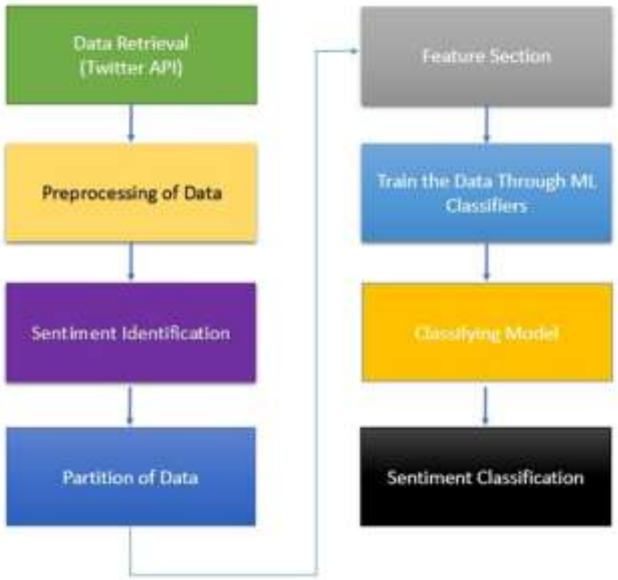


Fig. 2: Sentiment Classification

A. DATA

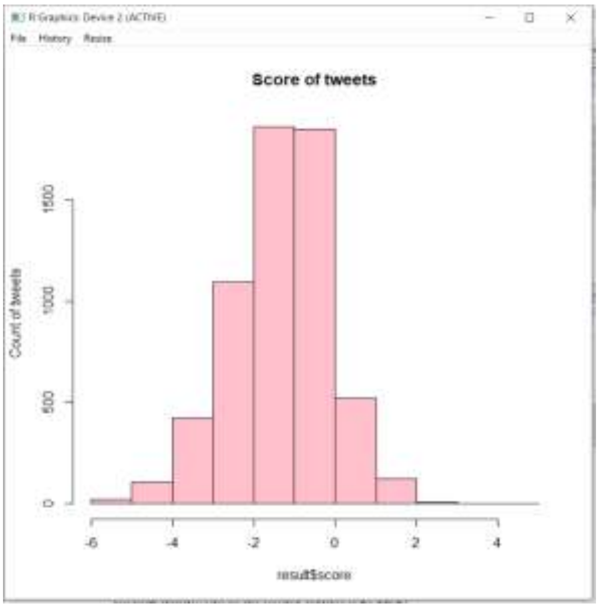
The data set used is obtained from Twitter streaming API. Data is selected on Bengaluru traffic. The tweets collected from Twitter API were 11,689 after the preprocessing stage retrieved data was 6013, out of which 655 was positive, 3506 negative, 1852 neutral, as shown in table 2. Few positive words samples are ‘good’, ‘clear’, ‘fast’, ‘early’, ‘smooth’, ‘safe’, ‘less traffic’. Few negative words samples are ‘traffic’, ‘waiting’, ‘bad’, ‘road’, ‘time’, ‘lost’, ‘problem’, ‘long’, ‘block’, ‘jam’, ‘pain’. Table 1 shows the sample tweets of positive, negative, and neutral tweets. The data is divided as 75% of data as training data, remaining 25% as test data.

Table 1: Example Tweet

Sentiment	Tweet
Positive	There is hope for motorists as BBMP begins preliminary works for the underpass at the HMT Road Pipeline Road
Negative	Hey Bangalore Thanks to an evil driver and your crappy traffic I have missed my flight Well done
Neutral	Fantastic experience with a startup in Bangalore called it’s basically a concierge service

Table 2: Class distribution

Class	Positive	Negative	Neutral
Tweets	655	3506	1852



B. NAIVE BAYES

Naive bayes classifier is a simple and most effective classification algorithm [14]. Naive Bayes Classifier assembles the facets in the feature vector and exams the features independently. Feature selection is about the vital text; to evaluate the model, we have used the n-gram model. Where n=10, so feature vector with n size as ten is selected for classification. The conditional probability for Naive bayes defined as

P(doc|yj) = IIm i=1 P(di|yj)

'doc' is a document that consists of a feature vector of words defined as doc={d1,d2,....dm}, and yj is the class label. Naive Bayes classification relies on a bag of words hence does not consider the relationships between features

C. SVM

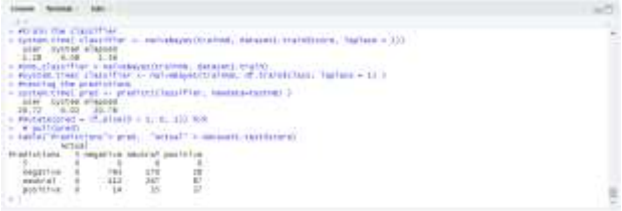
A support vector machine classifier is a linear classifier that's most suited for text classification due to the text's sparsity nature; that is, few words are irrelevant, hence separating classes linearly via hyper plane [15]. The linear kernel is used for classification. SVM uses a discriminative function i.e., defined as

g(doc) = wTØ(doc) + b

'doc' is the feature vector, 'w' is the weights vector, and 'b' is the bias vector. Ø is the nonlinear mapping from input space to high dimensional feature space [16]. 'w' and 'b' are learned automatically from the training set

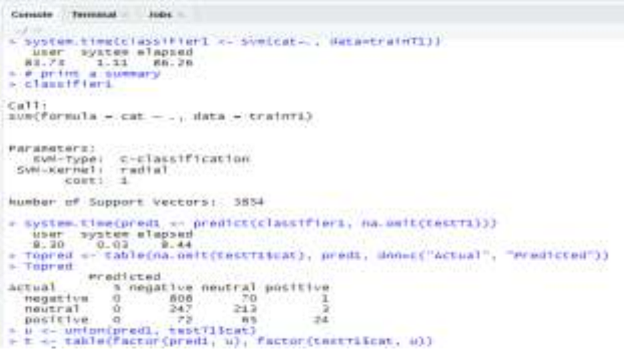
IV. RESULTS AND DISCUSSION

Fig. 3 shows the execution of the Naive Bayes Classifier. The system time function is used to evaluate the time taken by the system; the package e1071 is used to implement the naïve bayes classification algorithm; the following steps carry out the classification, first data is retrieved from Twitter API, preprocessed, tokenized, randomize function is used to fine-tune the data, sentiment identification of tweets are made via a lexicon-based method, data is given the sentiment as positive, negative and neutral, features selection was made, naive bayes classification was used to evaluate the tweets as positive, negative and neutral. The output is assessed with the original trained data, and results are computed, and a confusion matrix is used to find truly negative, truly positive, and truly neutral tweets.



```
> #train the classifier
> system.time( classifier <- naiveBayes(trainNB, dataset$trainStore, laplace = 1))
   user  system elapsed
   1.18    0.08    1.16
> #nmm_classifier = naiveBayes(trainNB, dataset$train)
> #system.time( classifier <- naiveBayes(trainNB, df.train$class, laplace = 1) )
> #testing the predictions
> system.time( pred <- predict(classifier, newdata=testNB) )
   user  system elapsed
  28.72    0.02   28.76
> #mytablecored = [illegible]
>  # pull(pred)
> table("Predictions"= pred,  "Actual" = dataset$testStore)
           Actual
Predictions   $ negative neutral positive
   $          0        0       0        0
   negative   0      781     178       28
   neutral    0      412     247       87
   positive   0       14      35       27
> 
```

Fig. 3: Naive Bayes Classifier Execution in R



```
> system.time(classifier1 <- svm(cat~., data=trainT1))
   user  system elapsed
  83.73    1.11   86.26
> # print a summary
> classifier1

Call:
svm(formula = cat ~ ., data = trainT1)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  3854

> system.time(pred1 <- predict(classifier1, na.omit(testT1)))
   user  system elapsed
   8.30    0.03    8.44
> Topred <- table(na.omit(testT1$cat), pred1, dnn=c("Actual", "Predicted"))
> Topred
          Predicted
Actual       $ negative neutral positive
  negative   0      808      70        1
  neutral    0      247     213        2
  positive   0       72      85       24
> u <- union(pred1, testT1$cat)
> t <- table(factor(pred1, u), factor(testT1$cat, u))
```

Fig. 4: SVMs Classifier Execution in R

Figure 5 shows the word cloud as a representation of data. A word cloud is a visualization tool used to find frequently occurring words. The minimum frequency is assigned as 40 in the below figure negative and traffic words, which is occurred to a large extent identifying itself as the most frequently occurring words. The positive words look very small self-describing itself as a less regularly occurring word.

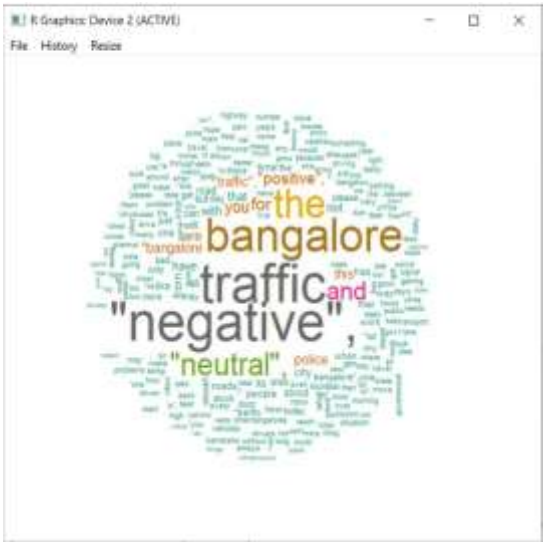


Fig. 5: Word cloud of Dataset

The Confusion Matrix of the Naive bayes classifier is represented by Fig. 5. The confusion matrix of SVMs is represented in Fig. 6.

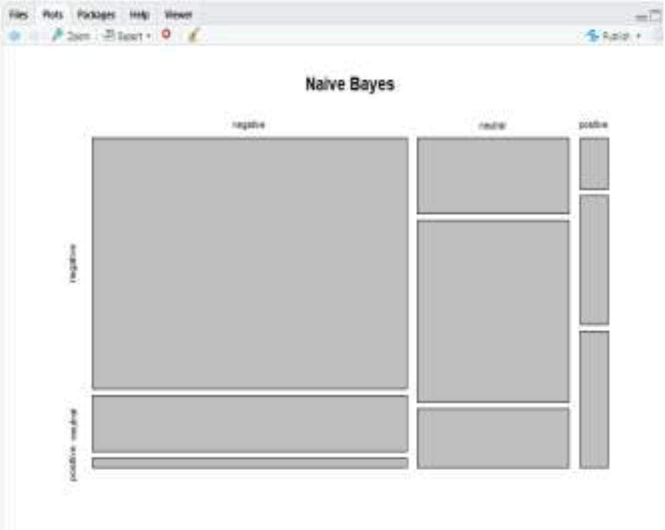


Fig. 6: Confusion Matrix for Naive Bayes

A. Performance Matric

Accuracy and F-Score: Accuracy measures performance via ratio of classification by total responses, whereas the f-score measures the performance through the harmonic mean of the precision and recall.

Precision: Precision is measured as a percentage of predicted labels that are correct [17].

P = (TP/((Tp+FP)))

Recall: Recall is measured as the percentage of correct tweets that are selected.

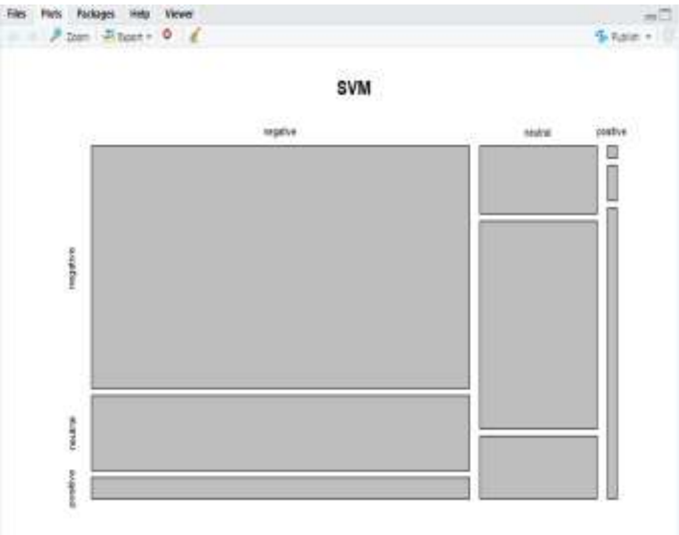
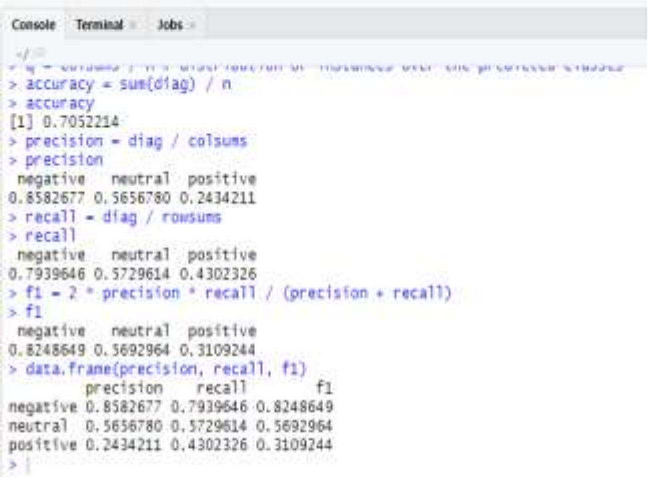


Fig. 7: Confusion Matrix for SVMs

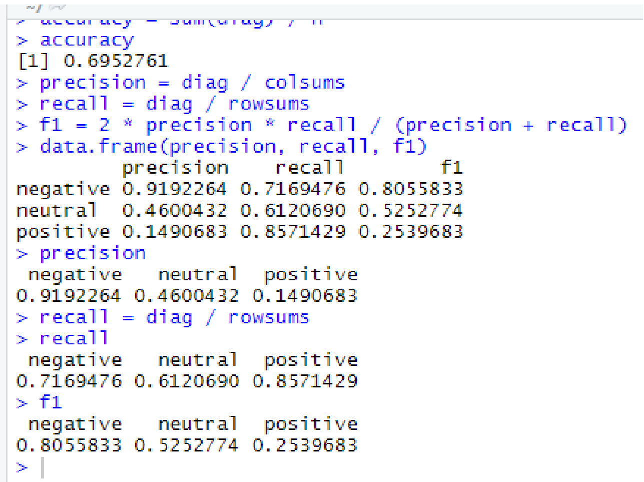
R = (TP/((Tp+FN)))

Fig. 7 shows the accuracy, precision, recall, and F-score of the naive bayes classification technique. Fig. 8 shows the accuracy, precision, recall, and F-score of the SVMs classification technique. It's clearly evident Naive bayes gives more accuracy than SVM, and SVM has a higher precision value.



```
> accuracy = sum(diag) / n
> accuracy
[1] 0.7052214
> precision = diag / colsums
> precision
 negative   neutral  positive
0.8582677 0.5656780 0.2434211
> recall = diag / rowsums
> recall
 negative   neutral  positive
0.7939646 0.5729614 0.4302326
> f1 = 2 * precision * recall / (precision + recall)
> f1
 negative   neutral  positive
0.8248649 0.5692964 0.3109244
> data.frame(precision, recall, f1)
         precision    recall        f1
negative 0.8582677 0.7939646 0.8248649
neutral  0.5656780 0.5729614 0.5692964
positive 0.2434211 0.4302326 0.3109244
> |
```

Fig. 7: Performance metric of Naive Bayes



```
> accuracy = sum(diag) / n
> accuracy
[1] 0.6952761
> precision = diag / colsums
> recall = diag / rowsums
> f1 = 2 * precision * recall / (precision + recall)
> data.frame(precision, recall, f1)
         precision    recall        f1
negative 0.9192264 0.7169476 0.8055833
neutral  0.4600432 0.6120690 0.5252774
positive 0.1490683 0.8571429 0.2539683
> precision
 negative   neutral  positive
0.9192264 0.4600432 0.1490683
> recall = diag / rowsums
> recall
 negative   neutral  positive
0.7169476 0.6120690 0.8571429
> f1
 negative   neutral  positive
0.8055833 0.5252774 0.2539683
> |
```

Fig. 8: Performance metric of SVM

V. CONCLUSION

The article provides an overview of Twitter data analysis through machine learning algorithms such as naive bayes and support vector machines. Also throws light on acquisition of different sentiment analyzers to analyze the highest accuracy, precision, recall, and F-score for Bengaluru traffic Twitter data. The user tweets visualized through word clouds. Both SVM and Naive bayes are models are computed. The SVM model manifests the higher precision, and naive bayes provides higher accuracy for sentiment analysis. The classification techniques show that many commuters are unhappy with Bengaluru city traffic. R serves as an effective tool for text classification of data collected.

REFERENCES

1. Nadia F. F. Da Silva, Eduardo R. Hruschka, and Estevam R. Hruschka Jr, “Tweet sentiment analysis with classifier ensembles,” Decision Support Systems, vol. 66, pp. 170-179, 2014.
2. Rui Xia, Chengqing Zong, and Shoushan Li, “Ensemble of feature sets and classification algorithms for sentiment classification,” Information Sciences, vol. 181, pp. 1138-1152, 2011.
3. Gang Wang, Jianshan Sun, Jian Ma, Kaiquan Xu, and Jibao Gu, “Sentiment classification: The contribution of ensemble learning,” Decision Support Systems, vol. 57, pp. 77-93, Jan. 2014.

4. M. S. Neethu and R. Rajasree, "Sentiment analysis in Twitter using machine learning techniques," in 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1-5.
5. Neelam Duhan Ashima Garg, "Sarcasm Detection On Twitter Data Using Support Vector Machine," Ictact Journal On Soft Computing, vol. VOLUME: 10, no. ISSUE: 04, pp. 2165-2170, July 2020.
6. Pablo Gamallo and Marcos Garcia, "Citius: A naive-bayes strategy for sentiment analysis on English tweets," in Proceedings of the 8th international Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 171-175.
7. Geetika Gautam and Divakar Yadav, "Sentiment analysis of Twitter data using machine learning approaches and semantic analysis," in 2014 Seventh International Conference on Contemporary Computing (IC3), 2014, pp. 437-442.
8. Yun Wan and Qigang Gao, "An ensemble sentiment classification system of Twitter data for airline services analysis," in 2015 IEEE international conference on data mining workshop (ICDMW), 2015, pp. 1318-1325.
9. Alexander Pak and Patrick Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining" in LREc, vol. 10, 2010, pp. 1320-1326.
10. Akshi Kumar and Teeja Mary Sebastian, "Sentiment analysis on Twitter," International Journal of Computer Science Issues (IJCSI), vol. 9, p. 372, 2012.
11. Sana Moin, Ahmad Karim and Shahaboddin Shamshirband Ali Hasan, "Machine Learning-Based Sentiment Analysis for Twitter Accounts," Mathematical and Computational Applications, vol. 23, no. 11, pp. 1-15, February 2018.
12. Anas Mahmoud Grant Williams, "Mining Twitter Data for a More Responsive," in 2017 IEEE/ACM 39th IEEE International Conference on Software Engineering Companion, Buenos Aires, 2017.
13. A. Vishal and S. S. Sonawane, "Sentiment Analysis of Twitter Data: A Survey of Techniques," International Journal of Computer Applications, vol. 139, pp. 5-15, Apr. 2016.
14. Akshay Amolik, Niketan Jivane, Mahavir Bhandari, and M. Venkatesan, "Twitter sentiment analysis of movie reviews using machine learning techniques," international Journal of Engineering and Technology, vol. 7, pp. 1-7, 2016.
15. Walaa Medhat, Ahmed Hassan, and Hoda Korashy, "Sentiment analysis algorithms and applications: A survey," Ain Shams engineering journal, vol. 5, pp. 1093-1113, 2014.
16. Nabizath Saleena and others, "An Ensemble Classification System for Twitter Sentiment Analysis," Procedia computer science, vol. 132, pp. 937-946, 2018.
17. Farhan Hassan Khan, Saba Bashir, and Usman Qamar, "TOM: Twitter opinion mining framework using hybrid classification scheme," Decision Support Systems, vol. 57, pp. 245-257, 2014.

AUTHORS PROFILE

.

M. Hanumanthappa, Professor, Department of Computer Science and Applications, Bangalore University, Bengaluru, India. His Research Interests are Data Mining, Machine Learning, Network Security, NLP

Annie Syrien, Assistant Professor, St.Joseph's Evening College, Research scholar, Department of Computer Science and Applications, Bangalore University, Bengaluru, India.

Mr. Ravikumar K, is a Ph.D. Scholar in Kalinga Institute of Industrial Technology (KIIT), Odisha. He is working at National Assessment and Accreditation Council as a professional assistant since 2003 in the administrative and academic departments. Mr. Ravikumar is a Computer Science graduate from Bangalore University and he is holding a master degree in Sociology from Kuvempu University, Shimoga, Karnataka. Presently he is supporting the Director, NAAC in various academic and administrative activities at his secretariat. He has authored several papers on higher education in many journals.