ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.12, 10195–10209
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.12.12 Spec. Iss. on AMABDA-2020

A COMPARATIVE ANALYSIS OF VARIOUS MACHINE LEARNING BASED SOCIAL MEDIA SENTIMENT ANALYSIS AND OPINION MINING APPROACHES

K. JAYAMALINI[1], M. PONNAVAIKKO, AND JAYAMALINI KOTHANDAN

ABSTRACT. Sentiment Analysis or opinion mining is the method of NLP to computationally identify and categorize user opinions expressed in textual data. Mainly it is used to determine the user's opinions, emotions, appraisals, or judgments towards a specific event, topic, product, etc. is positive, negative, or neutral. In this study, a huge amount of digital data generated online from blogs and social media websites is gathered and analyzed to discover the insights and help in taking business decisions. Because of advancements in machine learning approaches, these approaches are still popular to analyze the huge volume of data. In this study, a huge amount of data is collected from Twitter were analyzed using machine learning approaches like SVM, Naive Bayes, logistic regression, random forest and compared with their performance metrics. This paper also explains various enriched pre-processing and Feature extraction techniques. At the end of this paper, the accuracy obtained using different machine learning based text classifiers are compared and an optimal solution for the problem of sentiment analysis is provided.

[1]*corresponding author*

2020 *Mathematics Subject Classification.* 65D05, 65D05, 65Z05.

Key words and phrases. Machine Learning, Naive Bayes, Logistic Regression, Random Forest, SVM, Emotion Analysis, NLP, Opinion Mining(OM), Sentiment Analysis (SA), Twitter.

1. INTRODUCTION

Social media is websites [1] and mobile-based applications that allow users to generate and share information about their ideas, interests, opinions, emotions via virtual networks and communities.

Sentiment Analysis or opinion mining [2] is the method of NLP to computationally identify and categorize user opinions expressed in textual data. Mainly it is used to determine the user's opinions, emotions, appraisals, or judgments towards a specific event, topic, product, etc. is positive, negative, or neutral. In this approach, a huge amount of digital data generated online from blogs and social media websites is gathered and analyzed to discover the insights and help in taking business decisions.

Social media facilitate the development of online social networks by connecting user's with other individuals or groups. Online social media data is pervasive. It allows people to post their opinions and sentiments about products, events, and other people in the form of short text messages. For example, Twitter is an online social networking service where users post and interact with short messages, called "tweets". Hence, currently, social media become a prospective source for businesses to discover people's sentiments and opinions about an event or product.

Sentiment Analysis based on twitter can be useful for a variety of tasks such as predicting stock markets, opinions of a product, political outcomes, and much more. This paper focuses on the development of various ML-based classifiers to classify the given tweet into either positive or negative. This paper focuses on the development of a Machine learning Based social media data emotion analyzer and sentiment classifier. This paper also explains various enriched techniques of preprocessing of Text data and Feature Extraction techniques using word embedding. The machine learning approaches used in this paper are SVM, Naive Bayes, logistic regression, and random forest.

2. IMPORTANCE OF SOCIAL MEDIA SENTIMENT ANALYSIS

Social Media data is analyzed to find insights and thoughts about customers and competitors. It is important for business and helps the business to:

- make fruitful social campaigns using marketing analytics
- recognize influencers for their business

- to find strengths and weaknesses of competitors using competitive intelligence
- discover the real-time trending topics about their brand
- to identify their customer opinions
- Keep track the virality of content spreads on the web

Twitter is one of the most widespread microblogging website [3], where users can post short textual messages called "tweets" or “microblogs.” Twitter is the third most popular social network in the U.S. Tweet is a short, 140-character message Twitter users broadcast to their contacts. Twitter is frequently used as a place to report, react to, and engage with topics of national and international importance. That’s what Twitter data is widely used by researchers to search for users’ sentiments and opinions.

3. The architecture of the Proposed System

ML-based Sentiment Analyzer is used to analyze the twitter data using SVM, Naive Bayes, logistic regression, and random forest approaches. It uses enriched pre-processing techniques and feature extraction techniques. This system has been used by businesses to enhance customer experience. The framework of the proposed system is shown in the figure below. It comprises of:

- Classified Tweets Loader
- Enriched Pre-processor
- Feature Extractor
- ML Based Emotion Classifiers
- Accuracy Comparator

- Tweets Loader: It is used to load Tweets from labeled Twitter data corpus.
- Enriched Text Cleaner and Pre-processor: It is used to convert the raw text into clean text by removing numeric values, non-English characters, URLs, white spaces and stop words. It also handles case sensitive issues of text and stemming process.
- Feature Extractor: Human Being has easily understood the meaning of text data. But machines simply cannot process the raw text data. In Natural Language Processing, the text data need to break down into a numerical format that is easily readable by the machine. Word Embedding is one

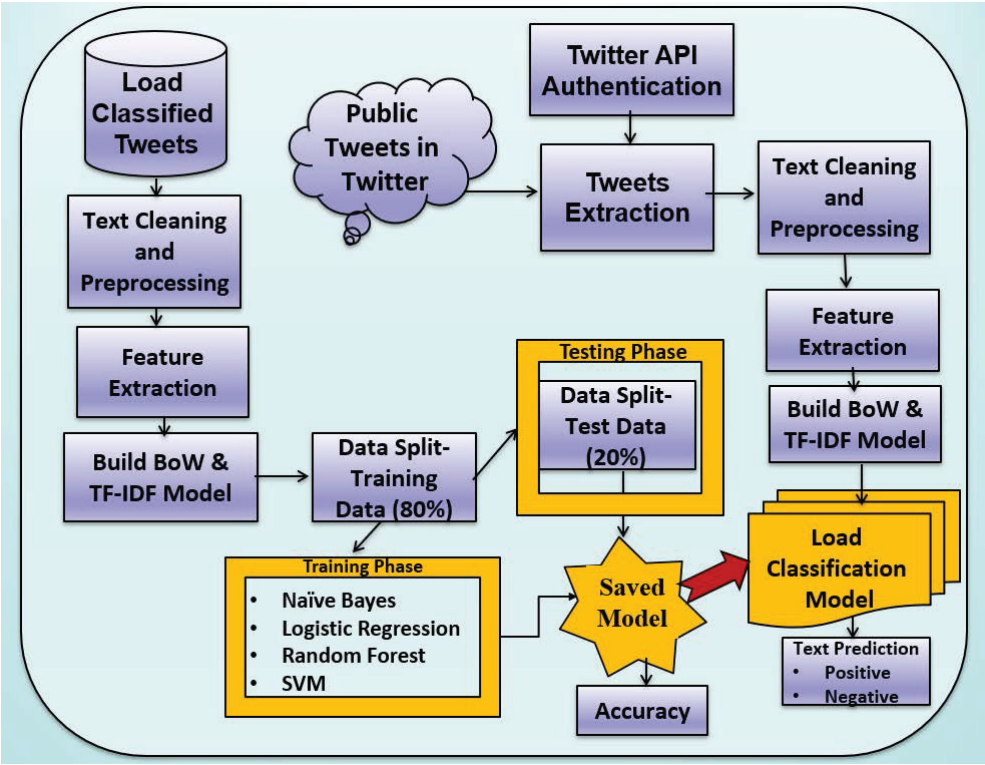

FIGURE 1. Architecture of proposed ML Based Emotion Classifier

famous technique for representing the text using vectors of numerals. The popular forms of word embeddings used here are Bag of Words (BoW).

- Emotion Classifiers: It is used to classify the unstructured test data into positive or negative using Naive Bayes, logistic regression, and random forest and SVM approaches.
- Accuracy Finder: It is used to find the accuracy of each of the classifiers.

Finally, the performance metrics of all the classifiers are compared to provide the optimal classifier.

4. METHODS OF IMPLEMENTATION

This system implementation is divided into three main categories:

A. Enriched Text Cleaner and Pre-processor [4]

B. Feature Extraction – Bag of Words (BoW)

C. Implementation of Various Sentiment Classifiers

A. Enriched Text Cleaner and Pre-processor.

a. Dataset. Tweets are slang words, which are used to express users' emotions about current affairs on Twitter. The sample dataset contains around 1600000 classified tweets. The Sentiment column is corresponding to "label class" holding a value 0 for the negative tweet and 1 for the positive tweet. The dataset is well-balanced between negative and positive sentiment.

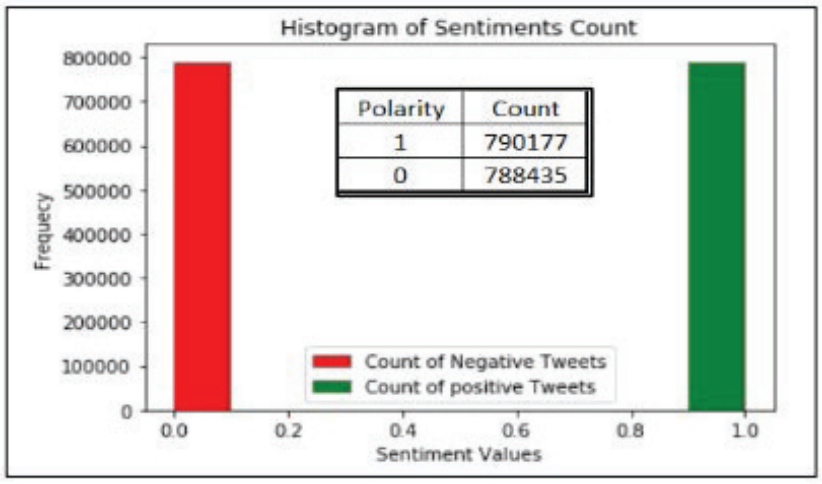

FIGURE 2. **Sample graph – count of labeled tweets**

b. Dictionaries Used for Pre-Processing. The following resources are used to facilitate the preprocessing module of our system:

- Emoticon dictionary
- Acronym dictionary
- Stop word dictionary
- Positive and Negative word dictionaries
- Negative contractions and auxiliaries dictionary – used to detect negation in each tweet

c. Enriched Text Cleaner and Pre-processor. The data preprocessing [4] can often have a significant impact on the performance of a supervised ML algorithm. The steps that are carried out by the enriched preprocessor of this system are as follows:

- Using the emoticon dictionary Substitute all the emoticons with their sentiment polarity value ||pos||/||neg||.

- Replace URLs with a tag ||url|| using Regular Expressions
- Removal of Unicode characters
- Decode HTML entities
- Reduce all letters to lowercase
- Replace usernames/targets @ with ||target||
- Replace acronyms with their translation
- Replace negations like not, no, never by tag ||not||
- Replace sequence of repeated characters by two characters (e.g.: "hellooooo" = "helloo") to keep the emphasized usage of the word

B. Feature Extraction.

a. Bag of Words (BoW) Model. The Bag of Words (BoW) model [5] is the simplest form of text representation in numbers. In this Model, a sentence is represented as a bag of words vector i.e. a string of numbers. For example, consider the following three types of reviews about a movie:

R1: This movie is very scary and long

R2: This movie is not scary and is slow

R3: This movie is spooky and good The core idea behind constructing a Bag of Words (BoW) model is explained below:

- To build a vocabulary, all the unique words in the all three reviews are extracted. It consists of 11 unique words: 'This', 'movie', 'is', 'very', 'scary', 'and', 'long', 'not', 'slow', 'spooky', 'good'.
- Find each of these words' occurrence in the three movie reviews.

TABLE 1. Bag of Words (BOW)

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the review(in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	0	1	1	0	1	0	0	8
Review 3	1	1	1	0	0	0	1	0	0	1	1	6

- Construct Vector for each review using the above data:
  Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]
  Vector of Review 2: [1 1 2 0 0 1 1 0 1 0 0]
  Vector of Review 3: [1 1 1 0 0 0 1 0 0 1 1]

The above word vector representations are used as input to all the machine learning approaches.

C. Machine Learning Algorithms. Once we have completed the different steps of the preprocessing part, we can now focus on the machine learning part [6]. There are three major methods used to classify unstructured text data into positive or negative: Naive Bayes, Logistic Regression, Random Forest, and SVM. The details about each classifier are elaborated as follows:

b. Naive Bayes Classifier [7]. A classifier is a machine learning model that distinguishes different objects based on certain features. Naive Bayes classifier is a machine learning based probabilistic model that is used for text classification. It works based on the principal of Bayes theorem

$$P(y \mid X) = \frac{P(X \mid y) \cdot P(y)}{P(X)}.$$

The probability of 'y' happening, given that X's occurrence had been calculated Using Bayes theorem. At this point, y is called the hypothesis, and X is called the evidence. The hypothesis made at this point is that features are independent of each other. It means the occurrence of one specific feature does not affect the other features. For example, if there are 'n' number of features. Then X is rewritten as:

$$X = (X_1, X_2, X_3, .., X_4),$$

Based on the above equation, Maximum a Posterior (MAP) estimation is constructed by looking for the optimal category which maximizes the posterior P(y|X):

$$c * \operatorname{argmax}(P(y \mid X))$$
$$c * \operatorname{argmax}\left(P(y) \prod P(Xi \mid y)\right) \quad \text{where i=1..... n}$$

There are 3 types of Naïve Bayes [6]:

- Multi-variate Bernoulli Model or Binomial model, useful if the feature vectors are binary (e.g. 0s and 1s). An application can be text classification with a bag of words model where the 0s are used to represent "words do not present in the document" and 1s are used to represent "words present in the document".
- Multinomial Naïve Bayes: This model is used for discrete counts. In-text classification, the Bernoulli model is extended to count the number of times the word 'wi' appears over the number of words rather than saying 0 or 1 if the word present or not.

- Gaussian Model: In this Model, Instead of discrete counts, it has continuous features.

The most used model for text classification is the Multinomial Naive Bayes Model.

c. Logistic Regression. Logistic Regression [8] is a Machine Learning and predictive analysis algorithm that is used for classification problems. It is based on the concept of probability. It is a Linear Regression model, but it uses a complex cost function called 'Sigmoid function' instead of a linear function. The hypothesis of logistic regression model bounds the cost function value between 0 and 1. Therefore linear functions have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression

$$0 \leq h_\theta(x) \leq 1.$$

In Logistic regression, to map predicted values to probabilities, the Sigmoid function is used. The sigmoid function maps any real values into another value which ranges between 0 and 1. In machine learning, sigmoid is used to map predictions to probabilities.

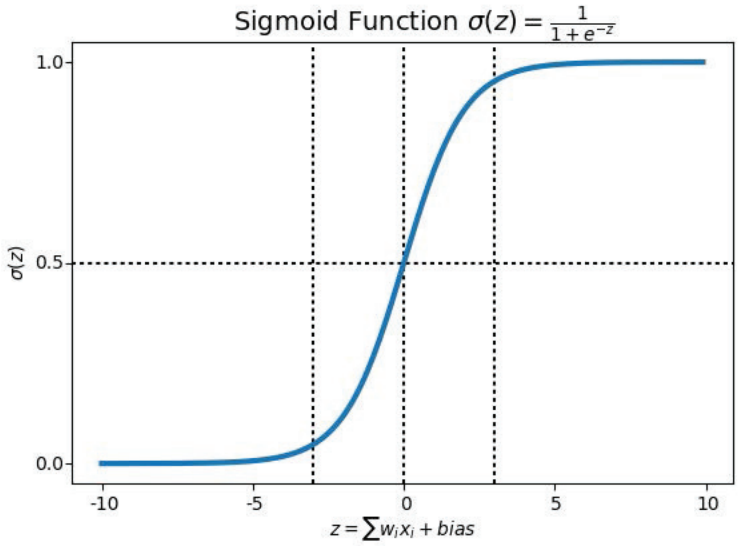

FIGURE 3. Logistic Regression Sigmoid Activation Function

The formula of a sigmoid function | Image: Analytics India Magazine:

$$f(x) = \frac{1}{1 + \mathrm{e}^{-(x)}}.$$

Hypothesis Representation of logistic regression is given by:

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}.$$

The classifier returns the decision boundary as a probability score between 0 and 1 when the inputs passed through a prediction function. For Example, we have 2 classes, Positive and Negative holding values '1' for positive and '0' for negative. Based on a threshold value, if the predicted value is above the threshold then it will be classified into 'positive class' and the predicted value goes below the threshold then it will be classified into 'negative class'.

d. Random Forest. Random Forest [9] is another popular supervised learning technique. It is based on the concept of ensemble learning, where multiple classifiers are combined to solve a complex real-life problem. This technique also improves the efficiency and performance of the model. Random Forest is a classifier that contains 'n' number of decision trees on random subsets of the given dataset obtained using Boostrap Sampling or Bagging. This algorithm predicts the final output by taking the prediction from each tree and based on the majority votes of predictions. The larger number of decision trees in the forest leads to higher accuracy and avoids the problem of overfitting. The working principle of the Random Forest Algorithm is explained below:

- Select 'k' random samples from a given dataset
- Construct a decision tree for every sample
- Obtain the prediction value from every decision tree
- Perform voting for every predicted result
- Obtain the most voted prediction result as the final prediction result

This voting concept principal is known as majority voting.

e. Support Vector Machine (SVM). SVM [10] is a discriminative classifier which is well-defined by a separating hyperplane. Otherwise it is defined as, for a given set of labeled training data, this supervised learning algorithm outputs an optimal hyperplane which classifies new unseen examples. The hyperplane is a line dividing a two-dimensional plane into two parts wherein each class lay on either side. Tuning parameters of SVM are explained below:

- **Margin** A margin is defined as a separation of the line to the closest class points. Many probable hyperplanes could be preferred to separate the two classes of data points. The hyperplane that has the maximum margin,

i.e. the maximum distance between data points of both classes and the hyperplane is considered as an optimal hyperplane. Maximizing the margin distance provides some reinforcement future data points that can be classified more accurately.

- **Kernel** The kernel is used to determine the separation line in a higher dimension when the data points are not possible to separate linearly. If there is no linear line that can separate the two classes in a two-dimensional x-y plane. In this case to draw the hyperplane SVM applies transformation and adds one more dimension called the z-axis. The value of points on the z-axis is calculated using the below formula:

$$w = x^2 + y^2.$$

  The plot in the z-axis will have a clear separation line, between the data points. If the transformation is back to the original 2D plane, it shows a circular boundary as shown in the image below. These transformations are called kernels.
- **Regularization** The regularization or 'C' parameter (lambda) serves as a degree of importance that is given to miss-classifications. For large values of C, a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Equally, a small value of C will cause the optimizer to seek a larger margin hyperplane, even if that hyperplane misclassifies more points.
- **Gamma** This parameter defines how far the influence of a single training example reaches, with low values of gamma means 'far' and high values of gamma means 'close'. That means with low gamma value, points far away from the plausible line are considered for calculating the hyperplane line. Whereas high gamma value means the points close to the plausible line are considered in the calculation.

f. Splitting of Dataset. First, the data set should be divided into training and test set. The following steps are carried out to split the dataset:

> Shuffling of data set to avoid keeping of any order

> Separate positive and negative tweets

> divide 75% of the dataset into training data and 25% of the dataset into testing data

g. Validation Set. It is used to validate the model against unseen data. Also, it is used to tune the possible parameters of the learning algorithm to avoid underfitting and overfitting problems used to occur while training the model. The training dataset is split into two parts 60%/ and 20% with a ratio of 2:8 where each part contains an equal distribution of example types. The classifier will be trained with the largest amount of dataset and predict with the smaller dataset to validate the model. K-fold cross-validation is used for validation. In this, the data set is split into k parts (k=10), hold out one, combine the others, and train on them, then validate against the held-out portion. The same process is repeated k times (each fold), holding out a different portion of data each time. Finally average out the score of each fold to find an accurate estimation of the model's performance.

5. Result Analysis

5.1. **Performance Metrics.** The accuracy of the classifier is evaluated using the following performance metrics:

- Precision
- Recall
- F1-score
- Accuracy
- Confusion matrix The F1 Score is used to measure the accuracy of a classifier and it is calculated as a weighted average of the precision and recall, and it is calculated using the given formula:

$$F1 = \frac{2 * (precision * recall)}{precision + recall}.$$

F1 score lies between 0 – 1 and it reaches its best value at 1 and worst value at 0. Precision is the number of true positives divided by the total number of elements labeled as belonging to the positive class, and it is given by:

$$Precision = \frac{TP}{TP + FP}.$$

The recall is the number of true positives divided by the total number of elements that belong to the positive class, and it is given by:

$$Recall = \frac{TP}{TP + FN}.$$

TABLE 2. Confusion Matrix

(True Positive)	(False Positive)
(False Negative)	(True Negative)

A confusion matrix is a table-like structure that is used to describe the performance of a "classifier" on a set of test data for which the true values are known. It also visualizes the performance of an algorithm. Table 3 shows the confusion matrix Structure.

The accuracy can be defined as the percentage of correctly classified instances and which is given by:

$$\text{Accuracy} = \frac{(TP + TN)}{TP + TN + FP + FN}.$$

5.2. **Experimental Results.** Table 3 describes the values of confusion matrix elements True Positive(TP), False Positive (FP), False Negative(FN), True Negative (TN) obtained with all the classifiers.

TABLE 3. Confusion Matrix of Classifiers

Classifier	True Positive	False Positive	False Negative	True Negative
Naive Bayes	319	88	106	287
Logistic Regression	8595	3401	2795	9209
Random Forest	8792	3204	3771	8233
SVM	9	11987	1	12003

The visualization of the confusion matrix for all classifiers is shown below in the figure.

Table 4 describes the values of performance metrics like Precision, Recall, F1-score, and Accuracy of trained classifiers tested against the test data.

Among all the four classifiers, High accuracy is obtained when tested against the Naïve Bayes classifier. Approximately 76% of accuracy obtained by Naive Bayes. Accuracy of 74% is obtained with logistic regression, 71% with Random Forest, and 50% with SVM.

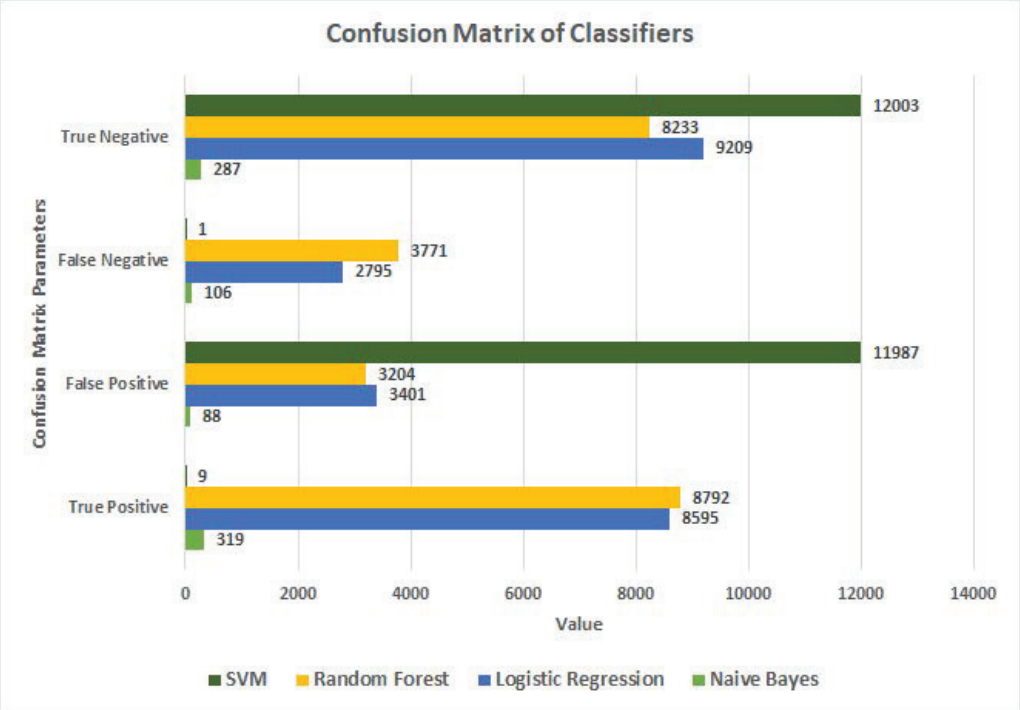

FIGURE 4. Performance Metrics of Classifiers

TABLE 4. Performance Metrics

Classifier	Precision	Recall	F1-score	Accuracy
Naive Bayes	0.76	0.76	0.76	0.758
Logistic Regression	0.75	0.75	0.75	0.741
Random Forest	0.71	0.71	0.71	0.709
SVM	0.7	0.5	0.33	0.5

CONCLUSION

This paper focused in detail about finding what kind of emotions and sentiments expressed in tweets using enhanced preprocessor techniques and various machine learning approaches like Naïve Bayes classifier, logistic regression, Random Forest, and SVM. Results obtained by all the models are compared and evaluated using performance metrics like Precision, Recall, F1-score, and Accuracy. Among all the four classifiers, High accuracy is obtained when tested against the Naïve Bayes classifier. Approximately 76% of accuracy obtained by Naive Bayes. Accuracy of

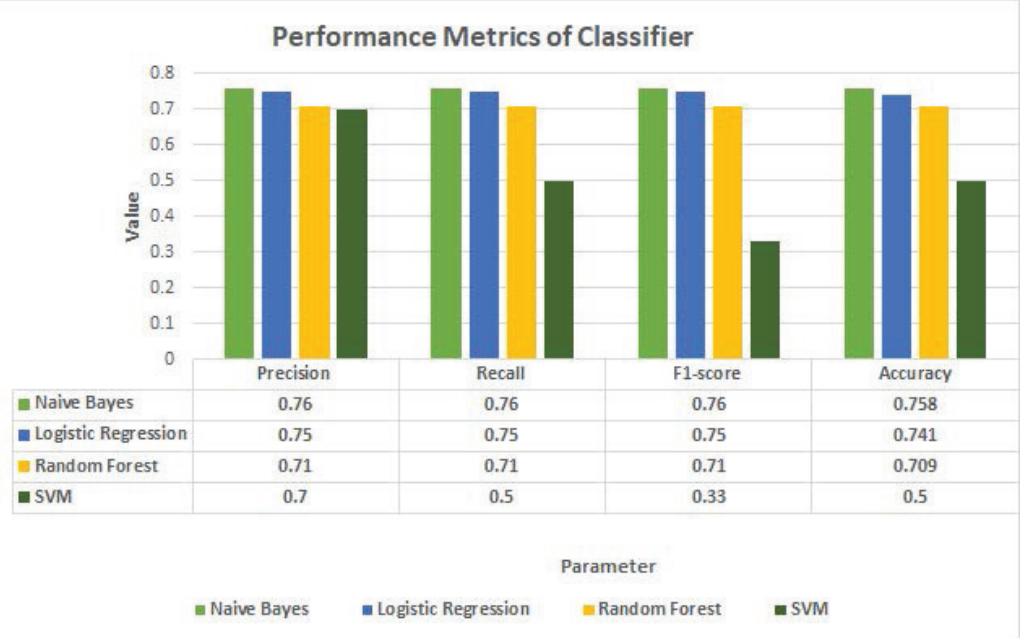

FIGURE 5. Performance Metrics of Classifiers

74% is obtained with logistic regression, 71% with Random Forest, and 50% with SVM.

It also elaborates on the need for the large volume of free social media data available online and finding different user opinions like positive, negative, or neutral. This method of finding user opinion helps the business to create successful social operations, identify influencers for their product, service and industry, compare strengths and weaknesses of competitors, discover the real-time trending topics ie what people are talking about the business and customer opinions and sentiment towards their business.

REFERENCES

[1] R. WAGH, P. PUNDE: *Survey on Sentiment Analysis using Twitter Dataset*, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, 2018, 208-211.

[2] K.JAYAMALINI, M.PONNAVAIKKO: *Enhanced social media metrics analyzer using twitter corpus as an example*, International Journal of Innovative Technology and Exploring Engineering, **8**(7) (2019), 822-828.

[3] *https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json*, accessed March 2020.

[4] B. BILLAL, A. FONSECA, F. SADAT: *Efficient natural language pre-processing for analyzing large data sets*, 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, 3864-3871.

[5] W. A. QADER, M. M. AMEEN, B. I. AHMED: *An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges*, 2019 International Engineering Conference (IEC), Erbil, Iraq, 2019, 200-204.

[6] M. RATHI, A. MALIK, D. VARSHNEY, R. SHARMA, S. MENDIRATTA: *Sentiment Analysis of Tweets Using Machine Learning Approach*, 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, 2018, 1-3.

[7] G. SINGH, B. KUMAR, L. GAUR, A. TYAGI: *Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification*, 2019 International Conference on Automation, Computational and Technology Management (ICACTM), London, United Kingdom, 2019, 593-596.

[8] A. PRABHAT, V. KHULLAR: *Sentiment classification on big data using Naïve bayes and logistic regression*, 2017 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2017, 1-5.

[9] P. KARTHIKA, R. MURUGESWARI, R. MANORANJITHEM: *Sentiment Analysis of Social Media Network Using Random Forest Algorithm*, 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019, 1-5.

[10] K. HAN, C. CHIU, W. CHIEN: *The Application of Support Vector Machine (SVM) on the Sentiment Analysis of Internet Posts*, 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2019, 154-155.

COMPUTER SCIENCE ENGINEERING,
BHARATH UNIVERSITY,
CHENNAI, INDIA
Email address: malini1301@gmail.com

BHARATH UNIVERSITY,
CHENNAI, INDIA
Email address: ponnav@gmail.com

SHREE L R TIWARI COLLEGE OF ENGINEERING
Email address: jayamalini.k@slrtce.in