Sentiment Analysis: An Assessment of Diverse Methods

Hrithik Goswami, Vaibhav Gupta, Rachna Jain

[1]Department of Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, Delhi, India

hrithikgoswami.cse1@bvp.edu.in, vaibhavgupta.cse1@bvp.edu.in, rachna.jain@bharatividyapeeth.edu

Abstract: Digitalization over the years has greatly impacted the inevitability of consumer reviews in the online sphere. Analysing a review given to a product has always been a crucial need, and these reviews are very vital as they shape the overall product, thereby allowing the customer to gain hindsight about the product that they might intend to buy. But a single product can itself have a colossal number of reviews, and thus it becomes very difficult at times for the customer to choose a product. Therefore, if there is a suitable mechanism that can help the buyer and seller to analyze the products, then it can greatly solve the problem of decidability. Hence, we have carried out this research in which we compared five machine learning classifiers: Multinomial Naïve Bayes, Support Vector Machine, Logistic Regression, Decision Trees, and Random Forest Classifier, on the Amazon phone reviews. We utilized the feature extraction technique of TF-IDF to convert the textual data into numerical form and used evaluation metrics such as precision, recall, f1-score, and accuracy to assess our models. Our evaluation and analysis show that a Random Forest gives the best possible suitable result for the chosen data; this was additionally evaluated by tuning the hyperparameters of the Random Forest using out-of-bag error and 3-fold cross-validation techniques, and it showcases an improvement in accuracy with the former method.

Keywords: Sentiment Analysis; Mobile Phone Reviews; TF-IDF; Multinomial Naïve Bayes; Support Vector Machine; Logistic Regression; Decision Tree; Random Forest Classifier.

1. Introduction

In the ever-growing and evolving virtual world, the needs of people have drastically shifted towards the online marketplace, and they also rely mainly on reviews to make a final decision. And not only that, but organizations also have a greater dependency on reviews as it helps them to improve and gives them a chance to meet the needs of the consumer and thus enhance their products. As there are enormous reviews available, there is a huge need to come up with a method that would classify and understand these reviews in a real sense, thereby helping the customer by giving them a general idea about the product. And for this, sentiment analysis and classification play a key role by extracting the important sentiments given in the form of the reviews and classifying them accordingly. Sentiment Analysis is a process that uses biometrics, natural language processing, computational linguistics, and analysis of text to extract information conveyed in a chunk of text that defines the human sentiments delivered through it [1]. In this paper, we have focused on this issue and have worked on the phone reviews given on Amazon.com, as it is one of the biggest e-commerce platforms that provide a range of mixed reviews. We worked with various classifiers and did sentiment analysis of the reviews and classified them as positive, negative, or neutral, thus helping the producer and consumer by providing them with a state of mind about the products.

With the advent of growth in the e-commerce market, the reliability of online reviews has considerably increased over the past years. It has become very crucial to classify the reviews to meet the deeper needs of both the buyer and seller on the online platform. The classified reviews can thereby greatly help to form a mindset for the product, as solely relying on the whole reviews, which are in huge numbers, is a critical task to comprehend [2]. So, in this research, we have worked on this challenge by finding the polarity of a review about a product to estimate the correctness of classification algorithms using several assessment metrics. In addition to this, methods like out-of-bag error and cross-validation were also applied to the best classifier to tune its performance and test it on the desired measures.

This paper is outlined as follows: in section 2, we discussed various literature reviews that have worked towards a similar problem. In section 3, we discussed the various machine learning methods used to carry out a comparative analysis. Section 4 focuses on the data, its features,

methodology, and implementation. In sections 5 and 6, we have given the experimental results obtained, and a comparison of the same is also done. In section 7, analysis of the best classifier is carried out based upon the experimental results attained in previous sections. Finally, we concluded the findings of our paper with possible future scope in section 8.

2. Related Work

Several distinct but similar work in this field on a variety of data has been done in the recent past, and they have been considered in this review.

Callen Rain in [3] utilized the current work done in the area of NLP on the reviews on Amazon and used Naïve Bayesian and Decision list classifiers for sentiment analysis and also compared features like bag-of-words and bigrams for their efficacy. Their analysis showed that the Naïve Bayes classifier worked well with over 800 features as well as the highest accuracy was also obtained with it. K. Ghag and K. Shah in [4] did a comparative investigation of sentiment analysis on the detection of the polarity of tweets as positive, negative, and neutral using the lexicon and non-lexicon methods and realized that the sentiment analyzers are centered around the language, with managing negation and language generalization being the major problems. Xing Fang and Justin Zhan in [5] worked on the problem of sentimental polarity categorization on Amazon product reviews with diverse classification algorithms like the Random Forest, Naïve Bayes, and Support Vector Machine, where the performance of each was studied based upon their ROC curves and f1-score metrics. Both the classifiers used by them, viz. Naïve Bayes and SVM were observed to be better than the Random Forest. Muhammad T. Khan et al. in [6] discussed the various sentimental analysis techniques and emphasized the numerous challenges that are faced with natural language processing. Mohan Kamal Hassan et al. in [7] did a sentimental analysis of the laptop product reviews on Amazon.com using Naïve Bayes. The above analysis showed us that it performed optimally with bigrams and stop words as compared to single words with an accuracy of approximately 90% for over 10000+ samples.

Heide Nguyen et al. in [8] used three machine learning and three lexicon-dependent techniques to carry out the sentiment analysis of product reviews on Amazon. The assessment showed that all the three former models outperformed the latter models on all the evaluating metrics: precision, recall, and f1-score. Abhilasha Tyagi and Naresh Sharma in [9] used Logistic Regression with a unigram feature vector to perform sentiment analysis on the data of Twitter by speeding up the classification process. They applied a useful word score heuristic to obtain the scores of frequently used words. Wanliang Tan et al. in [10] used traditional and modern machine learning methods, viz. Naïve Bayes, K-Nearest Neighbour method, Recurrent Neural Network (RNN), Support Vector Machines, etc. to perform sentiment analysis of product reviews on Amazon, and LSTM gave the best results. Momina Shaheen et al. in [11] mined the mobile-phone product reviews from Amazon to predict the ratings as positive and negative, and lastly, they did a comparative analysis of eight classifiers, of which the Random Forest showed the best result with 85% accuracy. Sara A. Aljuhani and Norah S. Alghamdi in [12] carried out a contrast of different algorithmic methods, namely Logistic Regression, Stochastic Gradient Descent, Naïve Bayes, and Convolutional Neural networks (CNN) of mobile phone reviews on Amazon and found that the convolution neural network with word2vec gave the best results with 92.72% accuracy for unbalanced data and 79.60% with balanced data. They also used the Lime technique for assessing the possible logical explanations for the reviews being classified into different polarities.

Jayakumar Sadhasivam and Ramesh B. Kalivaradhan in [13] used an ensemble approach with the currently existing models, viz. Naïve Bayes and SVM, to perform sentiment analysis on Amazon reviews available on the official product site, and then the product is recommended based on the analysis. Hui Zhang in [14] analyzed the Amazon Alexa reviews to study the sentimental aspect of the nature of such reviews by working with Naïve Bayes and Logistic Regression classifiers. The analysis predicted that Logistic Regression had performed slightly better than Naïve Bayes with an accuracy of 87.4% as compared to 87.1% for Naïve Bayes on

the unbalanced dataset. The dataset, which was unbalanced in nature, was balanced by the SMOTE technique, which led to advancing the ROC curve (AUC) scores of these two models from 0.5 to 0.8. Emilie Coyne et al. in [15] discussed the performance of three algorithms, namely Multinomial Naïve Bayes, Long Short-term Memory network (LSTM), and Linear Support Vector Machine (LSVM) based upon the sentimental analysis of 60,000 product reviews selected randomly from Amazon.com. The analysis determined that LSTM performed better among them, with an accuracy of 90%. The best results with LSTM were achieved on the remaining 3.94 million reviews with an accuracy of 92%. Vineet Jain and Mayur Kambli in [16] discussed the sentimental analysis on Amazon product reviews with different supervised and unsupervised machine learning techniques and models like Naïve Bayes, Logistic models, etc. were evaluated based upon their bag of words accuracy and TF-IDF scores, where both of these models performed similarly. K. Ashok Kumar et al. in [17] used supervised machine learning methods to perform sentiment analysis of Amazon product reviews, and their model is capable of determining whether the consumer intends to propose the product or not.

3. Machine Learning Methods

To carry out our research, we have done a comparison of five diverse methods on reviews to assign them with different polarities by building a prediction model on the Amazon mobile phone reviews dataset.

A. Multinomial Naïve Bayes

This model is a widely used classifier for the classification problem that has discrete features. It follows a probabilistic approach in which the feature vectors signify the count of frequencies, and certain events are produced by a multinomial. If we have a class y with n features, then the distribution is parametrized by vectors $\theta_y = \theta_{y1},\theta_{y2},\ldots,\theta_{yn}$ where θ_y is assessed using relative frequency counting:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Where in a sample of class y, θ_{yi} is the probability $P(x_i \mid y)$ of the feature i; N_{yi} is the total occurrences of i in y and for class y, N_y is the cumulative count of all the features, with α being the smoothing priors [18, 19].

B. Support Vector Machine

This model is a robust supervised learning algorithm that is created on a learning framework and is based upon statistics. This model is mainly used for classification problems. If we plot different groups or classes of data in an n-dimensional space then SVM performs classification by finding a hyperplane in this space, that differentiates the class of data. And it draws these hyperparameters by transforming the data using kernels.

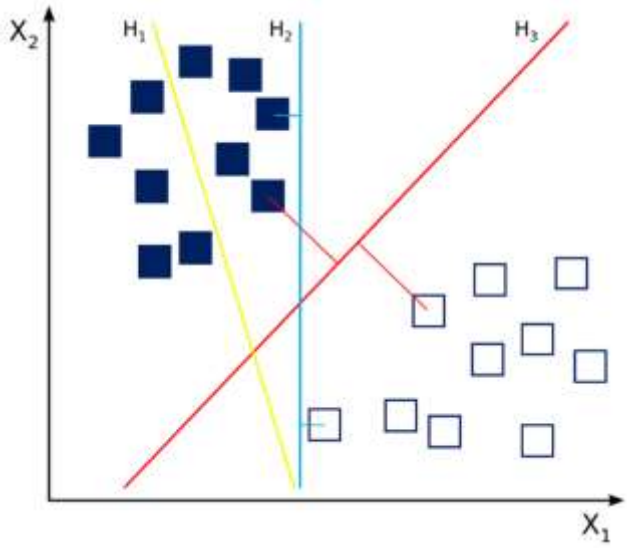

Fig. 1. Classification with the help of SVM.

The hyperparameter having the largest margin or distance from the classes of data is then chosen as the best hyperparameter [20]. In Fig. 1, the squares represent the support vectors with H_1, H_2, and H_3 as the margin classifiers. The H3 plane separates the data with the largest margin and therefore correctly divides the data.

C. ***Logistic Regression***

It is a predictive model that is significantly used for predictive analysis and in cases when the target variable is categorical [21]. And for our research, we have specifically used Multinomial Logistic Regression that classifies the review as positive, negative, or neutral by re-running the binary classification for each class multiple times. This method is implemented by choosing a threshold value that helps in differentiating a class of data and thereby helps in the analysis of the reviews.

D. ***Decision Trees***

It is one of the most basic and useful predictive models that can be used as a regressor or a classifier and has a tree structure in which each node signifies the test value of a certain attribute, each edge links to the result of a test, and it joins directly to the next node. The terminal nodes are the end nodes of the tree that ultimately predicts the sentimental outcome conveyed through the reviews. These work on the principle of binary recursive partitioning, where the data is split into partitions and then into branches [22].

E. ***Random Forest***

It is a model that can also be used as a regressor or a classifier, but it contains a collection of decision trees as an ensemble. And this is a much more efficient model than the Decision Tree, as a group of decision trees will outperform to give a much better result. For this, it makes use of bootstrapping and bagging and has two pre-requisites: one, there should be an actual signal in the features, and second, the predicted values of each decision tree should have less correlation with each other [23]. If we take an individual tree, then during the splitting of each node, every feature is taken into consideration, and the feature that has a significant number of separations between the observations on the left and the right node is preferred. In the Random Forest, each separate tree can choose from a subgroup of features, and hence this results in a lower correlation across many trees.

4. Data and Methodology

Here, we have unveiled the methodology and procedure utilized to classify the polarity of mobile phone reviews. The dataset is categorized into two groups, viz. training and testing. The

former set is used to understand the classifier, whereas the latter one is used to test and assess the score of our classifier. And Fig. 2 shows the methodology followed.

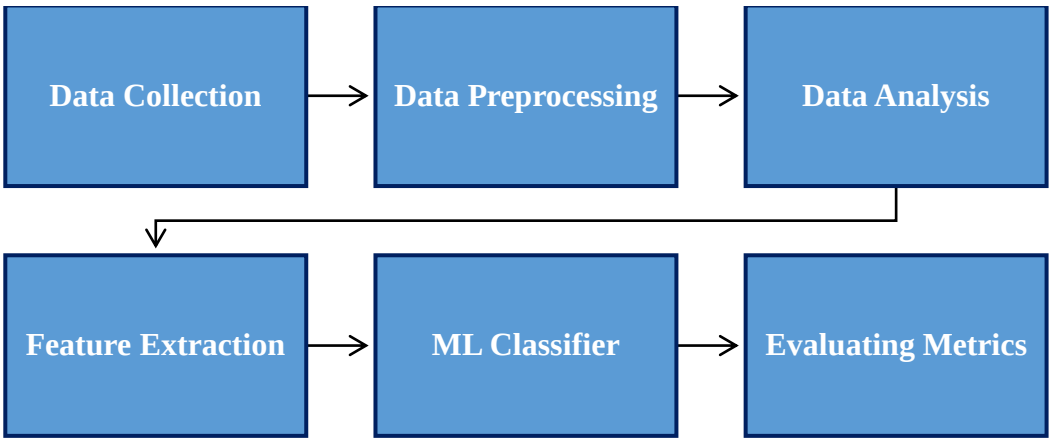

Fig. 2. Approach followed.

a. Data Collection

For this research, we chose the Amazon reviews of unlocked mobile phones, and they are available on Kaggle. This dataset file is available in a comma-separated values (CSV) format. The dataset has more than 400,000 reviews of different variety of unlocked mobile phones, and it primarily consists of 6 columns, namely:

i. *Product Name:* This column contains the name of the product. For example, the Sprint EPIC 4G Galaxy SPH-D7.
ii. *Brand Name:* This column contains the brand of the corresponding product. For example, Samsung.
iii. *Price:* This column contains the cost of the product. For example, the cost of the Sprint EPIC 4G Galaxy SPH-D7 is $199.
iv. *Rating:* This column contains the rating of the corresponding product in a range of 1 to 5.
v. *Reviews:* This column gives the description of the user experience that he/she has given to the product on Amazon.
vi. *Review Votes:* This column contains the number of people who find these reviews useful.

b. Data Preprocessing

After the data is collected, it is pre-processed for further analysis. This step involves multiple procedures that are carried out to ensure efficient working with data. It involves 6 steps:

i. *Tokenization:* In this, we separate a piece of text into smaller units which are termed tokens. These tokens can be words, characters, sub-words, phrases, and symbols in which we discard the punctuation marks to allow simpler and efficient analysis.
ii. *Removing Stop Words:* Here, we discard all stop words, that do not convey significant importance to the structure of the input sentence (review), and therefore helps in increasing the total efficiency of data preprocessing.
iii. *Conversion to Lower Case:* In this step, all the upper-case words were converted to lower-case to avoid ambiguity in the data.
iv. *Stemming:* Now in this step, we reduce the words into a root, also known as a stem, to allow effective working with the data. Basically, in this step, we remove the unnecessary suffix and thereby increasing the accuracy of the classification model.
v. *Removing Punctuations:* In this step, all the punctuation marks, like a full stop, comma, colon, etc., are removed.
vi. *Labeling the Data:* Finally, in this step, we categorize the column '*Rating'* into 3 parts: positive (labeled as 2), neutral (labeled as 1), and negative (labeled as 0).

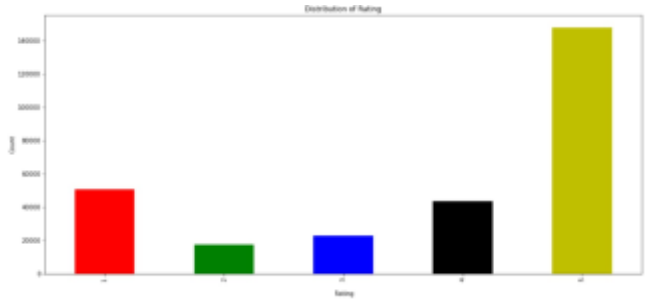

Fig. 3. Distribution of rating from 1–5 with respect to their count.

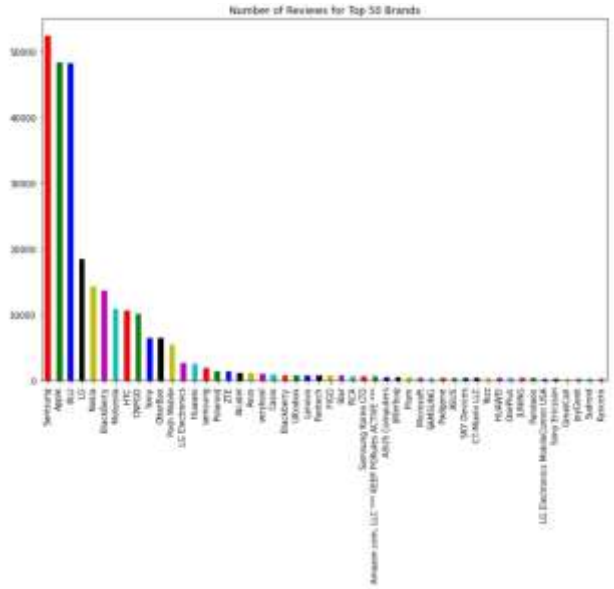

Fig. 4. The number of reviews for top 50 brands.

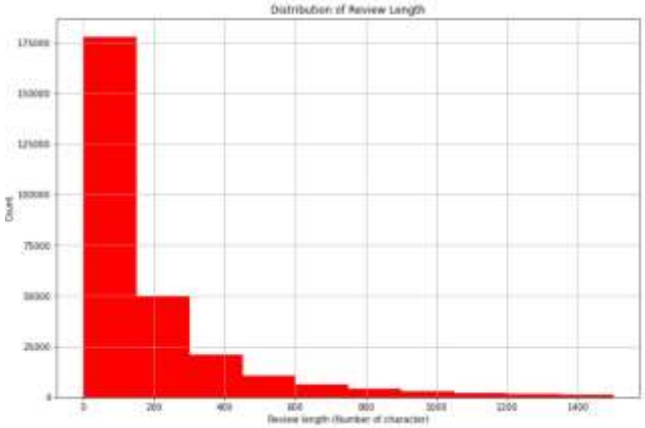

Fig. 5. Distribution of reviews length and their rating.

c. Data Analysis

After the pre-processing, the data is analyzed for further action. And we carried out the analysis in the following ways:

i.*First*, we analyzed the inclination and distribution of the ratings and observed that most of the ratings, which are approximately 140,000+, were given 5 stars for a variety of mobile phones. This rating is based on the scale of 1–5 of Amazon's rating scale. This is depicted in Fig. 3.

ii. *Second*, we analyzed the number of reviews that were given for the top fifty brands and observed that most reviews were given to 'Samsung', followed by other brands, with over 50,000+ reviews given to a variety of 'Samsung' products and approximately 45,000+ reviews were given to the 'Apple' brand. Fig. 4 illustrates this.

iii. *Finally*, we analyzed the distribution of the review length, which is the total count of characters in a particular review, and observed that over 175,000+ were less than 200 characters long. This is shown in Fig. 5.

d. Feature Extraction

After the analysis, the necessary features are extracted. As for this research, as we are working on a textual dataset, it can not be directly fed into the model. Therefore, they are first converted into numerical form and then are worked upon for further evaluation. This new format summarizes most of the information conveyed through textual data. And this is done using the Term Frequency-Inverse Document Frequency (TF-IDF) method. In this, the words are evaluated on the basis of their relevancy in the whole review [24]. Term Frequency is the frequency counter of the word in the entire corpus, and Inverse Document Frequency measures the informativeness of that word in the whole set of the corpus. Every individual word has its own set of TF and IDF scores, so multiplying these two scores results in a TF*IDF score for that word in the corpus [25]. This score helps in evaluating the rarity of a word, i.e., rarity is directly proportional to the value of this score. The higher the score, the higher is the rarity of that word. And with greater rareness, the word is more relevant and tends to appear in top search results. This helps in intercepting the usage of stop words easily [26].

e. Evaluating Metrics

Finally, after the feature has been extracted, the metrics are used to examine the performance of the models. We use a confusion matrix to describe the performance of each model. This matrix is a table with four different possible values for actual and predicted classes. The confusion matrix is illustrated in Table I.

True Positive (TP) depicts correctly predicted event values, False Positive (FP) depicts incorrectly predicted event values, True Negative (TN) depicts correctly predicted no-event values, and lastly, False Negative (FN) depicts incorrectly predicted no-event values [27, 28]. And this is used for measuring the following parameters:

i. *Precision:* This represents the proportion of predicted positives that are true positives. This metric measures the exactness of the review classified as a positive sentiment [28]. And it is represented by (1).

$$P = \frac{TP}{TP + FP} \tag{1}$$

ii. Recall: This is the proportion of real positives to the entire number of probable positive predictions that can be classified properly. It measures the susceptivity of the review classified as negative sentiment [28]. And it is represented by (2).

$$R = \frac{TP}{TP + FN} \tag{2}$$

iii. F1-Score: It represents the weighted harmonic mean of both precision and recall [27, 28]. And it is represented either by (3) or (4).

$$F = \frac{2 * P * R}{P + R} \tag{3}$$

$$F = \frac{2 * TP}{2 * TP + FP + FN} \quad (4)$$

iv. *Accuracy:* It represents the percentage of true results in the total number of cases that are investigated [28]. And it is represented by (5).

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

5. Discussion And Results

The experimental findings from five different classifiers, including Multinomial Naïve Bayes, Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest Classifier, are shown here. Tables II-VI depict the outcomes of the evaluating metrics for three types of labeled data, viz. 0 for a negative, 1 for neutral, and 2 for a positive. The overall result showed that the Random Forest Classifier worked better with the dataset and gave an overall accuracy of 92.33%.

TABLE I. CONFUSION MATRIX

		Actual Class	
		Positive	***Negative***
Predicted Class	***Positive***	True Positive	False Positive
	Negative	False Negative	True Negative

TABLE II. METRICS FOR MULTINOMIAL NAÏVE BAYES CLASSIFIER

Label	Precision	Recall	F1-Score	Accuracy
0	0.79	0.80	0.80	85.19 %
1	0.45	0.22	0.29	
2	0.89	0.94	0.92	

TABLE III. METRICS FOR SUPPORT VECTOR MACHINE CLASSIFIER

Label	Precision	Recall	F1-Score	Accuracy
0	0.81	0.86	0.83	87.55 %
1	0.69	0.19	0.30	
2	0.90	0.96	0.93	

TABLE IV. METRICS FOR LOGISTIC REGRESSION CLASSIFIER

Label	Precision	Recall	F1-Score	Accuracy
0	0.81	0.85	0.83	87.26 %
1	0.61	0.18	0.28	
2	0.90	0.96	0.93	

TABLE V. METRICS FOR DECISION TREE CLASSIFIER

Label	Precision	Recall	F1-Score	Accuracy
0	0.84	0.83	0.84	88.59 %
1	0.66	0.57	0.61	
2	0.93	0.94	0.93	

TABLE VI. METRICS FOR RANDOM FOREST CLASSIFIER

Label	Precision	Recall	F1-Score	Accuracy
0	0.92	0.85	0.90	92.33 %
1	0.97	0.52	0.68	
2	0.92	0.99	0.95	

TABLE VII. CLASSIFIERS AND THEIR CORRESPONDING ACCURACY

Classifiers	Accuracy
Multinomial Naïve Bayes	85.19 %
Support Vector Machine	87.55 %
Logistic Regression	87.26 %
Decision Tree	88.59 %
Random Forest	92.33 %

6. Comparison of the Classifiers

We have now compared the outcomes of the five classifiers that were obtained in the previous section. And this comparison is done by considering accuracy. Table VII shows the comparative analysis with accuracy. The results obtained from all the classifiers were analyzed. The Random Forest gives us the best outcomes and outperformed all the other classifiers with the highest accuracy of 92.33%, thus it works well with the given dataset of Amazon mobile reviews, while the Multinominal Naïve Bayes classifier showed the least accuracy of 85.19%. Hence, it is important to examine the reasons for the outperformance of the Random Forest over others in the dataset. As mentioned earlier, a Random Forest can be used as a classifier or a regressor that is majorly an ensemble of many Decision Trees. This factor alone provides it with the following advantages over Decision Trees and other bagging classifiers that have the same hyperparameters as the Random Forest when trained on a large variety of datasets:

i. *Performance:* The prediction score computed by the Random Forest is the prediction score of the majority of trees in the Random Forest for a given target variable, and the majority outweighs the prediction score of an individual tree. Additionally, the overall prediction error is also reduced when we take the average of prediction scores of multiple trees in the Random Forest giving the same numerical value for the target variable.

ii. *Robustness:* The probability of overfitting a Random Forest is low as compared to an individual Decision Tree or other classifiers that we have used. This is attributed to randomness in the feature selection while splitting the node. Moreover, when we compare a single Decision Tree with the Random Forest, we generally have a high-variance estimator in hand as the prediction estimated by a single Decision Tree can be greatly impacted if we make a small change in the dataset used for training the model. The Random Forest provides us with a chance to make a low-variance estimator by making an ensemble of many Decision Trees where we will use the sampling technique with the replacement of samples for every tree in the Random Forest that is going to be utilised in aggregation for the overall prediction of the model.

iii. *Scalability:* Another important advantage that comes with the Random Forest is its ability to automatically scale the importance of each feature by considering the impurity or error that comes into the prediction of the nodes of the trees. This can help the model to give more relative importance to a feature that introduces less impurity in the overall prediction.

The above inherent advantages of a Random Forest are not the only factors that make us biased towards using this model over others, but we can also improve its performance and training speed by tuning its hyperparameters to either increase its execution speed or the comprehensive prediction accuracy of the model.

7. Experimental Evaluation of Random Forest

The result of the comparative study shows that Random Forest performed optimally on the unbalanced dataset of Amazon mobile phone reviews. These results encouraged us to analyze the performance of the Random Forest using methods like out-of-bag and cross-validation to fine-tune the Random Forest's hyperparameters. The purpose of doing this analysis only on the Random Forest is inclined towards the aim of finding an optimal classifier that may have higher chances of performing better as compared to other standard classifiers used to carry out sentiment analysis for product manufacturers. The Random Forest is an adjustment provided for the bagged decision trees to form a large number of de-correlated trees that can additionally enhance predictive execution with reasonably less need for hyperparameter tuning [29]. However, simple alteration of bagged trees can result in tree-correlation that in turn restricts the impact of variance reduction. This is conquered by mixing more randomness into the tree-developing cycle [30]. As through the algorithm, a bootstrap sample is arbitrarily chosen for training and a random sample of features for each split, therefore a more different arrangement of trees is generated that will in general decrease the tree-correlation and raise the predictive power significantly. The variables and limits of the Random Forest are the parameters utilized to split each node throughout training, and Scikit-Learn [31] is not an assertion to provide an ideal solution as it applies a default set of reasonable hyperparameters for all the models. Thus, it is impractical to determine the best hyperparameters beforehand and a greater need arises to rely on an analytical approach.

We now have a trained Random Forest model which is optimal for the Amazon mobile reviews, but in pursuit of a greater degree of optimal performance, we analyzed the performance of this classifier. There are many ways to do this. For example, we can collect more data and then perform feature engineering, and this generally gives the best result in terms of the time contributed to the improved performance. But once all the data sources have been drained and are unknown to the variables of manipulation, then we can tune the hyperparameters of our model. We have tuned the parameters of the Random Forest in primarily two ways.

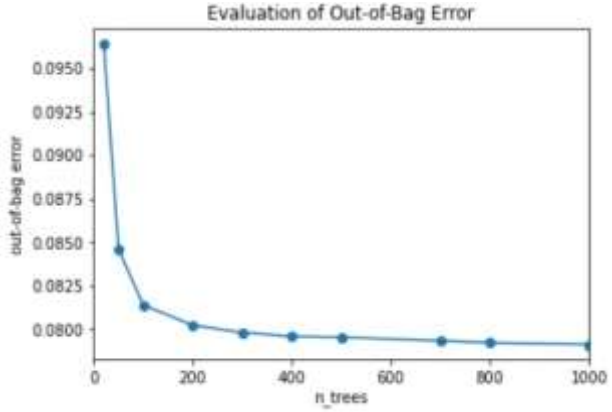

Fig. 6. Distribution of out-of-bag error and n_estimators (n_trees).

a. Tuning of Random Forest by Out-of-bag Error

Out-of-bag error is a technique utilized for estimating prediction errors for the lowest variance results. This error is essentially the average error for each training observation determined by utilizing the predictions from the trees that do not have these training observations in their corresponding bootstrap sample, thus allowing the random forest to be fit and validated while it is being trained [30]. When the samples are prepared, certain data points fail to be a part of a specific sample during training and form the out-of-bag points. In Fig. 6, we have shown a demonstration of how the out-of-bag error can be helpful for choosing a rough appropriate estimation of n_estimators, i.e., the number of trees (n_trees) at which the error balances out, and here it stabilizes near 1000.

b. Evaluation with Out-of-bag Error

The out-of-bag error estimate can be calculated from the out-of-bag score, i.e., the oob_score parameter of the Random Forest that is set to be true. It is represented by (6).

$$oob\_error = 1 - oob\_score \quad (6)$$

There are various advantages to using oob_score for the analysis of the Random Forest, like no data leakage, better predictive models, and no overfitting of the model resulting in less variance, which comes at the expense of more time spent on validating the model using oob_score. The overall evaluation metrics revealed that the Random Forest performed slightly better when we selected n_estimators as 1000, and the experimental result is depicted in Table VIII.

TABLE VIII. METRICS FOR RANDOM FOREST CLASSIFIER WITH OOB_SCORE

Label	Precision	Recall	F1-Score	Accuracy
0	0.92	0.88	0.90	92.55 %
1	0.97	0.52	0.68	
2	0.92	0.99	0.96	

c. Cross-validation of Random Forest

Cross-validation is a method that is used in the estimation of metrics that measures the performance of any classifier by first training it on one subgroup of the data and then evaluating its performance based upon certain metrics on another subgroup of the data from the original dataset [32]. Generally, we estimate the training error of the model by evaluating it after training. But this limits us to know the performance of our classifier only on the data on which the training was performed. However, when we test our model on an unknown dataset, our classifier might be overfitted [33]. An overfitted model may give impressive results on the training dataset but can not be applied to new real-world applications. Therefore, the standardization for optimizing the hyperparameters that account for overfitting of the data through cross-validation techniques is as follows:

i. *Holdout Method:* It is one of the simplest cross-validation techniques in which the input data is divided into different sets of data [32]. The Random Forest is first trained, and then it undergoes testing subjected to a different set of performance metrics. The dataset can be split into any standard ratio, like 80:20. The metrics obtained in Table VI are computed through the holdout method, in which "reviews" of the Amazon mobile reviews dataset are mixed up anyhow before it undergoes splitting. Though we have trained the model on a different combination of samples, it can not guarantee that the training set that has been selected demonstrates the whole data.

ii. *Cross-validation with K-fold:* It is a method that is used to enhance the basic holdout method as when the data is limited, removing a part of it for validation may give rise to underfitting, so we can utilise this method in which the data is separated into distinct subgroups of a number k and the simple holdout based idea is rehashed k number of times [32], thus ensuring that the score of the Random Forest is not dependent upon the way we select our training and testing set, resulting in a less biased model compared to other methods. And generally, k is taken as 3, 5, or 10 while using this method.

d. Evaluation of Metrics with Cross-validation

A hyperparameter is a model parameter that is defined before training begins [33]. Various models have numerous diverse hyperparameters that can be set, and we have validated our model using 3-fold cross-validation on the following four parameters for the analysis of the Random Forest:

i. *n_estimators:* This parameter represents the total number of trees in the Random Forest which will be generated when it grows during the training and testing of the data. The

standard value of this parameter is 10. We can get better performance if we use higher values of this parameter, but the training time of the model is compromised. The validation curve was created for the n_estimators parameter for a range of values from 20 to 300 and it is depicted in Fig. 7 and 8. The distributions indicate that despite the substantial divergence in training and cross-validation scores, for each of the three cross-validations, the mean training accuracy was more than 98%, while the mean cross-validation accuracy was between 88% and 90% for all n_estimator values. This demonstrates that despite the large number of n_esitmators employed, the Random Forest is moderately accurate.

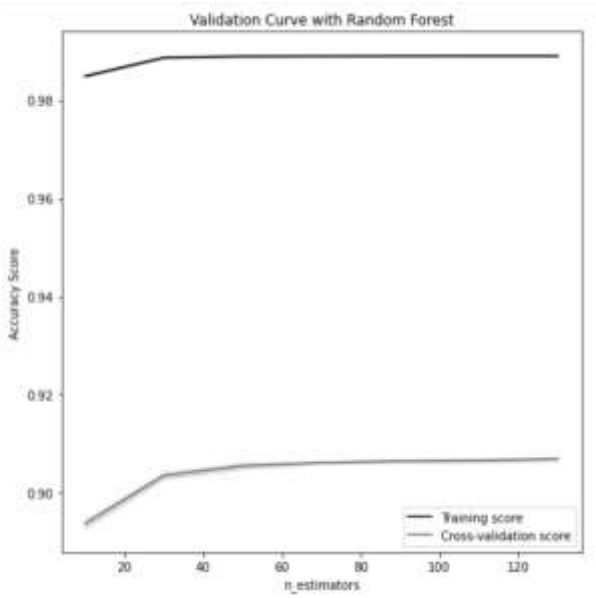

Fig. 7. Distribution of score and n_estimators.

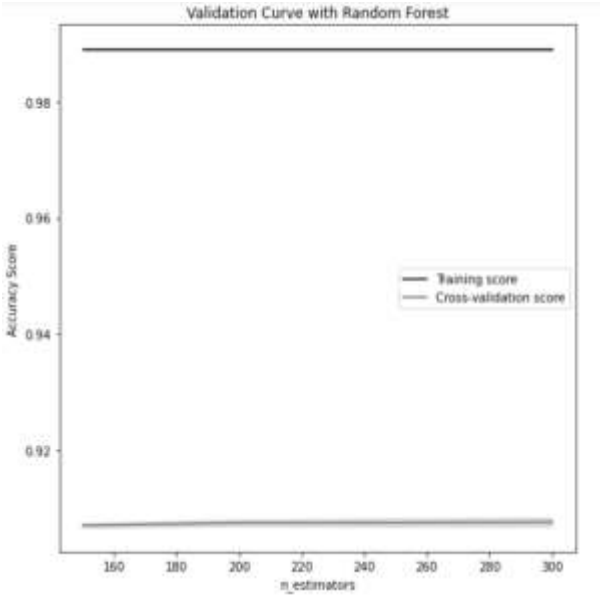

Fig. 8. Distribution of score and n_estimators.

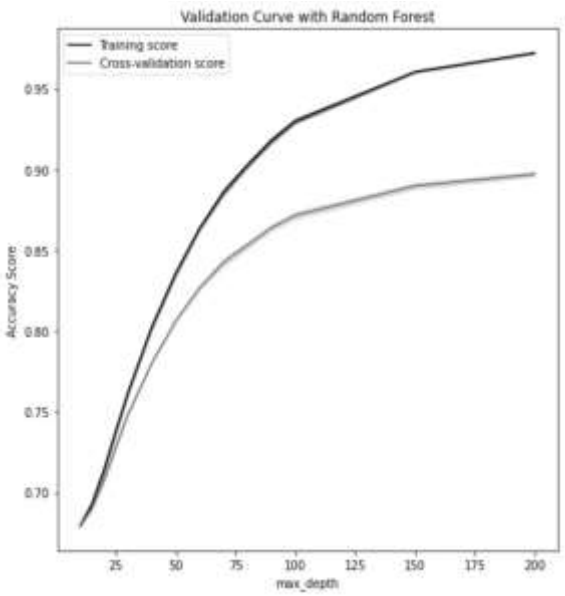

Fig. 9. Distribution of score and max_depth.

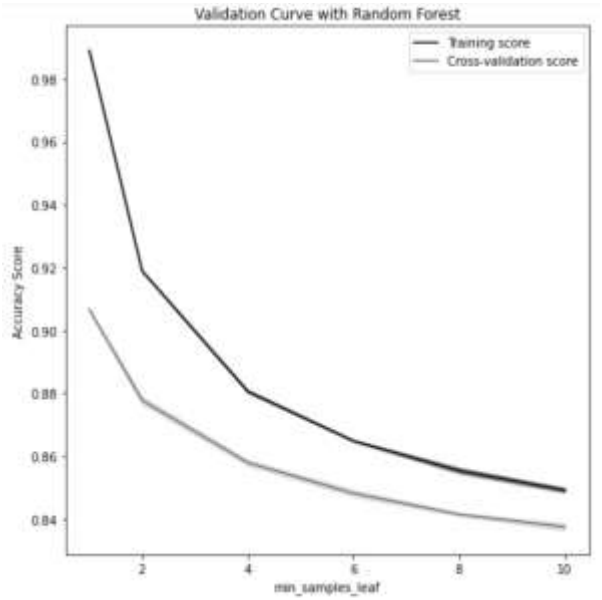

Fig. 10. Distribution of score and min_samples_leaf.

ii. *max_depth:* It determines the greatest profundity of each tree, and its standard value is none. It essentially controls the maximum depth of each tree in a Random Forest. The Random Forest was 3-fold cross-validated on values ranging from 10 to 200 for max_depth. Fig. 9 shows that when max_depth is 75, then the cross-validation score is above 80%, whilst the training score is above 85%. Although we may select a larger value that gives us

the maximum accuracy while training the Random Forest, this may however lead to overfitting of the training data and might result in a model that is not suitable for certain purposes.

iii. *min_samples_leaf:* It determines how many samples are required at each leaf node, and its standard value is one. The cross-validation curves in Fig. 10 suggest that the standard value of one is the best choice.

iv. *min_samples_split:* It signifies the lowest possible number of samples that are essentially needed to separate an internal terminal node, and its standard value is two. The cross-validation curve in Fig. 11 shows that the standard value of two is the most appropriate value for this parameter. We will have more generic terminal nodes if we choose bigger values for the minimal number of samples that are necessary before an internal node splits, which will affect the overall accuracy.

The suggested values of all the above parameters when tuned into the Random Forest show us that its performance based on the evaluating metrics in Table IX decreased even though we have cross-validated the model at the expense of a large amount of validating time spent for the purpose. Although we have obtained a deeper insight into tuning the hyperparameters, they can be used to carry out more exhausting hyperparameter tuning methods like GridSearchCV. It would be more useful to do feature engineering and use the default parameters of the Random Forest for better performance.

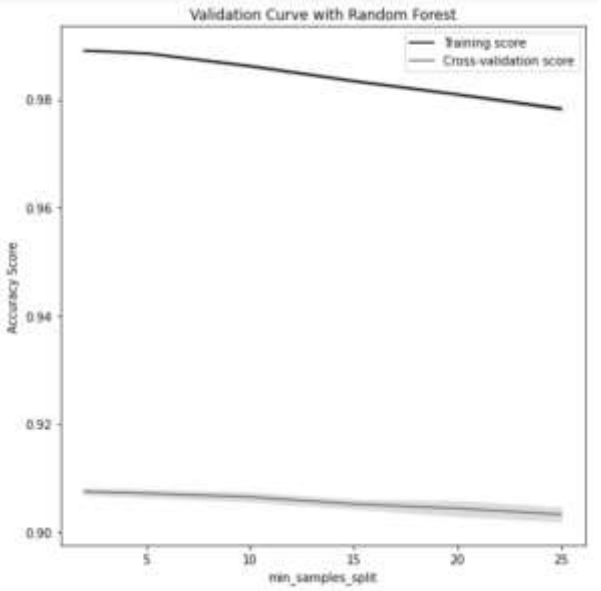

Fig. 11. Distribution of score and min_samples_split.

TABLE IX. METRICS FOR RANDOM FOREST CLASSIFIER WITH THE SUGGESTED VALUES OF HYPERPARAMETERS BY 3-FOLD CROSS-VALIDATION

Label	Precision	Recall	F1-Score	Accuracy
0	0.93	0.68	0.78	86.35 %
1	1.00	0.32	0.48	
2	0.84	0.91	0.91	

8. Conclusion and Future Scope

Reviews are a vital part of an e-commerce platform and are responsible for determining the overall product. With the escalating advancement in technology, the need to understand the sentiments expressed through a review has become highly essential. For our research, we used the TF-IDF feature extraction technique and utilized five classifiers, viz. Multinomial Naïve Bayes, Support Vector Machine, Logistic Regression, Decision Trees, and Random Forest to

carry out the sentimental analysis and classification of Amazon phone reviews and then later did their comparison. And for evaluating the results, we used a confusion matrix and applied four evaluating metrics, namely precision, recall, f1-score, and accuracy. The comparative analysis showed that the Random Forest gave the optimal outcomes with an accuracy of 92.33%. Though the Multinomial Naïve Bayes classifier showed the least accuracy, it can still be used with a smaller dataset as it had an accuracy of 85.19%, and it will work well with a lower number of reviews. Decision Trees were also found to be effective, and they can be utilized in certain cases as they had an accuracy of over 88%. In pursuit of getting a classifier that has a greater degree of optimal performance, the Random Forest was further assessed as it gave the best result with the methodology that was followed. And this was done by tuning parameters of our model with methods like out-of-bag error and 3-fold cross-validation. With the former technique, the accuracy was enhanced to 92.55%, but it was reduced to 86.35% with the latter technique. Therefore, a Random Forest tuned with the help of out-of-bag error can be used as a primary method to perform sentimental analysis and classification of the reviews to reach a decision. For the immediate future, we intend to use additional classifiers and a lexicon-dependent technique with different feature extraction techniques, viz. bag-of-words and word2vec, to get a greater outlook and thereby produce an enhanced model with better accuracy. Apart from the above-mentioned intended elements, such as the unsupervised learning approach for future work, we also aim to enhance the performance of the best-selected model from our comparative analysis approach that has been carried out so far. Also, we may work on reviews with emoji and a larger dataset at the same time, to generate an efficient version of the current model. And we additionally aim to work with reviews in different languages to have a bigger scope of reaching out to a wider audience.

References

[1]. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis", Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 347–354, 2005. Available at: https://dl.acm.org/doi/10.3115/1220575.1220619.

[2]. Zhu Zhang, "Weighing stars: Aggregating online product reviews for intelligent e-commerce applications", IEEE Intelligent Systems, vol. 23, no. 5, pp. 42–49, 2008. Available at: 10.1109/MIS.2008.95.

[3]. Callen Rain, "Sentiment Analysis in Amazon Reviews Using Probabilistic Machine Learning", Swarthmore College, Department of Computer Science, 2013. Available at: https://www.sccs.swarthmore.edu/users/15/crain1/files/NLP_Final_Project.pdf.

[4]. Kranti Ghag and Ketan Shah, "Comparative analysis of the techniques for Sentiment Analysis", International Conference on Advances in Technology and Engineering, no. 124, pp. 1–7, 2013. Available at: 10.1109/ICAdTE.2013.6524752.

[5]. Xing Fang and Justin Zhan, "Sentiment analysis using product review data", Journal of Big Data, vol. 2, no. 1, pp. 5, 2015. Available at: https://doi.org/10.1186/s40537-015-0015-2.

[6]. Muhammad T. Khan, M. Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid and Kamran H. Khan, "Sentiment analysis and the complex natural language", Complex Adapt Syst Model, pp. 1–19, 2016. Available at: https://doi.org/10.1186/s40294-016-0016-9.

[7]. Mohan Kamal Hassan, Sana Prasanth Shakthi, Sasikala Ra, "Sentiment analysis of Amazon reviews using naïve bayes on laptop products with MongoDB and R", IOP Conference Series Materials Science and Engineering, vol. 263, pp. 1–10, 2017. Available at: 10.1088/1757-899X/263/4/042090.

[8]. Heidi Nguyen, Aravind Veluchamy, Mamadou Diop, Rashed Iqbal, "Comparative Study of Sentiment Analysis with Product Reviews Using Machine Learning and Lexicon-Based Approaches," SMU Data Science Review, vol. 1, no. 4, Article 7, 2018. Available at: https://scholar.smu.edu/datasciencereview/vol1/iss4/7/.

[9]. Abhilasha Tyagi, Naresh Sharma, “Sentiment Analysis using Logistic Regression and Effective Word Score Heuristic,” International Journal of Engineering and Technology(UAE), vol. 7, no. 2.24, pp. 20–23, 2018. Available at: 10.14419/ijet.v7i2.24.11991.
[10]. Wanliang Tan, Xinyu Wang, and Xinyu Xu, “Sentiment Analysis for Amazon Reviews,” International Conference on Human and AI interaction, 2018. Available at: http://cs229.stanford.edu/proj2018/report/122.pdf.
[11]. Momina Shaheen, Shahid M. Awan, Nisar Hussain, Zaheer A. Gondal, "Sentiment Analysis on Mobile Phone Reviews Using Supervised Learning Techniques,” International Journal of Modern Education and Computer Science(IJMECS), vol. 11, no. 7, pp. 32–43, 2019. Available at: http://www.mecs-press.org/ijmecs/ijmecs-v11-n7/IJMECS-V11-N7-4.pdf.
[12]. Sara A. Aljuhani, Norah S. Alghamdi, “A Comparison of Sentiment Analysis Methods on Amazon Reviews of Mobile Phones,” International Journal of Advanced Computer Science and Applications, vol. 10, no. 6, pp. 608–617, 2019. Available at: 10.14569/IJACSA.2019.0100678.
[13]. Jayakumar Sadhasivam, Ramesh B. Kalivaradhan, “Sentiment Analysis of Amazon Products Using Ensemble Machine Learning Algorithm,” International Journal of Mathematical, Engineering and Management Sciences, vol. 4, no. 2, pp. 508–520, 2019. Available at: 10.33889/IJMEMS.2019.4.2-041.
[14]. Hui Zhang, “Sentiment Analysis on Amazon reviews,” pp. 1–13, 2019. Available at: 10.13140/RG.2.2.31090.53447.
[15]. Emilie Coyne, Jim Smit, Levent Güner, “Sentiment analysis for Amazon.com reviews,” pp. 1–9, 2019. Available at: 10.13140/RG.2.2.13939.37920.
[16]. Vineet Jain and Mayur Kambli, ”Amazon Product Reviews: Sentiment Analysis”, 2020. Available at: https://www.researchgate.net/publication/344677952_Amazon_Product_Reviews_Sentiment_Analysis.
[17]. K. Ashok Kumar, C. Jagadeesh, Pravin Kshirsagar, Swagat. M. Marve, “Sentiment Analysis of Amazon Product Reviews using Machine Learning,” Test Engineering and Management, vol. 82, pp. 5245–5254, 2020. Available at: http://www.testmagzine.biz/index.php/testmagzine/article/view/1670/1505.
[18]. Shuo Xu, Yan Li and Wang Zheng, “Bayesian Multinomial Naïve Bayes Classifier to Text Classification,” International Conference on Multimedia and Ubiquitous Engineering International Conference on Future Information Technology, pp. 347–352, 2017. Available at: 10.1007/978-981-10-5041-1_57.
[19]. Konstantinas Korovkinas, Gintautas Garšva, “Selection of Intelligent Algorithms for Sentiment Classification Method Creation,” International Conference on Information Technologies, pp. 152–157, 2018. Available at: http://ceur-ws.org/Vol-2145/p26.pdf.
[20]. Richard A. Berk, “Support Vector Machines, In: Statistical Learning from a Regression Perspective,” Springer Texts in Statistics, Springer, Cham, 2016. Available at: https://doi.org/10.1007/978-3-319-44048-4_7.
[21]. Chao-Ying Joanne Peng, Kuk Lida Lee, and Gary M. Ingersoll, “An introduction to logistic regression analysis and reporting,” The Journal of Educational Research, vol. 96, no. 1, pp. 3–14, 2002. Available at: 10.1080/00220670209598786.
[22]. Edgar C. Merkle and Victoria A. Shaffer, “Binary recursive partitioning: Background, methods, and application to psychology,” The British journal of mathematical and statistical psychology, vol. 64, pp. 161–81, 2011. Available at: 10.1348/000711010X503129.
[23]. Jehad Ali, Rehanullah Khan, Nasir Ahmad, Imran Maqsood, “Random Forests and Decision Trees,” International Journal of Computer Science Issues, vol. 9, pp. 272–278, 2012. Available at: https://ijcsi.org/papers/IJCSI-9-5-3-272-278.pdf.

[24]. Shahzad Qaiser and Ramsha Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” International Journal of Computer Applications, vol. 181, no. 1, pp. 25–29, 2018. Available at: 10.5120/ijca2018917395.
[25]. Q. Liu, J. Wang, D. Zhang, Y. Yang and N. Wang, “Text Features Extraction based on TF-IDF Associating Semantic,” IEEE 4th International Conference on Computer and Communications, pp. 2338–2343, 2018. Available at: 10.1109/CompComm.2018.8780663.
[26]. Ravinder Ahuja, Aakarsha Chuga, Shruti Kohlia, Shaurya Gupta and Pratyush Ahuja, “The Impact of Features Extraction on the Sentiment Analysis,” International Conference on Pervasive Computing Advances and Applications, vol. 152, pp. 341–348, 2019. Available at: 10.1016/j.procs.2019.05.008.
[27]. Cyril Goutte and Eric Gaussier, “A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation,” Lecture Notes in Computer Science, vol. 3408, pp. 345–359, 2005. Available at: https://doi.org/10.1007/978-3-540-31865-1_25.
[28]. Mohammad Hossin and Sulaiman M.N, “A review on evaluation metrics for data classification evaluations,” International Journal of Data Mining and Knowledge Management Process, vol. 5, no. 2, pp. 1–11, 2015. Available at: 10.5121/ijdkp.2015.5201.
[29]. Leo Breiman, “Random Forests,” Machine Learning, Springer, vol. 45, no. 1, pp. 5–32, 2001. Available at: 10.1023/A:1010933404324.
[30]. Trevor Hastie, Robert Tibshirani, and Jerome Friedman, “Random Forests, In: The Elements of Statistical Learning,” Springer Series in Statistics New York, NY, USA, vol. 1, pp. 587–604, 2009. Available at: https://doi.org/10.1007/978-0-387-84858-7_15.
[31]. Fabian Pedregosa et al., “Scikit-learn: Machine Learning in Python,” JMLR, vol. 12, no. 85, pp. 2825–2830, 2011. Available at: https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf.
[32]. Daniel Berrar, “Cross-Validation in Encyclopedia of Bioinformatics and Computational Biology,” Elsevier, pp. 542–545, 2018. Available at: 10.1016/B978-0-12-809633-8.20349-X.
[33]. Philipp Probst, Marvin Wright and Anne-Laure Boulesteix, “Hyperparameters and Tuning Strategies for Random Forest”, WIREs Data Mining and Knowledge Discovery, vol. 9, no. 3, 2019. Available at: https://doi.org/10.1002/widm.1301.