INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

A Hybrid of Proposed Filtration and Feature Selections to Enhance the Model Performance

E Sujatha[1]*, **R Radha**[2]

1 Research Scholar, Research Dept of Computer Science, SDNBV College for Women, University of Madras, Chrompet, Chennai, 600 044, India
2 Associate Professor, Research Dept of Computer Science, SDNBV College for Women, Chrompet, Chennai, 600 044, India

OPEN ACCESS

Received: 07.12.2020
Accepted: 05.07.2021
Published: 15.07.2021

Citation: Sujatha E, Radha R (2021) A Hybrid of Proposed Filtration and Feature Selections to Enhance the Model Performance . Indian Journal of Science and Technology 14(24): 2039-2050. https://doi.org/10.17485/IJST/v14i24.2017

* **Corresponding author**.

sujatha.ravi19@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: Toextract and identify the subjective information of social media user from the unstructured data. To overcome the high dimensionality and sparsity those are the two major challenges in sentiment analysis of text datasets. To increase the model performance by using possibly minimum feature sets in a text classification problem. **Methods:** We proposed a new filtration method which is applied for the removal of correlated features and zero importance features in addition to the various feature selection methods. The various feature selections such as Mutual Info, Lasso, Recursive Feature Elimination and dimensionality reduction, Principal Component Analysis (PCA) have been used along with the proposed filtration to find the compelling features. This approach was evaluated using three Indian Government Schemes and these tweets were classified using Random Forest classifier. The performance was evaluated using various metrics such as accuracy, precision, recall, f1_score, log loss and roc-auc. **Findings:** In this research, we proposed a model for selecting relevant and non-correlated feature subsets from the unstructured dataset. From this model, accuracy of 92% with the minimum log loss 0.22 was achieved through the minimum number of feature set. **Improvements:** This study proves that the performance of the model will be improved by overcoming those two problems (dimensionality and sparsity). Here various feature selection methods have been applied with the proposed filtration in order to minimize the number of features. The computing time and the model performance will be improved as a result of decreasing the features. And this will be more effective in case of large datasets. Even though Random Forest performs well in high dimensional datasets we need some more optimization.

Keywords: Mutual Information (MI); Lasso (L1); Recursive Feature Elimination (RFE); Random Forest (RF); Principal Component Analysis (PCA)

1 Introduction

According to Digital 2020 Global Overview Report on January 2020, nearly 60% of world's population is already active in social media and this will increase more than half of the world's population by the middle of this year. Between July and September 2020, more than 180 million people started using social media equating to an average of almost 2 million new users every day. The latest data indicates that more than two-thirds (68%) of world's population are using social media. Using social media people share their opinions every day about different issues such as events, persons, products, services, politics etc.,. So sentiment analysis in social media plays a vital role in monitoring of public opinion on certain topics. Sentiment analysis has various challenges in which high dimensionality and sparsity are the two.

Various Feature Selection Methods are experimented extensively in order to overcome these two problems by selecting the best set of features. Then different machine learning algorithms are used to train the selected features. The proposed method is effective by reducing the irrelevant features so that it suits for classification of high dimensional data. By reducing the feature sets, the time taken to classify the datasets also reduces without affecting the model performances. The two feature selection methods PCA and Random Forest were compared by using three classifiers namely CART, Naïve Bayes and LVQ. The proposed model was evaluated using movie reviews on Twitter Data. The Random Forest feature selection works better than PCA. It was found that Random Forest based feature selection with LVQ improves the accuracy, precision and recall. The best of 81.25% of accuracy, 81.45% of precision and 81.25% of recall were obtained [(1)]. The author adopted a deep learning approach to perform the aspect based sentiment analysis of demonetization tweets. With the help of polarity score computation and Word2vec the features were extracted and Fire Fly-oriented Multi-Verse Optimizer (FF-MVO) algorithm was used to optimize the weight of the polarity scores. The RNN classifier was used to classify the tweets as positive or negative. Finally, the comparative analysis of different machine learning algorithms proves the competent performance of the proposed model. The proposed model FF-MVO-RNN achieves the better results of accuracy 89%, precision 98%, F1-score 94% and recall 89% [(2)].

A comparative analysis of PCA, Chi2 and an ensemble of PCA and Chi2 feature selections was done to find the better outcome. The Amazon Fine Food Reviews dataset was analysed using Bernoulli Naïve Bayes, Logistics regression and Decision Tree classifiers. From the result, it was found that Chi2 works better than other two feature selections. Logistic Regression with Chi2 gained maximum accuracy of 85% [(3)]. The sentiment analysis was performed on two Amazon product reviews i.e. the electronics and automobile datasets. The three kinds of feature selections were done such as baseline features (BF), sentence level features (SLF), and a combination of sentence level features with domain sensitive features (SLF + DSF). For each case, precision, recall and F-measures were calculated as performance metrics. From the comparative analysis, SLF + DSF yield the better performance of 82.5% precision, 85.4% recall and 83.1% f-measure [(4)].

The authors developed a novel feature combination scheme to classify the sentiments of tweets which improves the accuracy with the better time efficiency. They implemented the proposed system using six popular machine learning algorithms. Among the various classifiers Naïve Bayes Multinomial classifier obtains the higher accuracy of 84.60% [(5)].

A hybrid of ensemble learning model was developed to classify the sentiments of twitter data. In this proposed model, a hybrid of Information Gain and Chi2 was used as a feature selection method. An ensemble of Ada Boost with SMO-SVM and Logistic Regression was implemented to classify the tweets. The proposed model obtains the better accuracy of 88.2% with a low error rate [(6)]. The proposed model uses an ensemble of stacked supervised algorithms and dictionary classifier along with RF and GLM Meta classifiers. To build the stacked ensemble classifier SVM, KNN and C5.0 classifiers were used. The result shows that RF Meta classifier perform better than GLM Meta classifier by achieving the highest accuracy of 90.66% for five-fold cross validation and 91.25% accuracy for ten-fold cross validation [(7)].

IWD based feature selection method was used to select the features from Turkish Twitter dataset. The performance of the selection was evaluated by using ME classifier. For all the features the best of 69.1% f-score was obtained. The proposed model achieves the highest f-score of 72.1% for 250 features [(8)]. Hybrid of three feature selections such as Information Gain, Chi Square and Gini Index was used to perform the sentiment analysis on various online reviews. SMO, MNB, RF and LR were the four classifiers used to classify the reviews. The effectiveness of the model was based on precision, recall, f-score and ROC-AUC. The proposed model achieves the highest of 92.5% precision, 86.1% recall and 92.31% f-score by using SMO classifier. And the highest of 94% of AUC was achieved by SMO classifier [(9)].

Based on the class-wise information, a novel feature selection method was proposed. The sum of each feature frequency which corresponds to a class was calculated. Then eliminate the low weighted features by using threshold value. From the result, it shows that for all 1586 features KNN obtains the highest accuracy of 85%. For 220 features the proposed method obtains high accuracy of 90% by using RF [(10)].

Various feature selection methods with ensemble classifiers were analysed to improve the model performance. Bagging and Random Subspace are the two ensemble techniques applied on classifiers LR, SVM, DT and NB to enhance the performance. Also compare these techniques with the varied neural networks. From the result, it shows that feature selection with the

ensemble classifiers outperforms neural networks. The best of 87.25% accuracy and 87.46% f1-score were obtained from the combinations of Count Difference feature selection with an ensemble of LR + Bagging and Count Difference feature selection with an ensemble of LR + Random Space[11]. Online reviews were analysed by using four classifiers NB, KNN, ME, and SVM with the combination of three ensemble methods such as Boosting, Bagging, and Random subspace. It was found that 88.95% accuracy achieved by using stand alone ME. The highest accuracy of 88.68% was obtained from an ensemble method, SVM+RS[12].

The impact of two feature extractors TF-IDF and Bi-grams were analysed on SS-Tweet dataset. Six different algorithms Decision Tree, SVM, KNN, Naïve Bayes, Random Forest and Logistic Regression were used to classify the sentiments as positive, negative and neutral. The result shows that TF-IDF performs 3-4% better than Bi-grams. The best of 57% accuracy, 57% precision, 50% recall and 50% f-score were obtained by using LR[13]. The tweets of Indian Railways were analysed to find the sentiments such as positive, negative and neutral. The model was evaluated in terms of accuracy, precision, recall and f1-score by using four classifiers C4.5, Naïve Bayes, SVM and Random Forest. The highest of 91.5% accuracy, 88.5% precision, 87.5% f1-score and 83% of recall were obtained by using SVM. It was found that SVM performs better than Random Forest, Naïve Bayes and C4.5[14]. The tweets from the three digital payment service providers of Indonesia were gathered to perform the sentiment analysis. KNN and NB classifiers were used to classify the tweets. From the result, it was found that KNN performs better than NB by getting the accuracy of 91%[15].

The authors developed a model to determine the proteins that belongs to which molecular function of electron transport proteins. The performances from PSSM with AA Index feature set were successful in identifying electron transport proteins in transport proteins with achieved sensitivity of 73.2%, specificity of 94.1%, and accuracy of 91.3%, with MCC of 0.64 for independent data set[16].

The authors developed a model to observe the impact of using feature selection on tweet sentiment classification. They experimented by using the combinations of four classifiers, ten feature rankers and ten feature subset sizes: 5, 10, 15, 20, 25, 50, 75, 100, 150, and 200 out of 2388 features available from the dataset of 3000 tweets. Generally using 200 features works best than 100 and 150 features. They proved that using feature selection to select 50 or fewer features generally results in poor performance. By performing ANOVA analysis they tested the statistical significance of their findings. It was found the performance improvement achieved by selecting 75 or more features was statistically significant[17]. A proposed model was developed to improve the performance of the classification of twitter data by using three different classifiers with three categories of feature selections. The Kern lab support vector machine with the third category of feature selection gives the best accuracy of 86.22%[18].

With the implementation of feature selection and feature weighting the accuracy has been increased by using SVM classifier. The combination of Chi2 and TFIDF has been used to improve the accuracy and system performance was evaluated by using 10 fold cross validation. As a result the accuracy improved from 68.7% to 80.2%[19]. The authors proposed a model to classify the sentiments of E-Commerce based tweets which are downloaded from the Twitter Cloud Repository. The Naïve Bayes classifier was used along with the feature selection method Information Gain to improve the model performance in terms of accuracy. The 88.80% of accuracy was obtained by using this proposed system[20]. A novel hybrid framework was developed to classify the twitter data. The model was proposed based on feature selection method local linear embedding (LLE) and three machine learning classifiers such as Random Forest, K-Nearest Neighbors and Logistic Regression. Random Forest has performed the best out of all by achieving the higher accuracy of 80%[21].

The authors proposed a model to classify the sentiments of twitter data regarding Citizenship Amendment Act 2020. The proposed system presents a faster approach of sentiment analysis using various classifiers to classify the tweets as positive, negative or neutral. For faster and accurate POS tagging VADER was used. Among the various classifiers SVM performs better which obtains the accuracy of 77.32%[22].

Table 1 given below compares the previous methods with the proposed method. Figure 1 shows the architecture diagram.

Table 1. Comparison of previous methods with proposed method

Reference	Year Published	FS methods	Classifiers	Accuracy %	Precision %	Recall %	F-Score %	ROC-AUC %
[5]	2015	Unigram + IG	NBM	84.60	-	-	-	-
[1]	2016	RF	LVQ	81.25	81.45	81.25	-	-
[16]	2017	AAIndex	PSSM	91.3	-	73.2	-	-
[3]	2018	Chi2	LR	85	-	-	-	-
[10]	2018	For all 1586 features	KNN	85	-	-	-	-

Continued on next page

Table 1 continued

		Class-wise + threshold (For 220)	RF	90	-	-	-	-
(9)	2018	IG + Chi2 + GI	SMO	-	92.5	86.1	92.31	94
(4)	2019	SLF + DSF	Random Tree	-	82.5	85.4	83.1	-
(8)	2019	For all features	ME	-	-	-	69.1	-
		IWD based (For 250)		-	-	-	72.1	-
(11)	2019	CD	LR + Bag, LR + RS	87.25	-	-	87.46	-
(19)	2019	Chi2 + TFIDF	SVM	80.2	-	-	-	-
(6)	2020	IG + Chi2	Ada Boost + SMO SMV + LR	88.2	-	-	-	-
(20)	2020	IG		88.8	-	-	-	-
(2)	2021	FF-MVO	RNN	89	98	89	94	-
(21)	2021	LLE	RF	80	-	-	-	-
Proposed work	-	Proposed filtration + Lasso, MI, RFE(RF)	Random Forest	92	92	92	92	99

2 Architecture Diagram

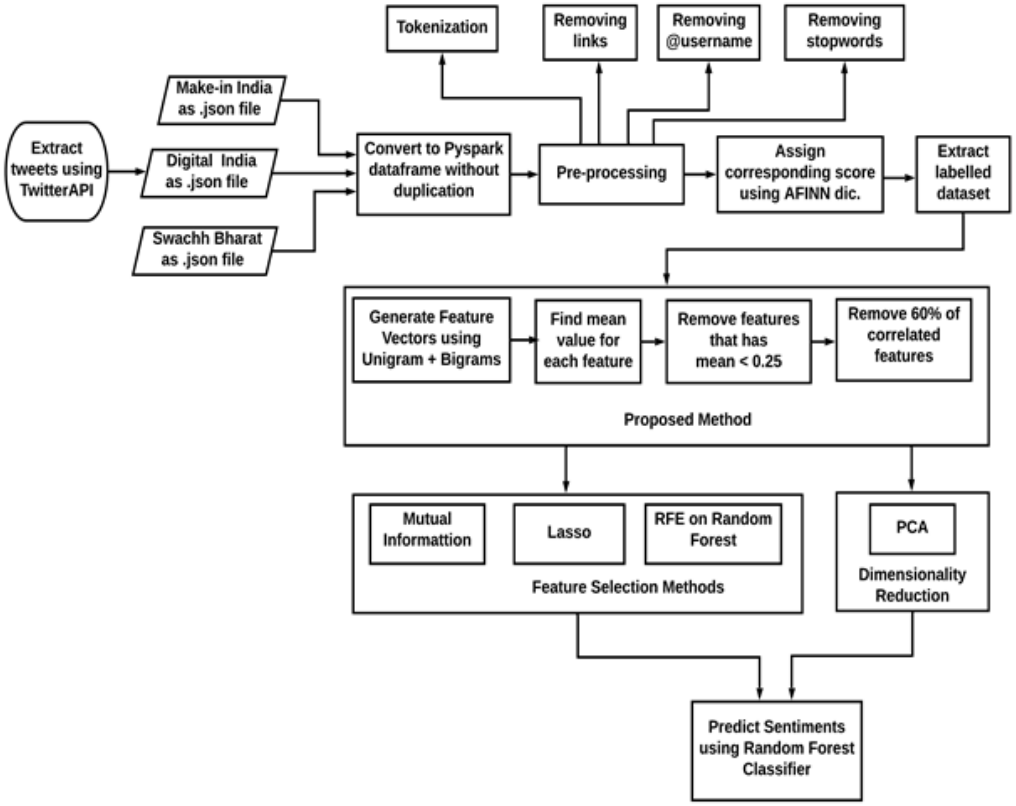

Fig 1. Architecture Diagram of Proposed Model

3 Proposed Model

The proposed model has been described in two phases.

3.1 Phase 1

The sentiment analysis has been processed in five steps. They are:

1. Tweets extraction: By using Twitter API, the tweets were collected. The analysis has been done by using three schemes of Indian Government such as Make-in India, Digital India and Swachh Bharat. About 7500 tweets were extracted regarding the schemes.
2. Pre-processing: It includes to kenization, removing links, stop words, unwanted symbols and characters. These have been done by using Pyspark package.
3. Tokens evaluation: The tokens have been assigned the corresponding ratings by using AFINN Dictionary. This dictionary contains English words associated with polarity score between -5 and 5. So, label each tweets by using these scores.
4. Feature Extraction: By using Hashing TF-IDF, the feature vectors were generated. In this model, the features were extracted based on the three feature sets such as Unigram, Bigram and Unigram + Bigram. When using text as the features, the Hashing TF-IDF improves the performance.
5. Classification: After pre-processing 7336 tweets were obtained this includes 7519 features. Before classification split the samples as Train set and Test set in the ratio 7: 3. Random Forest classifier was used to analyse the sentiments as positive, negative and neutral. The comparative analysis has been done between Unigram feature vectors, Bigram feature vectors and Unigram + Bigram feature vectors. The below algorithm, Algorithm1 has been used for extracting and classifying the tweets.

Algorithm1:

Input: Extracted tweets in JSON files.

Output: Classify the tweets as negative, positive and neutral.

1. The tweets from JSON files should be converted to Pyspark data frame.
2. The duplicate records in the data frame have been removed by using User_id.
3. By using Pyspark package, the tweets were extracted from the data frame and pre-processed.
4. The AFINN dictionary was used to label the tweets by its corresponding polarity scores.
5. Each token were organised as the combination of unigram and bigrams.
6. Using Hashing TF and IDF method the feature vectors were extracted based on three feature sets.
7. Taking the 70% of data as train set and 30% as test set.
8. The Random Forest classifier was used to predict the sentiments.

In general, when the number of trees increases in Random Forest the better results will be obtained. However, it will decrease the improvements for more number of trees.[23] It has been proved that between 64 and 128 numbers of trees in a forest will obtain a good balance between AUC, processing time, and memory usage. So, this model has been experimented for 10 trees and 100 trees.

By comparing the tables Tables 2, 3 and 4 given below, the combination of Unigram and Bigram for 100 trees with 160 features performs better than as they perform individually for 80 features. This achieves the highest of 92% accuracy, 92% precision, 92% recall, 92% f1-score and 99% of ROC-AUC with the minimum log loss 0.22.

As the number of features increases, the performance of the model was decreased by 1% of accuracy, precision, recall and f1-score. This result has been shown in Table 5 for the 400 features. So, the minimum number of 160 features with combination of unigram and Bigram has been taken for the next filtration method.

Table 2. Unigram, Feature size = 80

Metrics	Number of Trees	
	10	100
Accuracy	0.890	0.910
Log Loss	0.358	0.225
Precision	0.892	0.914
Recall	0.890	0.910
F1-Score	0.890	0.913
ROC-AUC	0.98	0.99

Table 3. Bigram, Feature size = 80

Metrics	Number of Trees	
	10	100
Accuracy	0.887	0.900
Log Loss	0.505	0.231
Precision	0.890	0.902
Recall	0.891	0.903
F1-Score	0.894	0.902
ROC-AUC	0.98	0.98

Table 4. Unigram + Bigram, Feature size = 160

Metrics	Number of Trees	
	10	100
Accuracy	0.901	0.916
Log Loss	0.341	0.223
Precision	0 .902	0.922
Recall	0.901	0.920
F1-Score	0.903	0.924
ROC-AUC	0.98	0.99

Table 5. Unigram + Bigram, Feature size = 400

Metrics	Number of Trees	
	10	100
Accuracy	0.901	0.908
Log Loss	0.368	0.223
Precision	0.904	0.914
Recall	0.901	0.908
F1-Score	0.901	0.909
ROC-AUC	0.98	0.99

3.2 Phase 2

In this phase, the three different feature selection methods have been applied for 160 features to select the relevant features. Before applying the feature selection methods, a filtration has been done. This filtration consists of two steps: first removes the features whose mean value is less than a threshold and in the second step, the correlated features have been filtered. The Algorithm2 given below describes these two filtration steps.

Algorithm2:

Input: Dataframe df, that has each features as columns.

Output: Returns a data frame in which zero importance features were removed.

1. Find the mean value for each and every 160 columns and save in a list j.
2. To find the total number of removed features, assign sum = 0.
3. For each item i in the list j,
   (a) Check the mean value of every column, j[i]
   (b) If j[i] is less than 0.25 ,
      i. Remove the column i from the data frame df.
      ii. Increment sum by 1.
4. Then find the correlation between columns and remove the correlated columns.
5. Convert to 137 x 137 matrix of the pair wise correlation coefficient between each pair of columns and save the matrix in a data frame cormat.

6. For each item i in cormat.columns,

   (a) For each item j in i,

      i. Find the absolute correlation coefficient between the pair cormat [i, j].
      ii. Remove the features whose correlated value is greater than 60%.

After step3, totally 23 columns have been removed. So (160 – 23 = 137) we are finding the correlation coefficient between each pair of 137 columns.

After step6, it was found that the again 23 columns are highly correlated for more than 60%. So the remaining 114 columns (137 – 23 = 114) has been undertaken for further feature selection process by using various feature selection methods.

Using feature selection methods such as Mutual Information (MI), Lasso (L1) and Recursive Feature Elimination, RFE (RF) with this proposed filtration method yields good result. Table 6 depicts the model performance by comparing the results with before and after filtration method. There will be only a slight difference occurs when we train the classifier for 160 features without filtration and for 114 features with filtration. Only 0.003% of f1-score, 0.002% of accuracy and 0.002% of recall have been decreased in 114 feature set. But there is an improvement in the log loss where 0.22 has been decreased to 0.21. So, the eliminated features may not be more important for the further classification process.

Table 6. Comparison of model performance before and after filtration

Filtration Methods	Number of Features	Accuracy	Log Loss	Precision	Recall	F1-Score	ROC-AUC
Without Filtration	160	0.916	0.223	0.920	0.916	0.917	0.99
Mean Value < 0.25	137	0.913	0.225	0.917	0.913	0.913	0.99
Correlation Coefficient > 60%	114	0.914	0.210	0.919	0.914	0.914	0.99

Table 7 given shows the performance of the model by applying the feature selections without filtration for 160 feature set. Here the model has been trained for least number of features to obtain the better results. This number of least feature varies between the feature selections. Selecting the 112 features in MI, 93 feature in L1 and 95 features in RFE (RF) produces the maximum results. When we further decreases these numbers in each method will affect the performance of model.

Table 7. Applying FS methods before Filtration (for 160 features)

Feature Selection Methods	Number of Features	Accuracy	Log Loss	Precision	Recall	F1-Score	ROC-AUC	Time Taken(in ms)
MI (70%)	112	0.917	0.229	0.921	0.917	0.918	0.99	908
LI (0. 013)	93	0.909	0.217	0.913	0.909	0.909	0.99	902
RFE(RF)	95	0.910	0.210	0.913	0.910	0.910	0.99	914

Table 8 shows the results of three feature selection methods that are applied after the proposed filtration method. By comparing Tables 7 and 8 , the model performance has been increased with the decreased number of features in each selection method. The better results have been produced by selecting 80 features instead of 112 in MI, 86 features instead of 93 in L1 and 51 features instead of 95 in RFE (RF). 1% of accuracy, precision, recall and f1-score have been increased in L1 method after filtration. MI and RFE (RF) produce the same result with the decreased number of features after filtration. As the number of features decreases the time taken for classification process also decreases. RFE (RF) takes the minimum features of 51 and thus the time taken will be reduced to 811 milliseconds.

From theTable 9, we understand how the filtration helps in achieving better results with minimum number of features. The numbers of features selected by the each selection method after filtration have been applied for 160 features. The performance decreases significantly by further reducing the features. By comparing the Tables 8 and 9 , the numbers of features are same for the each selection method but the result varies because of filtration process done in Table8 before applying selection methods.

Thus, the proposed method is effective in reducing the number of features to obtain better performance.

Table 8. Applying FS methods after Filtration (for 114 features)

Feature Selection Methods	Number of Features	Accuracy	Log Loss	Precision	Recall	F1-Score	ROC-AUC	Time Taken(in ms)
MI (70%)	80	0.915	0.214	0.918	0.915	0.915	0.99	888
LI (0. 02)	86	0.918	0.215	0.921	0.918	0.918	0.99	880
RFE(RF)	51	0.907	0.218	0.910	0.907	0.907	0.99	811

Table 9. Applying FS methodswithout Filtration (as same number of features after filtration)

Feature Selection Methods	Number of Features	Accuracy	Log Loss	Precision	Recall	F1-Score	Time Taken(in ms)
MI	80 (50%)	0.903	0.251	0.907	0.903	0.904	897
LI	87 (0.011)	0.909	0.245	0.913	0.909	0.909	882
	84 (0. 01)	0.911	0.244	0.914	0.911	0.911	876
RFE(RF)	51	0.902	0.265	0.905	0.902	0.902	834

Table 10 shows the result that the model performance increases when filtration is applied before the dimensionality reduction. There is no difference between 160 feature set and 114 feature set for zero components in PCA except the time taken. But with filtration, 1% of accuracy, precision, recall and f1_score have been increased for ten components of PCA.

Table 10. Applying Dimensionality Reduction PCA

Number of Components	For 160 Features (before Feature Filtration)		For 114 Features (after Feature Filtration)	
	PCA (0)	PCA (15)	PCA (0)	PCA (10)
Accuracy	0.914	0.907	0.905	0.918
Log Loss	0.219	0.220	0.220	0.216
Precision	0.918	0.911	0.910	0.921
Recall	0.914	0.907	0.905	0.918
F1_Score	0.914	0.908	0.905	0.918
ROC_AUC	0.99	0.99	0.99	0.99
Time Taken(in ms)	1690	896	1430	884

4 Performance Measures

Evaluating a model is a core part of building an effective machine learning model. To demonstrate the performance of the proposed method some metrics have been evaluated which are as follows,

- Accuracy: It represents the number of correctly predicted data points out of all the data points.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

- Precision: It is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision\ or\ Positive\ Predictive\ Value = \frac{TP}{TP+FP}$$

- Recall: It is the ratio between the numbers of positive observations correctly classified as positive to the total number of positive observations.

$$Recall\ or\ Sensitivity = \frac{TP}{TP+FN}$$

Where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

- F1-Score: It is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. The F-score has been widely used in the natural language processing literature.

$$F1\_Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

- Roc-Auc: The ROC curve is the plot between sensitivity and (1- specificity). (1- specificity) is also known as false positive rate and sensitivity is also known as True Positive rate. AUC itself is the ratio under the curve and the total area. It considers the predicted probabilities for determining our model's performance. It is widely used when the dataset is imbalanced since accuracy is not a reliable performance metric for imbalanced data.
- Log Loss: It is indicative of how close the prediction probability is to the corresponding actual/true value. It is useful to find out the performances of the model since from accuracy we cannot measure how good the predictions of the model are.

$$Log\ Loss = \frac{-1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij} * log(P_{ij})$$

Where y_{ij}, indicates whether sample i belongs to class j or not p_{ij}, indicates the probability of sample i belonging to class j.

5 Results And Analysis

In this section the performance of the proposed model using real-time datasets has been discussed. The performance of the proposed model was compared with the other existing works has been represented in Table 1. The results as shown in this table prove that the proposed model yields the better results than the other classification systems. In addition to the various metrics as evaluated in present works, log loss was also analysed in the proposed work. Moreover, the various combinations of feature size and feature sets such as unigram, bi-grams have been experimented in phase1 to find out the optimal solution. The improvement of proposed model was also analysed in terms of computational time for every feature selection.

The time taken to classify the overall 160 features was 1.02 seconds. The hybrid of proposed filtration and feature selection RFE (RF) takes the minimum of 51 features with the reduced time taken of 811 milliseconds. We obtain 20% by calculating ((1020-811) * 100/1020 = 20.49)), the percentage of differences in time taken. The 20% difference will be a significant one in case of big data.

The various metrics such as accuracy, precision, recall, f1_score, log loss and roc-auc have been analysed for the proposed model. It is important to have the completeness and exactness than higher accuracy. So in addition to accuracy, the three parameters such as precision, recall and f1-score have been evaluated. To measure the model performance, auc does not bias on the test data size whereas accuracy always biased on the test data size. In this model, only 30% data has been used as test data. So, it is better to evaluate roc-auc for the proposed model. ROC-AUC also ranges between 0 and 1. A good model will have a roc-auc near to 1. The Figure 2 represents that the proposed model performs better which has the roc-auc 0.99 for each class. From Figure 3 , it understands that the proposed method has reduced the features significantly for MI and RFE (RF) methods. Even though it shows only slight difference for Lasso method; it increases accuracy from 91% to 92%.

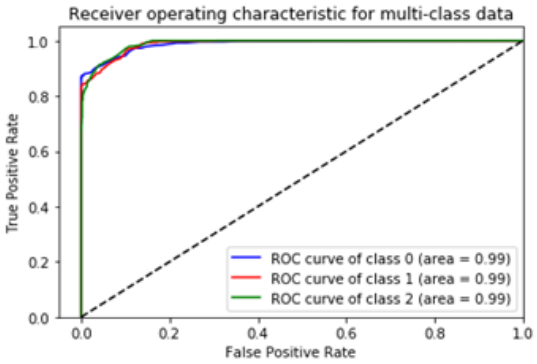

Fig 2. ROC curve for proposed model

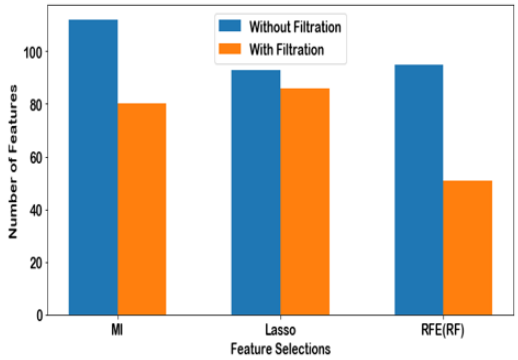

Fig 3. Comparison of number of features with/without filtration

The loss function acts as a good proxy for accuracy. The accuracy of a classifier is evaluated by log loss through penalising the false classification. Basically, minimising the log loss is equivalent to maximising the accuracy of the classifier. Log loss is a probability value between 0 and 1. A good model will have a log loss of 0. Consider the proposed model as a best model since it has the log loss 0.22 with the minimum number of features. The Figure 4 represents the differences in log loss obtained before and after filtration method with feature selections. The log loss reduced significantly for each feature selection with filtration.

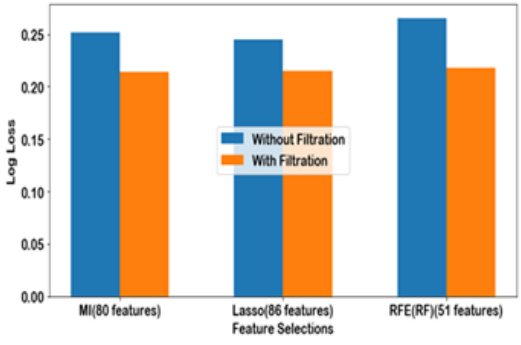

Fig 4. Comparing Log Loss with/without filtration

Finally, the sentiment classification has been performed on these tweets by using Random Forest classifier and found out the results which described in [24].

6 Discussions

Before applying the feature selection methods or dimensionality reduction method, a proposed filtration method has been used to filter the insignificant features and 60% of correlated features in order to improve the efficiency of model. Because of irrelevant and correlated features the performance of the model will be affected. Also it is proved that the proposed filtration method has reduced the features significantly and achieves the better results. After filtering the 60% of correlated features and applying tree based feature selection will produce efficient results that were proved in the proposed model. This model can also be applied for the large datasets as it is implemented in Apache PySpark framework. This model will also be more effective when size of the data/features increases. The performance of the model is evaluated based on various metrics such as accuracy, precision, recall, f1-score, log loss and roc-auc. The metrics used in the proposed system are the best suitable metrics for any text classifications. Since accuracy is not a reliable performance metric for imbalanced data roc-auc is evaluated and additionally log loss represents how close the predictions are towards the actual values. For any NLP approach f1-score is widely used.

The proposed model is not domain specific that can be implemented in various domains to classify the sentiments. Since this model is proved to be good in both time and performance with minimum feature set, it can be certainly obtain its application in the field of big data. When it comes to social data, its application can be surely extended.

7 Conclusion

In this study, a hybrid of filtration and feature selection method has been implemented in order to reduce the number of features. By using AFINN dictionary and Random Forest algorithm the multi class classification (positive, negative and neutral) has been performed. Along with, the three feature selections such as MI, L1 and RFE (RF) are used. The filtration has also been tested with dimensionality reduction method, PCA. Finally, the performance of the proposed model has been evaluated using various metrics such as accuracy, precision, recall, f1_score, log loss and roc-auc. This model achieves 92% of accuracy, precision, recall, f1_score and 99% of roc-auc with minimum log loss 0.22. From the results as shown in Table 1 , it proves that our proposed model performs better than the previous models.

An increase of 1% of accuracy is not the only advantage of proposed model, but also, we are gaining more or less same accuracy by using the minimum set of features that reduces the complexity of the model. So in case of large datasets the proposed model will be more useful by improving the model performances with the minimum set of features. With this dataset, we are getting 92% of accuracy with the minimum of 51 features instead of using 160 features i.e. we are using only one third of features to get the same or better results. As we refer[17], the authors proved that using feature selection to select 50 or fewer features generally results in poor performance. They also found that the performance improvement by selecting 75 or more features was statistically significant. From the results, we proved that 20% of computational time has been reduced which will be more effective in case of large datasets.

Future Scope

As the future work, the analysis should be improved to increase the percentages of evaluation metrics further. Another datasets that might be a large datasets have been implemented in the proposed model in order to find its performance. Need to analyse the sentiments of Tamil tweets.

Acknowledgement

The authors gratifyingly acknowledge the reviewers and editorial board for their valuable comments that led to the improvement of this manuscript.

References

1) Jotheeswaran J, Koteeswaran S. Feature Selection using Random Forest Method for Sentiment Analysis. *Indian Journal of Science and Technology*. 2016;9(3):1–7. Available from: https://dx.doi.org/10.17485/ijst/2016/v9i3/86387.
2) Datta S, Chakrabarti S. Aspect based sentiment analysis for demonetization tweets by optimized recurrent neural network using fire fly-oriented multi-verse optimizer. *Sadhana*. 2021;p. 46–79. Available from: https://doi.org/10.1007/s12046-021-01608-1.
3) Bhagat M. Sentiment Analysis using an ensemble of Feature Selection Algorithms. 2018. Available from: https://doi.org/10.31979/etd.xg3j-fty7.
4) Rintyarna BS, Sarno R, Fatichah C. Evaluating the performance of sentence level features and domain sensitive features of product reviews on supervised sentiment analysis tasks. *Journal of Big Data*. 2019;6(1):1–19. Available from: https://dx.doi.org/10.1186/s40537-019-0246-8.
5) Yang A, Zhang J, Pan L, Xiang Y. Enhanced Twitter Sentiment Analysis by Using Feature Selection and Combination. *International Symposium on Security and Privacy in Social Networks and Big Data (SocialSec)*. 2015;p. 52–57. Available from: 10.1109/SocialSec2015.9.

6) Sharma S, Jain A. Hybrid Ensemble Learning With Feature Selection for Sentiment Classification in Social Media. *International Journal of Information Retrieval Research*. 2020;10(2):40–58. Available from: https://dx.doi.org/10.4018/ijirr.2020040103.
7) Rani S, Gill NS. Hybrid Model For Twitter Data Sentiment Analysis Based On Ensemble Of Dictionary Based Classifier And Stacked Machine Learning Classifiers-Svm, Knn And C5.0. *Journal of Theoretical and Applied Information Technology*. 2020;98(04):624–635.
8) Parlar T, Sarac E. IWD Based Feature Selection Algorithm for Sentiment Analysis. *Elektronika ir Elektrotechnika*. 2019;25(1):54–58. Available from: https://dx.doi.org/10.5755/j01.eie.25.1.22736.
9) Ghosh M, Sanyal G. An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning. *Journal of Big Data*. 2018;5(1):1–25. Available from: https://dx.doi.org/10.1186/s40537-018-0152-5.
10) Kumar HMK, Harish BS. A New Feature Selection Method for Sentiment Analysis in Short Text. *Journal of Intelligent Systems*. 2018;29(1):1122–1134. Available from: https://dx.doi.org/10.1515/jisys-2018-0171.
11) Madasu A, Elango S. Efficient feature selection techniques for sentiment analysis. *Multimedia Tools and Applications*. 2020;79(9-10):6313–6335. Available from: https://dx.doi.org/10.1007/s11042-019-08409-z.
12) Arya P, Bhagat A, Nair R. Improved Performance of Machine Learning Algorithms via Ensemble Learning Methods of Sentiment Analysis. *International Journal on Emerging Technologies*. 2019;10(2):110–116.
13) Ahuja R, Chug A, Kohli S, Gupta S, Ahuja P. The Impact of Features Extraction on the Sentiment Analysis. *Procedia Computer Science*. 2019;152:341–348. Available from: https://dx.doi.org/10.1016/j.procs.2019.05.008.
14) Madhuri DK, D. A Machine Learning based Framework for Sentiment Classification: Indian Railways Case Study. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*. 2019;8(4):2278–3075.
15) Wisnu H, Afif M, Ruldevyani Y. Sentiment analysis on customer satisfaction of digital payment in Indonesia: A comparative study using KNN and Naïve Bayes. *Journal of Physics: Conference Series*. 2020;1444. Available from: https://dx.doi.org/10.1088/1742-6596/1444/1/012034.
16) Le NQK, Nguyen TTD, Ou YY. Identifying the molecular functions of electron transport proteins using radial basis function networks and biochemical properties. *Journal of Molecular Graphics and Modelling*. 2017;73:166–178. Available from: https://dx.doi.org/10.1016/j.jmgm.2017.01.003.
17) Prusa JD, Khoshgoftaar TM, Dittman DJ. Impact of Feature Selection Techniques for Tweet Sentiment Classification. *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference*. 2015;p. 299–304.
18) Payal B, Awachate, Kshirsagar PV. Improved Twitter Sentiment Analysis Using N Gram Feature Selection and Combinations. *International Journal of Advanced Research in Computer and Communication Engineering*. 2016;5(9):154–157. Available from: 10.17148/IJARCCE.2016.5935.
19) Larasati UI, Muslim MA, Arifudin R, Alamsyah A. Improve the Accuracy of Support Vector Machine Using Chi Square Statistic and Term Frequency Inverse Document Frequency on Movie Review Sentiment Analysis. *Scientific Journal of Informatics*. 2019;6(1):138–149. Available from: https://dx.doi.org/10.15294/sji.v6i1.14244.
20) Singh M, Gupta S. Sentiment Analysis using Naive Bayes Classifier and Information Gain Feature Selection over Twitter. *International Journal of Computer Trends and Technology*. 2020;68(5):84–91. Available from: https://dx.doi.org/10.14445/22312803/ijctt-v68i5p117.
21) Zahra T, Ghous H, Hussain I. Sentiment Analysis Of Twitter Dataset Using Lle And Classification Methods. *International Research Journal Of Modernization In Engineering Technology And Science*. 2021;3(1):1151–1164.
22) Narang A. Twitter Sentiment Analysis on Citizenship Amendment Act in India. *International Journal for Research in Applied Science and Engineering Technology*. 2020;8(7):1714–1724. Available from: https://dx.doi.org/10.22214/ijraset.2020.30636.
23) Oshiro TM, Perez PS, Baranauskas JA. How Many Trees in a Random Forest? *Lecture Notes in Computer Science*. 2012;p. 154–168. Available from: https://doi.org/10.1007/978-3-642-31537-4_13.
24) Sujatha E, Radha R. A Sentiment Classification on Indian Government Schemes Using PySpark. *International Journal on Emerging Technologies*. 2020;11(2):25–30.