M. Tech (Software Engineering)

SNEHA KUMAWAT

2019

Analysis and Improvement of Text Classifiers

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF DEGREE
OF

MASTER OF TECHNOLOGY
IN
SOFTWARE ENGINEERING

Submitted by
Sneha Kumawat
2K17/SWE/16

Under the supervision of

DR. RUCHIKA MALHOTRA
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road,
Delhi-110042

JUNE, 2019

Department of Computer Science and Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road Delhi-110042

CANDIDATE'S DECLARATION

I Sneha Kumawat, 2K17/SWE/16 of Master of Technology (Software Engineering) hereby declare that the Major Project-II Dissertation titled "**Analysis and Improvement of Text Classifiers"** which is submitted by me to the Department of Computer Science and Engineering, Delhi Technological University, Delhi in partial fulfillment of requirement for the award of degree of Master Of Technology (Software Engineering) is original and not copied from any source without proper citation. This work has not been previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi **Sneha Kumawat**

Date: **2K17/SWE/16**

Department of Computer Science and Engineering
Delhi Technological University
(Formerly Delhi College of Engineering)
Bawana Road Delhi-110042

CERTIFICATE

I hereby certify that the Project Dissertation titled "**Analysis and Improvement of Text Classifier** " submitted by Sneha (2K17/SWE/16) to the Department of Computer Science and Engineering, Delhi Technological University in partial fulfillment of requirement for the award of the degree of Master of Technology, is a record of project work carried out by the student under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

.

Place: Delhi

Date:

DR. RUCHIKA MALHOTRA

(Supervisor)

Associate Professor

Discipline of Software Engineering

CSE Department

Delhi Technological University

(Formerly Delhi College of Engineering)

Shahbad, Daulatpur, Bhawana Road Delhi-110042

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to follow the right path of the life. My greatest thanks is to my parents who bestowed the ability and strength in me to complete this work.

My thanks is addressed to my mentor Dr. Ruchika Malhotra, Department of Computer Science and Engineering who gave me this opportunity to work in a project under her supervision. It was her enigmatic supervision, unwavering support and expert guidance which has allowed me to complete this work in due time. I humbly take this opportunity to express my deepest gratitude to her.

Date:

Sneha Kumawat

M.Tech (SWE)-4th Sem

2K17/SWE/16

ABSTRACT

The number of textual documents are increasing at an incredible rate and very often, there is a need to classify those documents into some fixed predefined categories. The concepts of text mining and machine learning help a lot in this task of automated document classification. Since the classification is being done automatically, the classifier needs to be a good classifier so that there are as less misclassifications as possible. Therefore, the classification accuracy is very important and needs to be taken care of. There are various factors that can affect the classification accuracy of classifiers. One of the factors is the Feature Selection method used to reduce the number of features in the documents. Information Gain (IG) is one of the most popular methods employed for this task but there are few shortcomings in this method of evaluating the better words. In our thesis, we have used Term frequency inverse Document (TFID)and Bag of words (BOW) thus finding the better words which are more useful in the classification task. With these techniques we have used ensemble technique that is bagging in order to improve the classification process. We have also compared the results of both these feature selection techniques with and without ensemble learning for text classification and the results show that our method improves the average classification accuracy of a text classifier and is much more consistent in its classification accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

LR	Logistic Regression
NB	Naïve Bayes
RF	Random Forest
SVM	Support Vector Machine
BOW	Bag of Words
TFID	Term frequency inverse Document
IDF	Inverse document frequency
TF	Term frequency
IG	Information gain
ML	Machine learning
PCA	Principal component analysis
MAP	Maximum a posteriori probability

CHAPTER-1

INTRODUCTION

In today's world researchers have testified that nearly 80-85% of the information we use is not structured. The software library has countless research-related resources. It includes reports of fault, software requirement documents, and other related records. We should be able to analyze the above requirement and predict its categories such as safety, usability, accessibility, and maintenance. As they are mostly disorganized, the software files are usually hard to analyze and evaluate. It is therefore of utmost importance to put the information in a form that can be understood and evaluated by computer-aided instruments. Text Classifiers are useful for software project managers as they help in quantitative planning and management of the project. This chapter summarises the need and motivation for the study.

1.1 Overview

A typical text mining issue is to obtain appropriate data from software files such as fault descriptions or software requirement specification document kept in software repositories and analyze these descriptions using appropriate steps, instrument and methods. The quantity of textual information at the elevated rate increases every day. The document could be merely a set of words and often without any relation between words and without any semantic significance. The lengthy text descriptions usually comprise nearly all kinds of data, meaning it can be connected data, semi-structured data, numerical information, etc. Lots of information in books, libraries, etc. are unorganized and stored. Users must spend hours searching for appropriate content before finding some helpful data. This leads to wasting time and energy. A system that can categorize large quantities of present data in internet libraries and repositories of software has become very essential now. The text mining domain is used to acquire suitable information to decrease work and moment.

Feature Selection Technique i.e. TFID and BOW are used for selection of feature subset which is most relevant for the task of classification. Moreover, logistic regression (LR), Support Vector Machine (SVM), Random Forest (RF) and Naïve Bayes (NB) are used as classifiers for evaluation of the predictive accuracy of the agent. Ensemble techniques are also used with these base ML techniques in order to improve accuracy. For evaluation, three most widely used metrics i.e. Accuracy, F1_Scores and Receiver Operating Characteristic

curve Accuracy have been used. Classification accuracy alone can at times mislead and hence the other two metrics have also been considered for estimation. F1_Scores are calculated as a "weighted average of Precision and Recall taking into consideration both False Negatives and False Positives". Basically, it evaluates the harmonic mean of the above two. ROC curve analysis is considered as a complete report of specificity and sensitivity.

1.2 Research Objective

Machine learning techniques help the developers to analyze the data from different perspectives and enable them to retrieve useful information. Different machine learning techniques have been examined for prediction of severity from textual reports and comparative analysis has been performed to find out whether a classifier with which feature selection technique outperforms. In this study, we explore the two different approaches i.e. TFID and BOW for feature selection. We have also used bagging an ensemble technique to improve accuracy. Literature summarising these techniques into tabular form for quick analysis is done. The details of the method and results will be discussed in detail in later chapters.

1.3 Organization of the Thesis

In this thesis, we aim to find the best methods for the problem of the text classification. New methods are explored and compared in order to find increase the accuracy of the text classifiers.

Following this introductory section, there are a number of 6 more sections organized as follows: Chapter 2 summarises Literature Review and then Chapter 3 covers the steps of the text classification, Chapter 4 talks about the Machine learning (ML) techniques used in proposed work. Chapter 5 explains the experiment design and setup, Chapter 6 present the results and assessment of results learned after work and finally, Chapter 7 conclude the work and explain future possibility.

CHAPTER-2

LITERATURE REVIEW

The various studies related to Text Classifiers based on different feature selection have been illustrated below. In the previous studies various machine learning techniques such as Linear Classifier, Decision Tree, Artificial Neural Network and Naïve Bayes on the data sets for text classification been used and comparative performance analysis between them has been performed so as to find out classifiers with which techniques give better performance. The original research on text classification forecast focuses primarily on the use of statistical techniques. Below is a summary of the studies used in this research.

2.1 Background Work

Many software studies have investigated the severity level associated with text in software. But here we will only consider those studies that use ML techniques to predict Severity level based on text classification . The information present in different organisations is in big data's size, the data is in a large amount. Having such big volumes of information presents severe threats to its adequate assessment and exploration .Meskina [1]

This is due to the reality that manually analysing such a big quantity of information is practically impossible and therefore automated processes are necessary to conduct the assessment of this information. But even automated devices have restrictions on computing energy and memory, so it is computationally very costly to process such high-dimensional information. In addition, the high-dimensional data includes countless unrelated information that complicates the evaluation job and therefore results in other issues such as inaccurate classification precision (in the event of monitored learning) or bad performance clusters (in the situation of unsupervised learning) or over-fitting issues in particular cases. Giudici , Howley et al. [12,17]

Hence, there is a need to decrease the information sizes in order to eliminate or at least decrease these features . So, diverse techniques are available to lessen the size of this feature space. In the past, diverse techniques have been used for feature decrease and distinctive techniques have distinctive behaviours. Few techniques offer enhanced classification

precision then are very costly in terms of computation, however few are quite effective in runtime but then again do not always ensure outstanding classification precision.
The precision of classification differs with the classifier category used. SVM was looking better than others such as Naive Bayesian, K-NN, Decision Tree, etc. Then it too relies on the information collections used for the purposes of training and testing. It was noted that the same technique of decrease works enhanced than others for one set of information and works inferior to others in another set of information. Novakovic [2]

The classification precision differs considerably with the feature reduction and weighting technique used even for a specific classifier. Wrappers are regarded well than other techniques of selecting features such as filter techniques, but they are more computationally costly. This is because during their selection phase, they want to call the induction algorithm numerous times. IG was seen to demonstrate better classification outcomes than others among the filter techniques, but again it relies on data sets information gain (IG) demonstrates bad classification outcomes for some data sets than others. We cannot therefore ensure that a specific classifier or technique of reducing features is the best. Janecek et al. [3]

Principal Component Analysis (PCA) technique is the most normally used amongst the dimensionality reduction methods. Unlike feature selection techniques, the dimensionality reduction techniques do `not select a subset of the initial characteristics but instead convert the initial feature space into a new decreased feature space and thus the new characteristics are produced either from a linear or non-linear mixture of the initial characteristics and thus /no data about any variable is wasted in these techniques. PCA technique also uses less time to produce the same size of feature space than many feature selection techniques. Aha et al. [6]

Few writers even attempted to enhance the current techniques of feature decrease in order to enhance the classification precision of the classifiers using those techniques. The TFIDF algorithm used in the classification method of written papers has presented that there are few weaknesses in the assignment of weights to terms and therefore the author has proposed few changes in the current old-style strategy to TFIDF calculation Guo and Yang [4], but the author claims that the enhanced algorithm is more complex.

Another author has put effort into introducing bi-grams to advance the classification precision of text classifiers .In bi-grams we take the occurrence of two words together .It means how many times the two words have simultaneously come together in the document.. The author claimed that adding bi-grams enhanced the classifier's classification precision, but it also sometimes deteriorated the classifier's output. The findings are generally dependent on the dataset, so adding bi-grams sometimes increases classification precision while sometime also degrading it. Roy and Rossi [5]

That is why a lot of studies happening in the field of enhancing text classifier classification precision. Many writers proposed many modifications in current techniques of feature decrease or intact in the whole process of classification. But the maximum of the advances requires high computational costs also provide only negligible improvements in classification accuracy in return and even worsen the classifier's presentation in particular cases. There is a dependency on the data sets in addition to other factors, so conclusive which enhancements to integrate needs a well thought-out decision.

However, there were considerably fewer studies about the usage of ensemble learning methodologies specifically for text mining field where we can predict the severity of the text. The results produced using ensemble learning methods and by using different feature weighting techniques are presented in this thesis and we also include a comparative analysis with the all these machine learning techniques.

CHAPTER 3
TEXT MINING METHOLOGY

Since the amount of information is vast, it is almost impossible to estimate this big amount of information manually, since the manual analysis of this huge information would take a portion of cash and human effort, and a lot of time. So, this method of evaluating the information needs to be automated and this is where algorithms and ideas of machine learning and text mining play a crucial part. In almost every domain, there is a necessity for text classification such as software maintenance, software development, medical bodies, educational institutions, governmental organization, private organisations, etc. This chapter deliberates the following things: the necessity for document cataloguing, a short summary of the classification process and finally a short description of feature reduction methods.

3.1 Text classification steps

The goal of text mining is basically to abstract the significant features these features can be used to predict the model. Text classification is done in the following stages as shown in figure 3.1.It is a process of 5 stages, firstly data is represented in proper format then data is pre-processed and extraction of features from data is done then apply feature weighting techniques. Finally we can use this data for prediction.

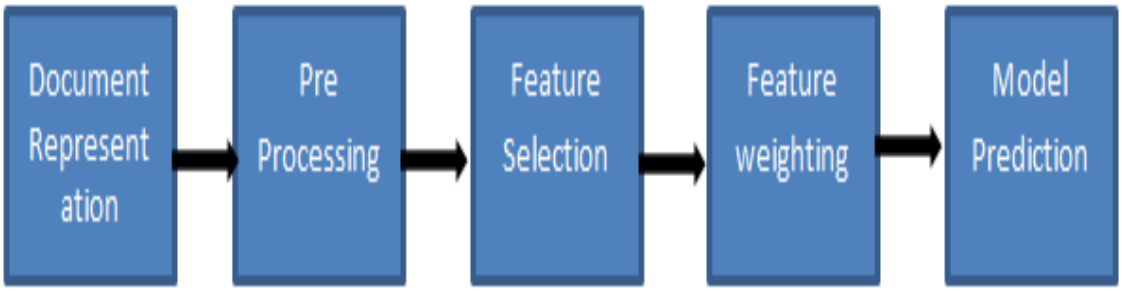

Fig. 3.1 Steps for Text Classification

3.1.1 Representation of Textual Document

The first stage is to portray documented records in a form that can be understandable and manageable. The documents must be depicted in a form in which they can be more readable .The bag-of-words is the most commonly used method to demonstration the text. The whole document is regarded as a set of words in this representation strategy. The phrases in the document may consist of verbs, nouns, or numbers, or punctuations, or articles, etc. Usually the text is collecting of thousands or millions of phrases. In the perspective of text mining, the words in the documents are denoted as a feature. Thus set of all the words forms feature space. Not all the features in the feature space are useful. Some of them even destroy the accuracy of the model. Also, with the rise in the magnitude of the feature space, memory requirements increase.Therefore, reducing this space size is very essential.

3.1.2 Pre-processing Steps

The text includes a bunch of phrases that are not helpful aimed at the assignment of cataloguing and their existence obscures the classifier and therefore leads to misclassification of a different document. Moreover, the higher the number of features, the higher the systems computational and memory demand. It is therefore very essential to remove such phrases and proceeds place in a sequence of stages known as pre-processing steps. Steps in document pre-processing are as follow:

1. Tokenization
2. Removal of Stop Word
3. Stemming

Text is regarded such as a tokens pool wherever a token is a character, a number, or a word, or punctuation, etc. Non-printable characters all punctuation characters will be detached and replaced by empty seats during the tokenization phase. The entire text is also transformed into characters in the lower case.

There are still many words in the classification method that are not helpful and are quiet existent in big quantities and if not removed, they will devour memory resources and a lot of computational energy. So, it is very vital to remove them. These arguments are called stop words. They are more often used phrases and therefore does not benefit in the classification process and they are either be an article, verb, adjective, noun, etc. that is meaningless to the classification process, e.g. Anything, most, a, an, the, is, if ,etc. The count of characteristics is significantly decreased after these two steps, but a significant step still requirements to be occupied to convert all the words that are derived from a shared origin into origin words.

3.1.2.1 Tokenization

The first step of pre-processing is Tokenization. The process of tokenization as shown in figure 3.2 includes removal of punctuation, replacement of special characters with blank spaces and irrelevant number. If the process of tokenization, the document is divided into well-formed collections of the token. These well-formed tokens are 1-gram,2-gram,N-grams.After the elimination of all the irrelevant letterings, the whole document is transformed into lowercase.

```
In [6]: import nltk

In [7]: text="the product shall be available during normal business hours as long as the user has access to the client pc the system will

In [8]:
tokenizer=nltk.tokenize.WhitespaceTokenizer()
tokens=tokenizer.tokenize(text)
print(tokens)

['the', 'product', 'shall', 'be', 'available', 'during', 'normal', 'business', 'hours', 'as', 'long', 'as', 'the', 'user', 'ha
s', 'access', 'to', 'the', 'client', 'pc', 'the', 'system', 'will', 'be', 'available', 'of', 'the', 'time', 'during', 'the', 'f
irst', 'six', 'months', 'of', 'operation']
```

Fig 3.2 Tokenization

3.1.2.2 Stop Words Removal

There are some words that make no sense in the document and these are not important words. So these words with no relevance should be removed as shown in figure 3.3. These words are verbs, articles, punctuations, adjectives, nouns, adverbs, etc. For example “the”, “a”, ”for”, ”is”, ”was”, “of” are of no relevance. These are not helpful for classification.

```
import nltk
from nltk import word_tokenize
from nltk.corpus import stopwords
stop = set(stopwords.words('english'))
sentence = "the product shall be available during normal business hours as long as the user has access to the client pc the syste
print([i for i in sentence.lower().split() if i not in stop])
```

```
['product', 'shall', 'available', 'normal', 'business', 'hours', 'long', 'user', 'access', 'client', 'pc', 'system', 'availabl
e', 'time', 'first', 'six', 'months', 'operation']
```

Fig 3.3 Stop word Removal

2.1.2.3 Stemming Algorithm

Porter's Stemming Algorithm is the most commonly used algorithm as shown in figure 3.4 for Stemming. Stemming Algorithms removes all the suffixes and prefixes of the. For example "sadness" is converted in the word "sad"," ness" suffice is removed from the word. There are many Stemming Algorithms that can be used for this purpose.

```
stemmer=nltk.stem.PorterStemmer()
" ".join(stemmer.stem(token)for token in tokens)
```

```
'the product shall be avail dure normal busi hour as long as the user ha access to the client pc the system will be avail of th
e time dure the first six month of oper'
```

Fig 3.4 Stemming algorithm

These steps do not decrease the number of features, but it is required to convert the words into their root form that is needed in the later classification. After all these three pre-processing steps, the original magnitude of the feature space is expressively decreased, but it quiet includes a percentage of phrases that are not very significant to the grouping task, thus we still want to decrease the magnitude of the feature space. These unrelated or fewer meaningful words resolve to complicate the classifier, resulting in bad classification precision. This further reduction of features is accomplished by feature reduction techniques.

3.1.3 Feature Selection

Pre-processing steps significantly decrease the magnitude of the feature set, but then there are many words in the documents that are not significant to the classification job. These phrases are considered otherwise they will significantly boost the system's computing power, and the request for storage rises as the number of characteristics in the feature space rises. It is therefore essential to use feature reduction techniques to lessen the dimension of the feature set. They are widely divided into two kinds – choice of subsets and decrease of dimensions. But their change is identical, i.e. to lessen the dimensions of the original feature space and to present the features beneficial for the resolution of classification.

Feature choice methods use an assessment function that is given to each document phrase and then the phrases that yield additional significance of the assessment function (or less significance based on the assessment function selected) are selected to illustrate the records. For this intent, there are several distinct assessment features, such as odds ratio, chi-squared, term frequency, relief-F, information gain, document frequency, symmetrical variance, etc.
By using this different approach, the magnitude of the original feature space is reduced in dimension. The objective now is to convert the original feature space into a fresh feature space that is lesser in magnitude than the input feature space original size. So, now the features are merged and an entirely fresh decreased range of feature space is produced. In this work we have used two different feature selection techniques.

3.1.3.1 TFIDF calculation

After applying the function choice technique to the phrases in the document, the files comprise only a few phrases that the selected function choice technique considers helpful for the evaluation process. Now, it is necessary to represent the records in the shape of a type N vector where N is the list of the phrases or terms chosen by the technique of selecting the function used previously. It is possible to represent these N words or features as t_1, t2,..., t_N. Thus, the ith document can now be represented in the form of a vector of N dimensions such as (X_{i1}, X_{i2},..., X_{iN}), where each X_{ij} in document I represents a weight of the term j and simply indicates the importance of that term in that document.

Vector Space model contains the collection of vector, vectors are N-dimensional vector calaculated form document in the set TFIDF calculation, takes account two things –TF and IDF. The TF basically is a measure of how frequently the word or term *t* is existent inside that document and can be calculated as:

TF = 0, if frequency count is zero 3.1a

TF = 1 + log{1 + log[frequency(t)]} , otherwise 3.1b

TDF or the inverse document frequency gives a measure of how rare the word or term is present across all the documents in the set and gives higher value or importance to those terms that are present rarely across the documents and a smaller value of importance to those which are frequently seen across the documents in the set. The idea is that the discriminative power of a term reduces if the term is present across many documents and increases if the term is present rarely across the documents. The IDF value of jth term can be calculated as:

$$IDF=\log\left(\frac{n}{nj}\right) \quad 3.2$$

where:

n_j is the total number of documents containing jth term

n is the total number of documents

Then TFIDF value of jth term can be calculated as:

$$TFID(X_{ij})=t_{ij} \times \log(n/n_j) \quad 3.3$$

where:

t_{ij} is the frequency of the jth term in document *i*

n_j is the total number of documents containing jth term

n is the total number of documents

So, we calculate the TFIDF values for all N terms in each of the documents and these values are then represented in a 2-dimensional matrix called the TFIDF matrix. But before using this matrix, we need to perform a very important step called normalization. In normalization, we simply normalize the weights of the terms for every document. Now the TFIDF matrix is ready and we can proceed towards our model prediction as discussed next.

3.1.3.2 Bag of Words

Bag of Words is a technique of extracting features from text files. These characteristics can be used to train algorithms for machine learning. It generates a language of all the distinctive phrases that occur in all the documents. For example, if you have three documents-
D1 - “I wish I could go to watch TV”

D2- “I am happy mood today”

D3 - “I am not good today”

First, it creates a vocabulary using unique words from all the documents –

A unique list of words –

I wish could go to watch TV am happy today not good

I	wish	could	Go	to	watch	TV	Am	Happy	Today	not	good
3	1	1	1	1	1	1	3	2	1	2	2

Formerly, for every word the frequency of the word in the equivalent document is inserted. The above table depicts the training features encompassing the term frequencies of every word in each document. This is called a bag-of-words approach since the number of occurrence and not order or sequence of words matters in this approach.

3.1.4 Model prediction

After converting training documents into numerical data, different ML techniques can be applied to predict model. The whole data is divided into 2 set i.e. training set and testing set. pre-processing steps are applied then TFID or BOW matrix is calculated for the data. Different techniques can be used for model prediction are like SVM ,NB , LR ,etc.

3.2 Overview of feature reduction methods

Data to be explored or analysed in most cases comprises a huge number of variables that are also entitled data dimensions. It is both helpful and dangerous to have this high-dimensional information. We have a ton of data information, so there is a need to properly evaluate and investigate data. On the other side, there are some severe issues with this elevated dimensionality of the information. One of the biggest issues is the system's elevated computing load. In addition, the requirement for storage rises significantly with the rise in data sizes.

Not only that, elevated information sizes degrade even a classifier's ranking precision. This is owing to the dimensionality curse. Having high dimensions implies that features or characteristics may be useless or meaningless to the evaluation assignment. Including such insignificant phrases can confuse the classifier's teaching method and this can contribute to many issues such as information overfitting. In the case of supervised learning, these meaningless phrases or characteristics may degrade the classifier's identification precision and may generate low-quality clusters in the event of unsupervised learning. We therefore need to decrease the volume of the feature space in order to solve all these problems.
The methods of function extraction can be widely categorized into two classifications-features selection (FS) and decrease of dimensionality (DR). We're only going to speak shortly about feature selection methods.We pick some characteristics from the original feature space in this function decrease strategy and extract the residual characteristics as they are meaningless for the intent of ranking. Since we extract some characteristics, this technique has some information loss but the data related to helpful phrases is maintained and used for ranking purposes. There are several methods from the original big feature space to

find the finest or ideal collection of characteristics. These approaches can be classified into 3 types – filters, wrappers and embedded one.

3.2.1 Filters

This selection technique works independently of the machine learning algorithm to be used later for classification. The filters approach work by removing irrelevant or redundant features from the feature space. The filter techniques make use of the data set itself to decide which attributes to discard and do not take into account any biases of the induction algorithm to be used later for classifying the data and due to this reason, they sometimes fail to achieve the desired accuracy as biases are inherent in some induction algorithms and they degrade the classification accuracy of the classifier.

3.2.2 Wrappers

This class of feature selection technique works as a feedback method and finds the optimal set of features by incorporating the induction algorithm in the process. That means it uses the machine learning algorithm to be used later to decide which subset of features is the best for classification. Since the number of possible subsets grows exponentially with an increase in the size of feature space, the wrappers approach is very costly in terms of computational burden and memory demands. The situation becomes even worse if the induction algorithm itself is computationally expensive.

3.2.3 Embedded approach

In embedded approach, the feature selection is an inherent part of the learning process of the induction algorithm, for example, artificial neural networks, decision trees, etc. do not need an explicit feature selection step as they have their own feature selection step present in their induction process.

CHAPTER-4
ML TECHNIQUES

There are many ML techniques. These ML Techniques can be applied to the text with ensemble learning to improve the classification method. We have taken two different ways of feature selection i.t TFID and BOW in order to improve the accuracy. The ML and ensemble technique used for the improvement of the model used in the study are summarized below.

4.1 Ensemble technique: Bagging

Bagging, a name comes from "bootstrap aggregation". It was a very effective and simplest technique of ensemble learning. According to Breiman, meta algorithm is one of the special cases for model averaging which was initially proposed for the classification and generally applies to decision tree models but in recent times it is used for regression or classification. Bagging uses different versions of a training data set in order to train a different model with the help of bootstrapping, which means sample with replacement. For single output, the output of all models is combined by voting (in case of classification) or averaging (in case of regression). Bagging is very effective if we used unstable nonlinear models because a minor alteration in the training data sets can result in an important change in the model. So we can say that bagging just like a bootstrap aggregation which works as a technique of cumulative accuracy that frequently samples from a particular dataset along with a uniform probability distribution. A sample size of each bootstrap is the same as the initial dataset since sampling finished through substitution.

Certain instances may seem numerous times in the identical training dataset, whereas others may possibly eliminate from the given training data set. Gathering of multiple predictors is the greatest significant feature of bagging. Bagging improves the ML algorithms and improves their stability and accuracy. It decreases variance and avoids over fitting. We can have more than one bootstrapped sample which will be used for the training purpose. In our study the models that we have used are the homogenous models. The performance of the bagging classifiers can be improved by varying the base classifiers. In this we have chosen 4 different base classifiers for the bagging technique. Following are the classifiers:

1. Bagged Linear SVM
2. Bagged Logistic Regression
3. Bagged Naïve Bayes
4. Bagged Random forest

Figure 4.1 shows the bagging process. Here we have three bootstrap sample i.e. sample 1,2 and 3. These samples are passed through learning techniques that are our ML techniques. The outcome of this pass is now combined and then a combined classifier is used for the prediction of the final output.

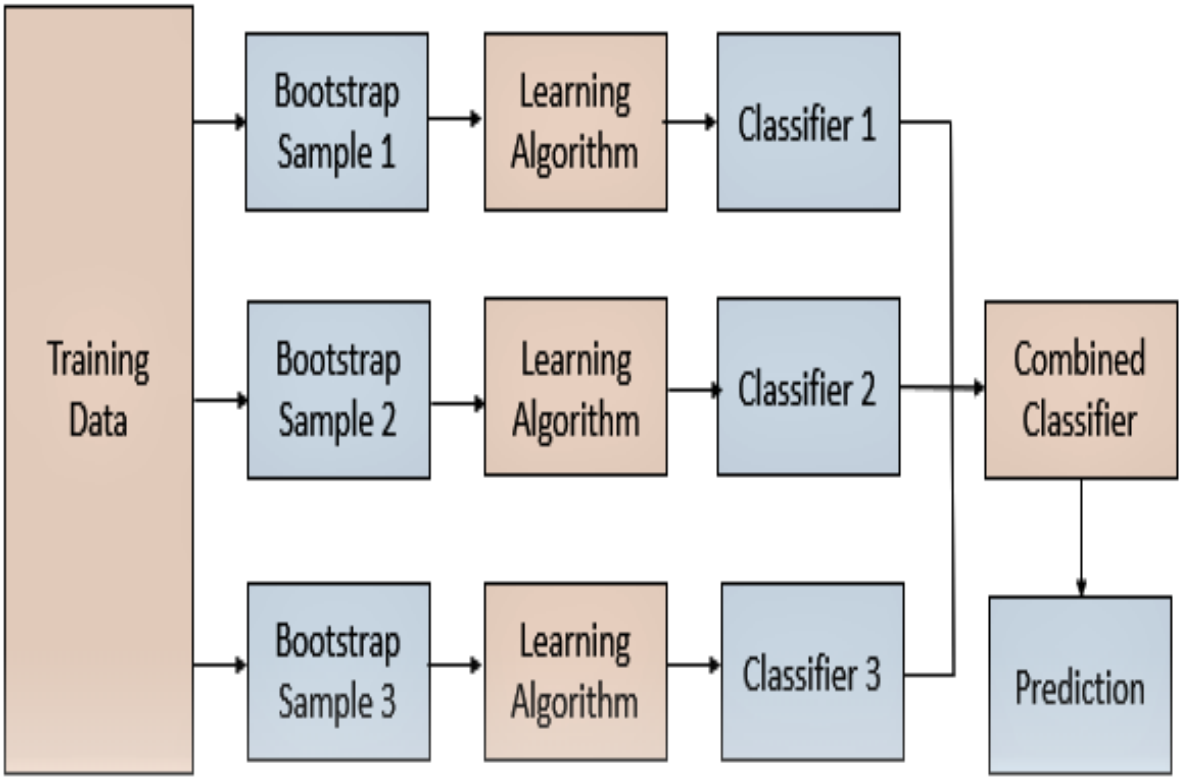

Fig 4.1 Bagging Process

4.2 ML Techniques

In our research, we have used some modern ML techniques available. Here NB , SVM ,LR and RF are used with various modern ensemble techniques like bagging. We have taken two different ways of feature selection in our study i.e. TFID and BOW.

4.2.1 Random Forest

The random forest starts by way of the random selection of n features from the complete N features. In the subsequent stage, by making the usage of the best split approach, we use the arbitrarily selected 'n' features in search of the root node. In the next stage, we'll use the best split approach to calculate the daughter nodes. The initial 3 sequential stages are reiterated until a root node forms the tree and the target is the node of the leaf. At last, to make 'n' randomly produced trees as shown in figure 4.2 we repeat one to four stages, this randomly created trees form the random forest. There are several unpruned regression or classification trees in random forests. Using a random selection of features, these trees are induced from training data bootstrap samples. Each data sample in the random forest is fed down each of the trees in classification problems. Then, the latter outputs the class that received most of the votes from the individual trees as its decision class.

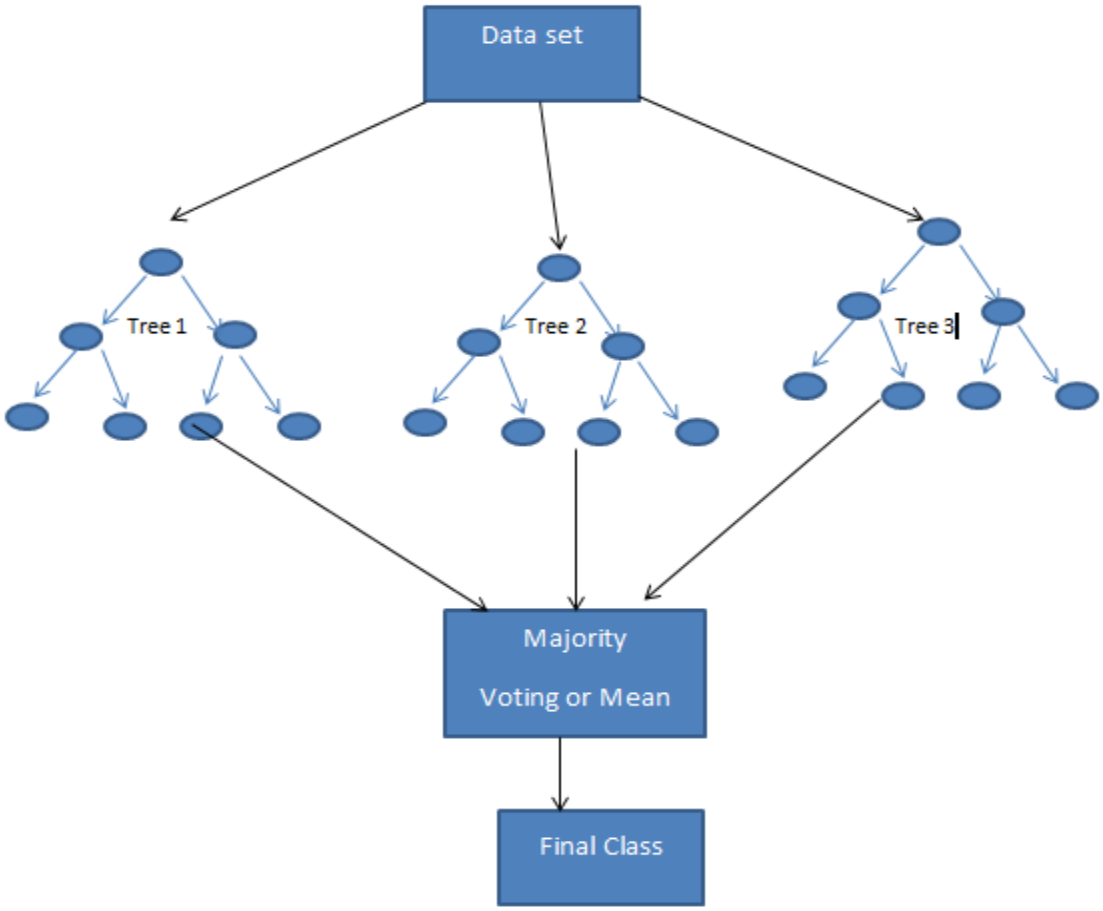

Fig 4.2 Random Forest

In order to predict the class using the random forest algorithm, we need to cross the test traits through the rules of each tree that have been created randomly. Suppose we were forming 50 random decision trees to form the RF for the same test feature, each RF will predict different targets. Then every predicted target vote will be considered.

4.2.2 Logistic Regression

Logistic Regression uses binary dependent variables. The binary dependent variable is those where the output can only take binary values and are used to represent such positive/negative outcomes. There are more than two outcomes of dependent variables. In the regression model which is used in Logistic Regression, the dependent variable is categorical. The generalized linear model is a super algorithm class that includes linear regression.In 1972, Nelder and Wedderburn proposed a model with the aim of providing a means to use linear regression to problems that were not directly suited to linear regression application. With the help of the logistic function, the relationship between the categorical dependent variable and independent

variables is measured using logistic regression. The fundamental equation of the linear model is given by equation 4.1:

$$LRM(E(y)) = \alpha + (\beta * x_1) + (\Upsilon * x_2) \quad 4.1$$

where :

LRM() is the link function

E(y) represents the expectation of the target variable

$\alpha + (\beta * x_1) + (\Upsilon * x_2)$ is the linear predictor and α, β, Υ are to be predicted. The link function is used for the linkage of expectation of y to that of the linear predictor.

4.2 .3 Naïve Bayes

Naive Bayes methods are a set of supervised learning algorithms based on the application of Bayes theorem with the “naive" assumption that each pair of features is independent.

Naive Bayes is a sort of classifier that makes the use of Bayes theorem. It calculates membership likelihoods for each class, such as the likelihood that a particular class belongs to a given record or data point. The most probable class is the class with the highest probability. This is also referred to as Maximum a posteriori probability (MAP). Equation 4.2 represents the relation between H and E.

$$\text{Maximum } (P(H/E)) = \text{Maximum}((P(E/H) * P(H))/P(E)) = \text{Maximum}(P(E/H) * P(H)) \quad 4.2$$

where

H is Hypothesis

E is Evidence

So, the crucks of naïve Bayes are that the classification of the Naïve Bayes depends as a simple classifier on the Bayes rule theorem of conditional probability. It assumes the values

of attributes are independent and unrelated, it is called the model of the independent feature. In many of the applications, Naïve Bayes uses the maximum probability methods to estimate its parameters.

4.2.4 Support Vector Machines

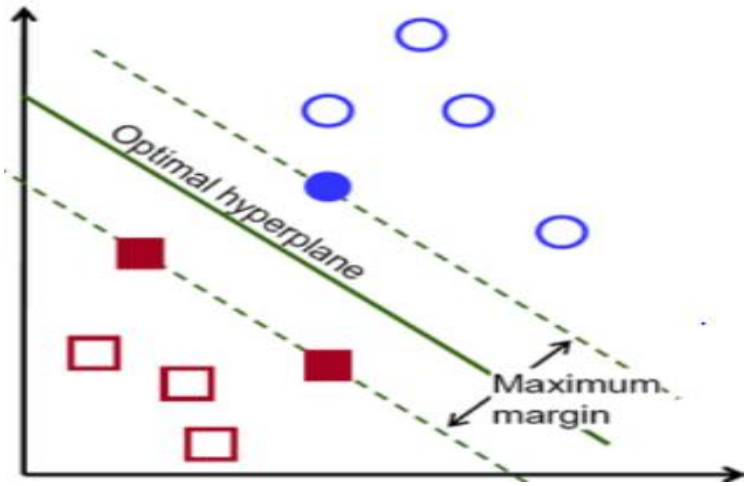

Fig 4.3 SVM Hyperplane

A Support Vector Machine (SVM) is interpreted experimentally as a discriminatory classifier by a separate hyperplane alternatively we can understand SVM as, given the supervised learning data (labelled training data), the algorithm that classifies new examples produces an ideal hyperplane. This hyperplane as shown in figure 4.3 itself is a line that separates a plane of data points into two areas into the two dimensional spaces where it sits in each class on either side. It uses the kernel trick generally to classify data that cannot be classified linearly. The algorithms main objective is to predict a plane that maximizes the distance between classes in order to reduce the possibility of over fitting and reduce the likelihood of misclassification of the new data point.

CHAPTER 5
EXPERIMENTAL DESIGN

This chapter is going to address the experimental design configuration needed for the outcomes. Method of text classification includes a sequence of measures such as the elimination of stop words, tokenization, stemming, choice of features choice and generating the TFIDF matrix and BOW. It explains the validation method used in the work. It also talks about the performance matrix used in the study.

5.1 Data sets for the experiment

Data set taken into account is PITS-An information collection provided by the Independent Verification and Validation (IV & V) program of NASA technology. Issues or difficulties related to humanoid rated devices and robotic satellite tasks have been gathered and included in this information collection for about added 10 years. The collections of information contain accounts of fault. A fault study involves the definition of the faults, their identification and their amount of seriousness connected with them. The errors can be split into 5 seriousness rates, which are very small, small, intermediate, high, and very high, according to NASA technicians.

The faults with high severity levels are serious as they are treated to safety and security. Likewise, such faults are terrible towards mend. That's why, in the empirical study, the severity level 1 is well-thought-out and only following 4 severity levels have been deliberated, i.e., severity 5 (very low), severity 4 (low), severity 3 (medium), and severity 2 (high). The collection of information comprises a sum of 960 faults of varying rates of seriousness (from 2 to 5). Their allocation to these four rates of seriousness is as follows:

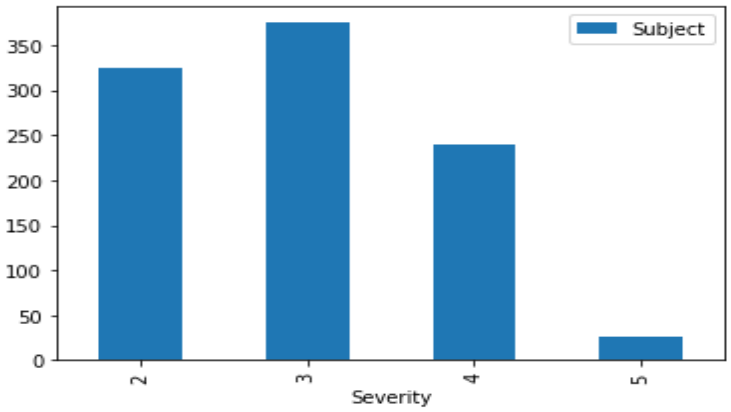

Fig 5.1: Fault counts and related severity levels in the data set

Table 5.1: Fault counts and related severity level in the data set

Severity Level	Fault count
2	320
3	375
4	239
5	26

5.2 Validation technique

For validation purpose k-fold cross validation is taken into account. The entire data is separated into k folds .Among these k folds 1 fold is taken for test data and the residual (k-1) folds are used for training data and this method is recurrent k times so that each time a fresh fold is used as test data. This guarantees that each roll or document is used both for training purposes and for testing purposes. We took the significance of k as 10 in our situation. We have therefore split our information collection with 960 fault explanations into two collections-training set and test set, and this happens for 10 times. So that each file is used as practice and test information.

The data set includes faults corresponding to four concentrations or categories of seriousness, in both practice and experiment information collections, we have attempted to keep the percentage of faults assigned to these groups. Therefore, the distribution of faults in the training and test data sets to four severity levels is alike to the deliveries publicized respectively in table 5.2 and table 5.3

Table 5.2: Fault count and related severity levels in training data set

Severity Level	Fault count
2	288
3	337
4	215
5	24

Table 5.3: Fault count and related severity levels in the testing data set

Severity Level	Fault count
2	288
3	337
4	215
5	24

5.3 Performance metrics

To assess the work of any model or process, quality assessments are needed. Since our primary objective is to enhance the classifier's ranking precision. The quality of ranking can be evaluated by various quality metrics, such as F-measure, precision, accuracy, recall, etc. We have used Accuracy and F1-score . Accuracy is computed as the ratio of correctly predicted instances of the testing dataset to the total number of instances of the testing dataset . Accuracy is given by:

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+TN+FN+FP)} \qquad 3.1$$

where,

TP is known as true positive,

TN is known as true negative,

FN is known as false-negative and

FP is known as a false positive.

F1-Scores is computed as the harmonic mean of Precision and Recall, which are then calculated by using the confusion matrix Precision is computed as the ratio of the correctly positive predicted (buggy) instances to the total count of the positively predicted instances. The recall is computed as the ratio of correctly positive predicted (buggy) instances to the total count of the positive instances

CHAPTER 6
RESULTS AND ANALYSIS

The two typed of feature weighting techniques used are i.e. TFID and BOW. With both of these techniques bagging is also combined. First the data was pre-processed and then feature selection and feature weighting algorithms are applied to it. Then the analysis is done using k-fold validation

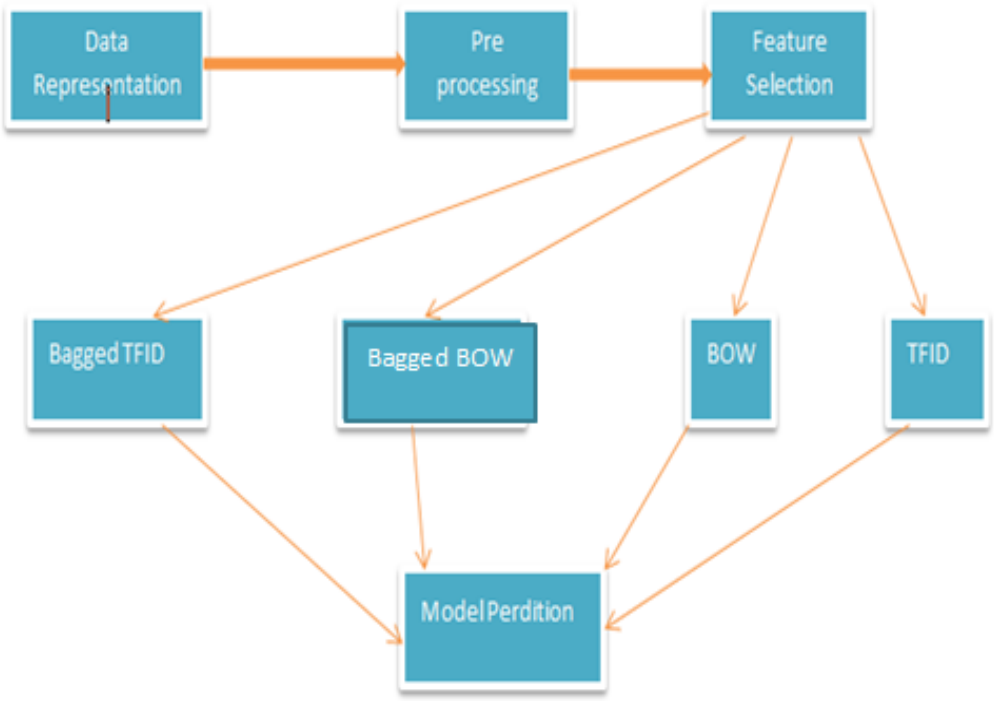

Fig 6.1: Techniques used for text

Figure 6.1 represents the different types of feature weighting technique in the proposed work for text classification. This figure shows the process of text classification with starting from data representation, pre –the processing of data, selection of features and applying different feature weighting Techniques and finally predicting the tags.

6.1 Analysis

The data was fed to the model which was implemented using python. The performance metric used is Accuracy. It measures the number of correct samples predicted over the total

number of samples. For example, if the classifier is correct for 80 percentages, it means that it correctly predicts the class for 80 of them out of 100.

6.1.1 Frequency of words in Data Set

Figure 6.2 shows the frequency of the top 50 words that are used in the subject column of the dataset. This graph represents exactly how many times a particular word seemed in the whole document. This frequency calculation is used for computation of BOW.

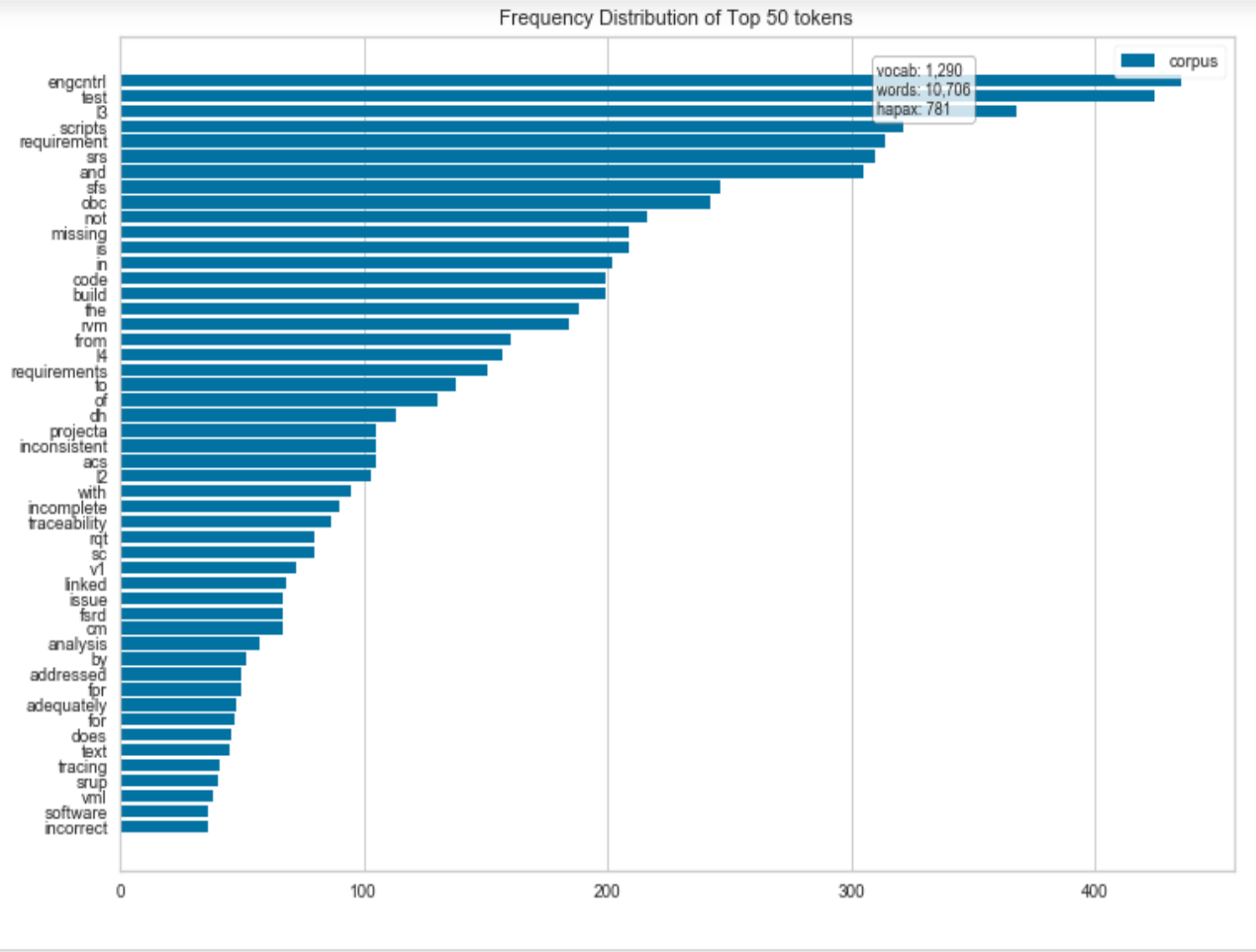

Fig 6.2: Frequency of words in Data set

6.1.2 Accuracy

The performance metric used is Accuracy. It measures the number of correct samples predicted over the total number of samples. For example, if the classifier is correct for 90 percentage, it means that it correctly predicts the class for 90 of them out of 100 instances.

6.1.2.1 Accuracy of TFID with and without Bagging

Figure 6.3 depicts graphically the accuracy of TFID with and without bags. Here we can see observe two things. Firstly TFID with bagging gives better results as compared to normal TFID. Secondly, NB gives better accuracy as compared to SVM, LR, and FR when TFID is used for the feature weighting.

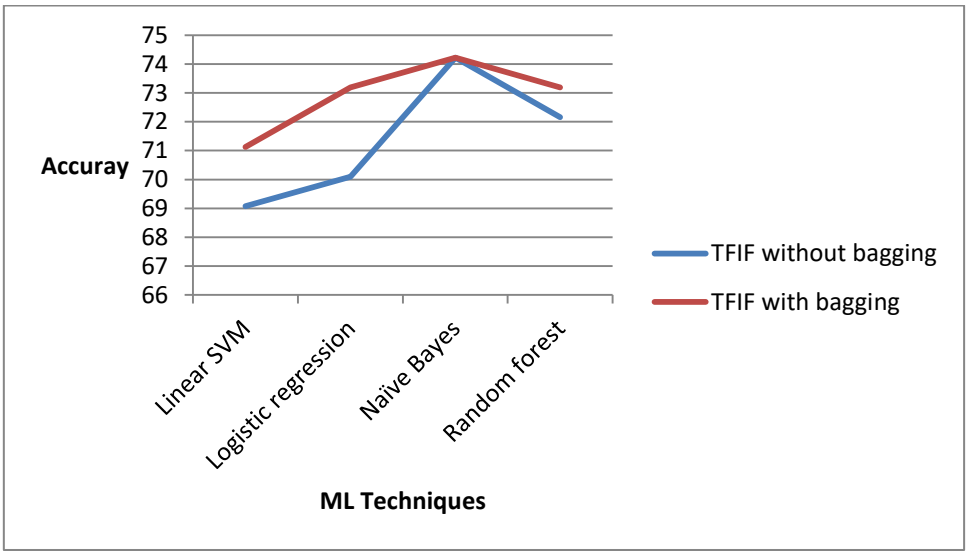

Fig 6.3: Accuracy of TFID with and without bagging

6.1.2.2 Accuracy of BOW with and without bagging

Figure 6.4 depicts graphically the accuracy of BOW with and without bagging. Here we can see observe two things. Firstly BOW with bagging gives better results as compared to normal BOW. Secondly, SVM gives better accuracy as compared to NB, LR, and FR when BOW is used for the feature weighting.

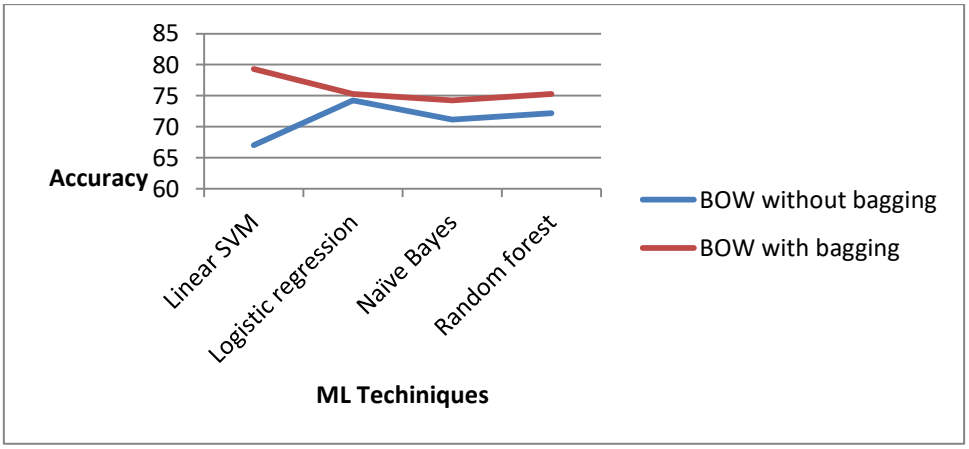

Fig 6.4 Accuracy of BOW with and without bagging

6.1.2.3 Accuracy comparison of TFID and BOW

Table 6.1 and Figure 6.5 depicts the comparison between both the weighting techniques. LR with BOW and NB with TFID perform well as compared to other machine learning techniques.

Table 6.1: Accuracy of TFID and BOW

Techniques	TFIF	BOW
Linear SVM	69.07	67.01
LR	70.1	74.22
NB	74.22	71.13
RF	72.16	72.16

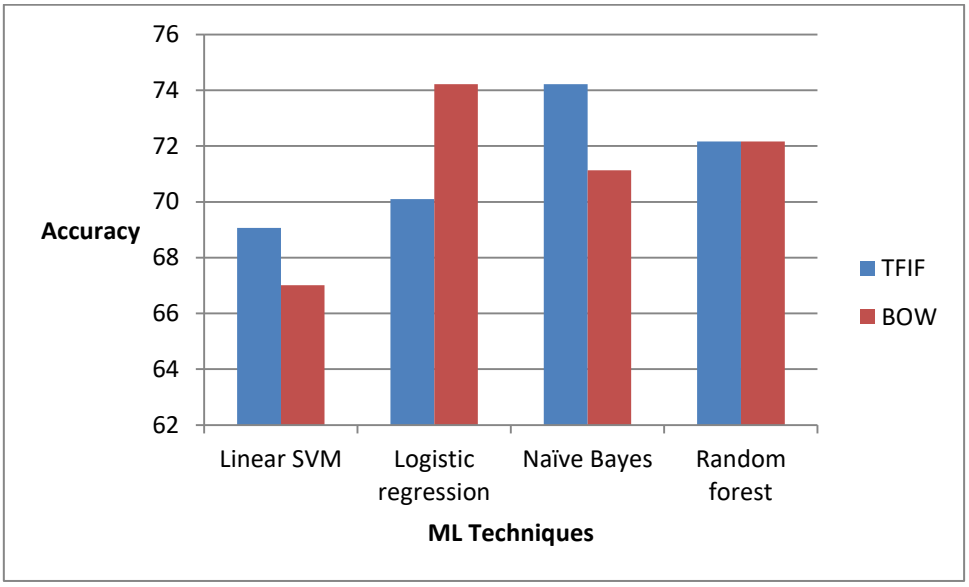

Fig 6.5: Accuracy of Techniques using TFID and BOW

6.1.2.4 Accuracy comparison of TFID and BOW with bagging

Table 6.1 and figure 6.5 depicts the comparison between both the weighting techniques plus ensembles technique is included.SVM with BOW performs well as compared to other machine learning techniques.

Table 6.2: Accuracy of TFID and BOW with bagging

Techniques	TFIF	BOW
Bagged Linear SVM	71.13	79.3
Bagged LR	73.19	75.25
Bagged NB	74.22	74.22
Bagged RF	73.19	75.25

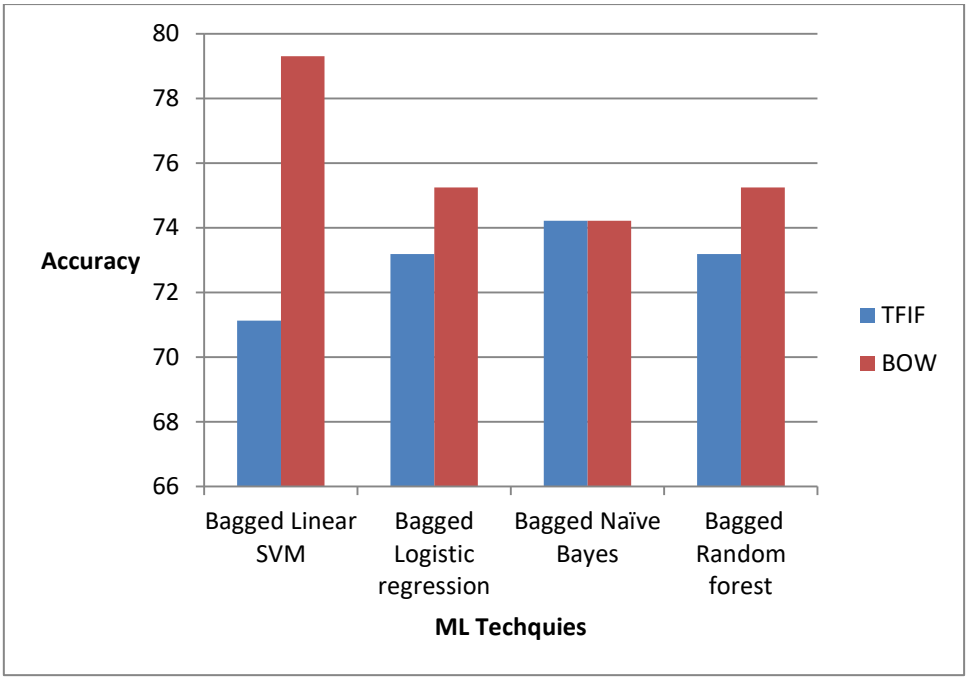

Fig 6.5: Accuracy of Techniques using TFID and BOW with bagging

6.1.2.5 Accuracy comparison for all the techniques

From table 6.3 and figure 6.6 we can conclude that bagged SVM with BOW as the feature weighting techniques perform well as compare the other techniques. Ensemble techniques can improve the accuracy of the model. Here accuracy of SVM is increased with ensemble learning and change in weighting Techniques.

Table 6.3: Accuracy of all Techniques

Techiques	TFID	BOW	Bagged TFID	Bagged BOW
Linear SVM	69.07	67.01	71.13	79.30
LR	70.1	74.22	73.19	75.25
NB	74.22	71.13	74.22	74.22
RF	72.16	72.16	73.19	75.25

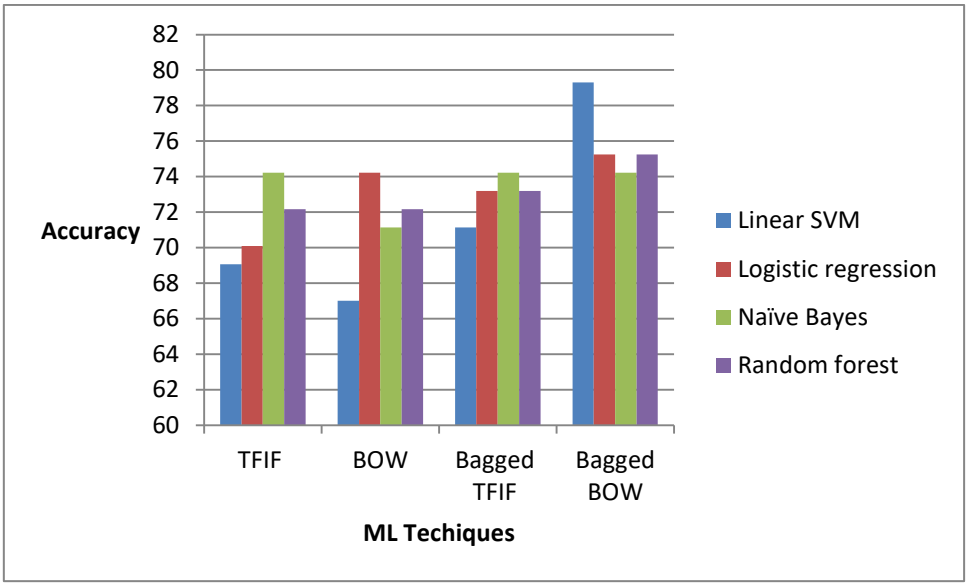

Fig 6.6: Accuracy of all Techniques

Therefore, we can say that the BOW method of feature selection is better than the TFID method. By using ensemble learning accuracy is improvised further but again accuracy depends on the data set to be used.

6.1.3 F1 –score comparison for all the techniques.

From table 6.4 and figure 6.7 we can conclude that bagged SVM with BOW as the feature weighting techniques perform well as compare the other techniques. Ensemble techniques can improve the F1-score of the model. Here F1-score of SVM is increased with ensemble learning and change in weighting Techniques.

Table 6.4: F1-score of all Techniques

Techiques	TFID	BOW	Bagged TFID	Bagged BOW
Linear SVM	69	67	71	79
LR	70	74	73	75
NB	74	71	74	74
RF	72	72	73	75

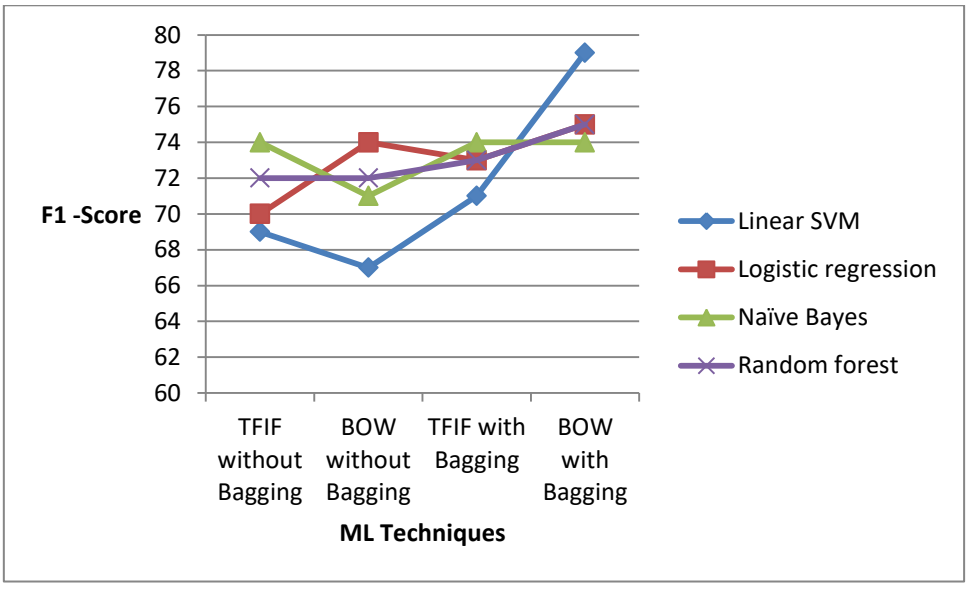

Fig 6.7: F1-score of all Techniques

CHAPTER-7

CONCLUSION AND FUTURE WORK

In today's world, there is a strong demand for text classification as the information grows at an alarming rate and this huge data cannot be evaluated continuously and therefore automated procedures are needed to evaluate and investigate this big quantity of data. The feature reduction technique used is one such significant consideration. Different techniques of feature decrease have varying impacts on the ranking process's accuracy

It is concluded from the above discussion that various techniques that are involved in text mining are explored to present knowledge in a concise format. Good Research work is going on in the field of Text mining to improve the accuracy of the model. Still a lot of like parameters like which method is best for weighting in the process of feature selection or which techniques are best for text classification..

With the help of ensemble learners bagging and using different techniques for feature selection we find an increase in the accuracy of the model. Thus they helped to gain the performance increase in base learners. Future work may involve the exploration of other feature selection techniques and ensemble learning for the text classification and their capabilities can be used to increase the performance of the model further.

REFERENCES

[1] S. B. Meskina, “On the effect of data reduction on classification accuracy,” presented at *IEEE 3rd International Conference on Information Technology and e-Services*, Sousse, Tunisia, 2013.

[2] J. Novakovic, “The impact of feature selection on the accuracy of Naïve Bayes classifier,” *18th Telecommunications forum TELFOR*, Serbia, Belgrade, November 23, 2010.

[3] A. G. K. Janecek, W. N. Gansterer, M. A. Demel, and G. F. Ecker, “On the relationship between feature selection and classification accuracy”, *Journal of Machine Learning Research*, vol. 4, pp. 90-105, 2008.

[4] A. Guo, and T. Yang, “Research and improvement of feature words weight based on TFIDF algorithm,” *Information Technology, Networking, Electronic and Automation Control Conference, IEEE*, Chongqing, China, May 20-22, 2016.

[5] N. K. S. Roy and B. Rossi, “Towards an improvement of bug severity classification,” *40th Euromicro Conference on Software Engineering and Advanced Applications*, Verona, Italy, August 27-29, 2014.

[6] D. W. Aha, D. Kibler and M. K. Albert, "Instance-based Learning Algorithms", Machine Learning, vol.6, no.1,1991.

[7] L. Almuallim, and T. G. Dietterich, “Learning With Many Irrelevant Features”, in the *Proceedings of the Ninth National Conference on Artificial Intelligence*, vol. 2 pp 547-552, (AAAI-1991).

[8] H. G. Callan, J. G. Gall, and C. Murphy, Histone genes are located at the sphere loci of Xenopus lampbrush chromosomes, Chromosoma 101, pp. 245-251,1991.

[9] M. Dash, and H. Liu, “Feature Selection for Classification”, *Intelligent Data Analysis*, vol. 1, no. 3, 1997, pp.131-156.

[10] P. A. Devijver, and J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, Englewood Cliffs, NJ 1982.37

[11] F. Flitti, Techniques de reduction de données et analyse d'images multispectrales astronomiques par arbres de Markov, PhD thesis, Louis Pasteur University, 2005.

[12] P. Giudici, Applied Data Mining: Statistical Methods for Business and Industry, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, England, 2003.

[13] I. Guyon, and A. Elisseeff, "An introduction to variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182,2003.

[14] W.M. Hartmann, "Dimension Reduction versus Variable Selection", Lecture Notes in Computer Science, vol. 3732, pp. 931-938,2006.

[15] R. Malhotra. Empirical Research in Software Engineering, CRC Press, pp. 365-389, 2015.

[16] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer series in statistics*, Springer, New York, 2001.

[17] T. Howley, M. G. Madden, M. L. Connell and A. G. Ryder, "The effect of principal component analysis on Machine Learning accuracy with high dimensional Spectral Data", In the *Proceedings of AI-2005,25th International Conference en Innovative Techniques and Applications of Artificial Intelligence*, Cambridge, 2005.

[18] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant features and the subset selection problem", on *Proceedings of the Eleventh International Conference of Machine Learning*, pp. 121-129, Canada, 1994.