# Improving Classification Accuracy of Text Classifiers

THESIS SUBMITTED IN PARTIAL FULFILMENT OF REQUIREMENT
FOR THE AWARD OF THE DEGREE OF

**Master of Technology**
**in**
**Computer Science and Engineering**

Under the guidance of
**Dr. Ruchika Malhotra**
**(Associate Head and Associate Professor**
**– Computer Science and Engineering)**
Delhi Technological University

Submitted By-
**Shivam Rastogi**
(Roll No. - 2K16/CSE/14)


**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**DELHI TECHNOLOGICAL UNIVERSITY**

(Formerly Delhi College of Engineering)

Shahabad Daulatpur, Main Bawana Road, Delhi-110042

June 2018

# CERTIFICATE

This is to certify that project report entitled "**Improving Classification Accuracy of Text Classifiers**" submitted by **Shivam Rastogi** (Roll No. 2K16/CSE/14) in partial fulfilment of the requirement for the award of degree MASTER OF TECHNOLOGY in Computer Science and Engineering at DELHI TECHNOLOGICAL UNIVERSITY is a record of the original work carried out by him under my supervision.

Place : Delhi

Date :

**SUPERVISOR**

DR. RUCHIKA MALHOTRA

Associate Head and Associate Professor

Department of Computer Science and Engineering

Delhi Technological University

Bawana Road, Delhi -110042

# DECLARATION

I hereby declare that the thesis work entitled "**Improving Classification Accuracy of Text Classifiers**" which is being submitted to Delhi Technological University, in partial fulfilment of requirements for the award of degree of Master of Technology (Computer Science and Engineering) is a bonafide report of Major Project-II carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Place : Delhi

Date :

SHIVAM RASTOGI

2K16/CSE/14

# ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Dr. Ruchika Malhotra (Associate Professor, Computer Science & Engineering Department) for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to her for the support, advice and encouragement she provided without which the project could not have been a success.

Secondly, I am grateful to Dr. Rajni Jindal, HOD, Computer Science & Engineering Department, DTU for her immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

**SHIVAM RASTOGI**

**Roll No. 2K16/CSE/14**

**M. Tech. (Computer Science and Engineering)**

**Delhi Technological University**

# 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 devised a new formula for evaluating the words in the documents and thus finding the better words which are more useful in the classification task. Our method aims to find those words which have more discriminating power than others and therefore, we have named our formula as Discriminating Power (DP). So, we need to find DP of every word in the document and then select those which have more value of DP as higher the value of DP of a word, the better it is for the classification purpose. We have also compared the results of using Infogain method and our Discriminating Power method 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 for different values of feature counts selected.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| 1. SVM: | Support Vector Machine |
| 2. KNN: | K-Nearest Neighbour |
| 3. TFIDF: | Term Frequency Inverse Document Frequency |
| 4. IG: | Information Gain |
| 5 DP: | Discriminative Power |
| 6. PCA: | Principal Component Analysis |

# CHAPTER 1

# INTRODUCTION

In today's world, the data is one of the most important asset of any organization. The normal business processes generate a huge amount of data everyday and the good management of the business requires the proper analysis of its data. But since the amount of data is huge, it is practically impossible to analyse that data manually. Therefore, there is a need of automated procedures for doing so. One of the main challenges that organisations face is to classify their textual documents into some fixed predefined categories and after the documents have been classified, the decision on managing those documents becomes quite easier. For example, if we have a bug report file containing descriptions of the bugs, then it would be a great help if those bugs could be classified into their severity levels and then those with higher severity can be dealt with first on priority. This will ensure that bugs which are more important than the others are handled first. This will also ensure better utilisation of resources like money, human power, time, etc. [5].

So, there is a need for automated document classification which could classify a set of documents into some fixed predefined categories. But then the classification accuracy of the classifier becomes an important factor, and we want the classifier to be as accurate as possible. The classification accuracy of a text classifier depends on many things like machine learning method used (Support Vector Machine, Decision Tree, Bayesian Network, etc), feature reduction method used, etc. [1], [3].

The feature reduction method used to reduce the count of features is a very important step in the process of text classification and depending on how well the method performs its selection of features have a direct impact on the classification accuracy of the classifier [1] [18]. One of the measures used for finding most informative words is Information Gain (IG). But this method of evaluating the usefulness of words has few shortcomings. For example,

suppose there are 1000 documents belonging to category A and 10 documents belonging to category B and there are two words $W_1$ and $W_2$ such that $W_1$ is present only in category A documents and $W_2$ is present only in category B documents. So, it is clear that they both have "same" discriminative power since both are present in exactly one of the categories and are absent in the rest of the categories but still the Information-Gain method will prioritise the word $W_1$ over $W_2$ as the number of documents in category A are more than those in category B. Therefore, we have devised a new measure for evaluation of the discriminative powers of the words present in the documents which will give same priority to those words which have same discriminative powers and higher priority will be given to any word only if the word has more discriminative power. We have named our evaluation measure as Discriminative Power (DP). So, we evaluate the DP value for each word in the document and those with higher values are preferred to those with lower values as higher the DP value of a word, the better it is for classification purpose.

So, we have implemented a classifier that uses Nearest Neighbour technique for classification and we have used Information-Gain and our Discriminative-Power measures in the feature selection step and obtained the results after using each of these two methods separately. We found the classification accuracy improves when our Discriminative Power method is used for feature selection. The details of the method and results will be discussed in details in later chapters.

There are total 6 more chapters after this introduction chapter which are organised as follows : Chapter 2 discusses the theoretical background and motivation. It gives an overview of the whole classification process and the factors affecting the classification accuracy. Then we have Chapter 3 which covers Literature Review and it is followed by Chapter 4 which discusses our proposed methodology. Chapter 5 gives information about Experimental design and is followed by Chapter 6 which shows results and analysis of the outcomes obtained after experiment. Chaper 7 then talks about the conclusion and is followed by the list of the references used.

# CHAPTER 2

# BACKGROUND AND MOTIVATION

This chapter discusses about the following : need for automated classification of documents, a brief overview of the classification process, classification accuracy and factors affecting it, brief overview of feature reduction methods and lastly, the shortcomings in Information-Gain method.

## 2.1 NEED FOR AUTOMATED TEXT CLASSIFICATION

The data in almost every organization is expanding at an incredible rate. Usually, most of the data is in the form of textual documents and there is a constant demand of analyzing this huge data as this data contains a lot of information required for the proper management of various processes in the organization. Since the amount of data is huge, it is practically impossible to manually analyze this large amount of data as manual analyzing of this enormous data would require lot of money and human efforts and of course lot of time too.

Therefore, there is a need to automate this process of analyzing the data and that is where text mining and machine learning algorithms and concepts play a vital role. One of the main challenges in this era of huge growth of textual documents is to classify these textual documents into some fixed predefined categories or classes. This categorization of documents is very helpful in many cases, for example, consider a Software Requirements Specification (SRS) document which contains all sorts of software requirements in usually unorganized manner. The document may contain requirements associated with different dimensions like availability, usability, security, maintainability, etc. So, it would be nice if we could categorize the unlabelled requirements into their associated categories and then we can handle requirements in a well organized manner, for instance, security requirements are very important and so can be taken on priority right from the beginning of the software design.

Thus, there is a need for automated text classification in almost every domain like software development, software maintenance, educational institutions, medical bodies, government organizations, private organizations, etc.

## 2.2 STEPS INVOLVED IN AUTOMATED TEXT CLASSIFICATION

There are a series of steps required to be followed in the process of text classification as shown in Fig.2.1


**Fig. 2.1 Steps involved in Text Classification**

The brief overview of each of these steps is as follows :

### 2.2.1 Representation of Textual Documents

The first step is to represent the textual documents in a form that can be understood and managed by the underlying classification software. The documents should be represented in a form so that they can be analyzed further easily and *bag-of-words* approach is one of the most widely used representation approach. In this representation approach, the entire document is considered as a collection of words. The words present in the document may be noun, or verb, or articles, or punctuations, or numbers, etc. The document is thus usually a collection of hundreds or thousands of words. But not all these words are significant as there may be redundant or irrelevant words too. The words in these documents are referred to as features in the context of text mining and thus the set of all these words constitute the feature space. Not all the features or words present in the feature space contribute to the process of classification as many of them are irrelevant to the process of classification and if not removed from the feature space, they may degrade the classification accuracy of the classifier very badly. Also, the computational cost and memory requirements increase with the increase in the size of the feature space.

Therefore, it is very necessary to reduce this feature space size by removing the unnecessary features. This is done by a series of preprocessing steps as discussed next.

### 2.2.2 Preprocessing Steps

The document contains a lot of words that are not useful for the classification task and also, their presence confuses the classifier and thus it results in misclassifications of various documents. Also, the more the number of features, the greater is the computational and memory demand of the system. Therefore, the removal of such words is very much necessary and this takes place in a series of steps called preprocessing steps. The steps involved in preprocessing of the documents are :

1. Tokenization
2. Stop words removal
3. Stemming

The document is considered as a collection of tokens where a token may be a word, a punctuation character, or a number, etc. In the process of tokenization, all the punctuation characters and non-printable characters are removed and replaced by blank spaces. Also, the whole text is converted into lowercase characters.

There are still many words which do not help in the classification process and are still present in a large amount and if not removed, they will consume a lot of computational resources and memory resources. Therefore their removal is very important. These words are called stop words and these are very commonly used words and thus do not help in the classification activity and they may be either an article, adjective, noun, verb, pronoun, etc which are irrelevant for the classification process, for example, a, an, the, is, if, are, anything, most, etc.

After these two steps, the count of features is considerably reduced but still an important step needs to be taken which converts all words that are stemmed from a common root to their root word. This step is called stemming and it replaces all the words that are generated from a particular root word by that concerned root word, for example, move, moves, moved etc all are generated from a common root word “move” and so they all are replaced by the root word “move”. This step does not reduce the feature count but still is

required to bring out this root word transformation which is very much necessary in the classification to be done later.

After all these 3 preprocessing steps, the initial size of the feature space is considerably reduced but still it contains a lot of words that are not much relevant for the classification task, so still we need to reduce the size of the feature space further otherwise these irrelevant or less relevant words will confuse the classifier and this will lead to poor classification accuracy and also increased computational and memory demands of the system. This further reduction of features is carried out by feature reduction methods as discussed next.

### 2.2.3 Feature Selection

The preprocessing steps considerably reduce the size of the feature set but still the documents contain a lot of words that are not relevant for the classification task. These words need to be removed otherwise they will considerably increase the computational burden on the system, Also, the memory demand increases with the increase in the number of features in the feature space. Therefore, feature reduction techniques are required which aim to reduce the dimensionality of the feature set. They are broadly of two types – feature subset selection and dimensionality reduction. But their purpose is same, i.e., to reduce the size of the initial feature space and present the features that are useful for the classification purpose.

The feature selection methods use an evaluation function which is applied to every word of the document and then the words which produce more value of the evaluation function (or less value depending upon the chosen evaluation function) are chosen to represent the documents. There are several different evaluation functions that have been used for this purpose, for example, information gain, mutual information, odds ratio, term frequency, document frequency, symmetrical uncertainty, chi-squared, relief-F, etc.

In dimensionality reduction techniques, the size of the initial feature space is reduced using a different approach. Here, there is no subset selection like in feature selection methods. The idea here is to transform the initial feature space to a new feature space which is smaller in size than the initial size of the input feature space. So, here the features get combined and produce a totally new reduced set of feature space. Principal component analysis is one of the most popular methods of this category which produces a linear combination of the initial feature set.

IG is one of the most widely used method that is used for feature selection. It ranks all the words based on the values of their IG measure and then top N scoring words are chosen to represent the documents and rest of the words are simply ignored. The classification accuracy, however, depends on the value of N and varies considerably sometimes with the change in the value of N.

### 2.2.4 TFIDF calculation

After the feature selection method is applied to the words in the documents, the documents contain only some words that are considered useful for the classification process by the chosen feature selection method. Now, the documents are required to be represented in the form of a vector of size N where N is the count of the words or terms selected by the feature selection method used earlier. These N terms or attributes can be represented as $t_1$, $t_2$, …, $t_N$. So, the $i^{th}$ document can be represented now in the form of a vector of N dimensions as ($X_{i1}$, $X_{i2}$, …, $X_{iN}$), where each $X_{ij}$ represents a weight of the term *j* in the document *i* and simply signifies the importance of that term in that document. This N-dimensional vector is calculated for each of the documents in the set and this whole collection of vectors then constitutes the vector space model.

In TFIDF calculation, there are 2 things – term frequency (TF) and inverse document frequency (IDF). The term frequency (TF) simply is a measure of how frequently the word or term *t* is present inside that document and can be calculated as :

$$TF = 0, \text{ if frequency count is zero} \quad (2.1a)$$

$$TF = 1 + \log\{1 + \log[\text{ frequency}(t) ]\} \text{ , otherwise} \quad (2.1b)$$

The IDF 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 smaller value of importance to those which are frequently seen across the documents in the set. The idea is that 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 $j^{th}$ term can be calculated as :

$$\text{IDF} = \log_2\left(\frac{\text{n}}{n_j}\right) \qquad (2.2)$$

where :

$n_j$ is the total number of documents containing $j^{th}$ term

n is the total number of documents

Then TFIDF value of $j^{th}$ term can be calculated as:

$$\text{TFIDF}(X_{ij}) = t_{ij} \times \log_2\left(\frac{\text{n}}{n_j}\right) \qquad (2.3)$$

where :

$t_{ij}$ is the frequency of the $j^{th}$ term in a document *i*

$n_j$ is the total number of documents containing $j^{th}$ 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 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.

### 2.2.5 Model prediction

We take the training data set and then apply all the earlier steps on those documents and finally get the TFIDF matrix where each row represents a document in terms of N chosen terms. These values represent a pattern as for training set , we also have their labels of classes to which they belong and so for each class or category, we have usually several patterns. Now, this knowledge of patterns can be used to predict the class or category of the unlabelled test set document. We can use many machine learning algorithms at this stage to train the classifier using this knowledge of patterns and then the classifier can be used to predict the class of the unknown instance. One of the most common classifier used for this purpose is K-Nearest Neighbor (K-NN) classifer.

## 2.3 CLASSIFICATION ACCURACY AND FACTORS AFFECTING IT

The automated classification of documents classifies the documents into some fixed predefined categories but not every classification done is correct and there are some misclassifications done by the classifier. So, the classification accuracy is an important factor and has to be taken care of properly.

It has been observed that classification accuracy of a classifier depends on several factors. One of the major factors is the classifier itself, i.e., for same training and test data set, different classifiers give different results and have different classification accuracy. Sometimes, Support Vector Machines (SVMs) perform the best classification whereas sometimes Bayesian classifiers perform better or sometimes some other classifier like K-NN or Decision Tree classifier performs better than the others.

Even if we use a particular classifier, the classification accuracy varies considerably sometimes depending upon chosen training and/or test data sets. For different training and/or test data sets, the classification accuracy comes out to be very different sometimes. Even for the same training set, the classification accuracy changes for different data sets. It can give an excellent accuracy of 99.1% for one test data set while it can just give a poor accuracy of 30% for another test data set. Similarly for same test data set, the classification accuracy may vary considerably when different training data sets are used to train the classifier.

Another factor is feature reduction method used. It has been observed that same classifier gives different results on using different feature reduction methods. Sometimes, dimensionality reduction methods like PCA (Principal Component Analysis), etc give more accurate results whereas sometimes feature selection methods like IG , etc give better results. Even when a particular feature reduction method is used, the classification accuracy may change considerably on changing the parameters in that chosen feature reduction method, for example, in IG method, we select top N features to represent the documents and then the model predicts the category of the unknown instance based on the N-dimensional patterns it has for training data set and the classification accuracy changes a lot sometimes on changing the value of N. For N = 5 (say), the classifier may give excellent accuracy of 95% (say) whereas for different value of N, say N = 20, the same classifier may give very poor accuracy of 30% (say).

So, there are many factors which affect the classification accuracy of a classifier and so, we cannot guarantee a certain accuracy for all cases for any classifier.

## 2.4 BRIEF OVERVIEW OF FEATURE REDUCTION METHODS

In most of the cases, the data to be analyzed or explored contains a large number of variables which are also called the dimensions of the data. Having this high dimensional data is both useful as well as harmful. The good points about having high dimensional data is that we have lot much information about the data and hence we can better analyze and explore the data in other domains of interest as more information leads to better and informed decisions. On the other hand, this high dimensionality of the data poses some serious problems. One of the major problems is high computational burden on the system. Also, the memory demand increases a lot with the increase in the dimensions of the data. Not just this, high dimensions of data degrades even the classification accuracy of a classifier. This is due to the curse of dimensionality. Having high dimensions means there may be words or features that are redundant or irrelevant or for the classification task. Inclusion of such irrelevant words may confuse the learning process of the classifier and this may lead to many problems like overfitting of the data. These irrelevant words or features may degrade the classification accuracy of the classifier in case of supervised learning and may produce clusters of low quality in case of unsupervised learning. Therefore, we need to reduce the size of the feature space so that all these problems are eliminated or at least reduced.

The feature reduction techniques can be broadly classified into 2 categories – feature selection (FS) and dimensionality reduction (DR). We will talk only about feature selection techniques briefly.

### 2.4.1 Feature Selection

In this approach of feature reduction, we select some features from the initial feature space and remove the remaining features as they are irrelevant for the classification purpose. Since we remove some features, there is some loss of information in this method but the information corresponding to useful words is retained and is used for the classification. There

are several approaches to finding the best or optimal set of features from the initial large feature space. These approaches can be classified into 3 types – filters, wrappers and embedded one.

**(A) 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.

However, the filter techniques are efficient in terms of computational cost and memory requirements as they just need all variables' scores or measures and then simply sorting is required to select N features out of the total features. They are further classified into 3 categories – entropies (Information Gain, Odds Ratio, Chi-squared, etc), statistic ones (Pearson Correlation, Fisher Score, etc) and implemented algorithms (Focus, Relief, etc).

**(B) 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. But this approach often produces very high accurate results as it takes into account the biases of the induction algorithm to be used later and so decides the best subset accordingly. Often the computation cost is reduced by following an alternative search process for the optimal feature set, i.e., heuristic search or greedy approach is often used to cut down the run time of finding the optimal subset of features in wrapper techniques.

**(C) 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.

## 2.5 SHORTCOMINGS IN THE INFORMATION GAIN MEASURE

Let us start with the formula for finding IG measure for a word and then we will explain the shortcomings of this measure. The IG of a word or term t can be written as :

$$IG(t) = -\sum P(Ci) \log P(Ci) + P(t) \sum P(Ci|t) \log P(Ci|t) + P(\bar{t}) \sum P(Ci|\bar{t}) \log P(Ci|\bar{t}) \qquad (2.4)$$

- where

  C = set of document collection

  P(Ci) = probability of the $i^{th}$ class or category

  P(t) = probability that the term t appears in the documents

  $P(\bar{t})$ = probability that the term t does not appear in the documents

  P(Ci|t) = conditional probability of the $i^{th}$ class given that the term t appeared

  $P(Ci|\bar{t})$ = conditional probability of the $i^{th}$ class given that the term t has not appeared

Now, we will show the shortcomings in the IG measure and they are as shown below:

**(1) Gives "different" scores to words having same discriminative ability**

Let us consider there are 5 categories or classes – A, B, C, D and E and suppose there are 24 documents belonging to class A and each of the rest classes from B to E have 4 documents each belonging to exactly one of them. So, total number of documents is 40. Now, further suppose that there are 2 words – $W_1$ and $W_2$ - such that $W_1$ is present in every document belonging to category A and is not present in any other document belonging to other categories. Similarly, $W_2$ is present in every document belonging to category B and is not present in any other document belonging to other categories. So, it is clear that both $W_1$ and $W_2$ have same discriminative power as both are present in exactly 1 category out of total

5 categories but IG measures for both of them are different. Let us calculate the information measures for both $W_1$ and $W_2$.

System entropy = -24/40 * log (24/40) – 16/40 * log (4/40) = 0.53
$IG(W_1)$ = 0.53 + 24/40*(24/24 * log 24/24) + 16/40*(16/16 * log 4/16) = 0.29
$IG(W_2)$ = 0.53 + 4/40*(4/4 * log 4/4) + 36/40*(24/36 * log 24/36 + 12/36 * log 4/36) = 0.14

The IG measures for $W_1$ and $W_2$ are 0.29 and 0.14 respectively. So, the IG measure gives more importance to the word $W_1$ because the number of documents belonging to class A is more than those belonging to class B. But more number of documents of class A does not increase the discriminative ability of the word $W_1$, so the IG is biased towards the number of documents. So, if there is another word $W_3$ which has more discriminative ability than $W_2$ (or $W_1$) and suppose the IG measure for $W_3$ is 0.24 (more than that of $W_2$) and suppose we can select only one word further, then $W_1$ will be selected and not $W_3$ even though the discriminative ability of $W_3$ is more than that of both $W_1$ and $W_2$ which share equal discriminative ability but due to higher score given to $W_1$, it will be selected and thus the word having more discriminative ability will be unnecessarily removed and word having less discriminative ability will be included.

**(2) The score of words changes by different factors on increasing the count of documents belonging to a particular category only**

Suppose we have 5 categories – A, B, C, D and E and there are 4 documents under each of these categories or classes. So, total number of documents is 20. Now, further suppose there are 2 words – $W_1$ and $W_2$ - such that word $W_1$ is present in every document that belongs to category A and is not present in any other document belonging to other categories and similarly, $W_2$ is present in every document belonging to category B and is not present in any other document belonging to other categories. Then the IG measure for $W_1$ and $W_2$ can be calculated as :

System entropy = -20/20 * log (4/20) = 0.7
$IG(W_1)$ = 0.7 + 4/20*(4/4 * log 4/4) + 16/20*(16/16 * log 4/16) = 0.22
$IG(W_2)$ = 0.7 + 4/20*(4/4 * log 4/4) + 16/20*(16/16 * log 4/16) = 0.22

Now, suppose we add 20 more documents belonging to category A so that the total number of documents now becomes 40 (total number of documents now belonging to category A becomes 24). This increase in the number of documents in a particular category does not increase or decrease the discriminative ability of any word but still the IG measures of words will change now and that too by different factors as shown below :

System entropy = -24/40 * log (24/40) – 16/40 * log (4/40) = 0.53
$IG(W_1)$ = 0.53 + 24/40*(24/24 * log 24/24) + 16/40*(16/16 * log 4/16) = 0.29
$IG(W_2)$ = 0.53 + 4/40*(4/4 * log 4/4) + 36/40*(24/36 * log 24/36 + 12/36 * log 4/36) = 0.14

So, the first issue is the scores have changed unnecessarily while they should have remained constant and the second issue is they have changed by different factors for even those words which have same discriminative ability as can be seen in case of $W_1$ and $W_2$, both have same discriminative ability but still their scores have changed by different factors. Ideally, there should not be any change in the scores of $W_1$ and $W_2$ by increasing just the count of documents belonging to category A as it does not change the discriminative abilities of either $W_1$ or $W_2$ but still their scores have changed and that too by different factors.

We have therefore devised a new evaluation measure that takes care of both of these shortcomings of IG. Our measure gives same score to both words $W_1$ and $W_2$ if both have same discriminative power ($1^{st}$ shortcoming removed) and also the scores of the words remain constant if there is an increase in the number of documents belonging to a particular category as there is no change in discriminative ability of any word by this change of number of documents ($2^{nd}$ shortcoming removed). We will discuss this in detail later.

# CHAPTER 3

# LITERATURE REVIEW

Most of the data present in various organizations contain few common properties. One of them is the large size of the data, i.e., the data is present in bulk amount. Another common property is that data consists of so many variables, which are often called as dimensions of the data. Having such high dimensional and large amount of data poses some serious threats to its proper analysis and exploration [1]. This is because of the fact that it is practically impossible to manually analyze such a large amount of data and so automated procedures are required to perform the analysis of this data. But even the automated systems have their limited computational power and memory and so processing of such high dimensional data is very computationally expensive. Further, the high dimensional data contains many irrelevant information which confuses the task of analysis and thus leads to other problems like poor classification accuracy (in case of supervised learning) or poor quality clusters (in case of unsupervised learning) or over-fitting problems in general [12] [17].

Therefore, there is a very strong need to reduce the dimensions of the data so that these problems are removed or at least reduced [14]. These variables of the data are called as features in the context of text mining. So, there are various methods to reduce the size of this feature space. Different methods for feature reduction have been used in the past and different methods have their different behaviours. Few methods provide better classification accuracy but are very computationally expensive whereas few are quite efficient in runtimes but do not guarantee good classification accuracy every time [1], [2].

The classification accuracy also varies with the type of classifier used. SVM has been seen to be more accurate classifier than others like Decision Tree, Naive Bayesian, K-NN etc [1]. But then, it also depends on the data sets used for training and test purposes. It has been observed that same feature reduction method performs better than others for one data set and performs worse than others in another data set [2].

Even for a particular classifier, the classification accuracy varies a lot with the feature reduction method used [2], [3]. The wrappers are considered to be better than other feature selection methods like filter methods but they are computationally more expensive [3]. This is due to the fact that they need to call induction algorithm many times during their feature selection process. Among the filter methods, IG has been seen to show better classification results than others but again it depends on data sets and for some data sets, IG shows poor classification results than others [3]. So, we cannot guarantee that a particular classifier or a particular feature reduction method is the best among all as it depends on data sets too [1], [2], [3].

Among the dimensionality reduction techniques, Principal Component Analysis (PCA) method is the most widely used. Unlike feature selection methods, the dimensionality reduction methods do not select a subset of the original features but rather transform the original feature space into a new reduced feature space and thus the new attributes are generated either from a linear or non-linear combination of the original attributes and so no information about any attribute is lost in these methods. Also, PCA method takes lesser time than many feature selection methods to generate same size of feature space [1].

Few authors have even tried to improve the existing feature reduction methods so that the classification accuracy of the classifiers using those methods can be improved. TFIDF algorithm used in the process of classification of textual documents has been shown to have few shortcomings in assigning weights to terms and thus the author has suggested few improvements in the existing traditional approach of TFIDF calculation [4], however, the author asserts that the improved algorithm is much more complex and so, using this improved algorithm may provide a little better results but then it will take much more runtime than the traditional TFIDF approach.

Yet another author has made another attempt in improving the classification accuracy of text classifiers by the introduction of bi-grams [5]. The author has asserted that addition of bi-grams has improved the classification accuracy of the classifier slightly but it has also sometimes worsened the performance of the classifier. The results are usually dataset dependent and so addition of bi-grams sometimes improves the classification accuracy whereas sometimes degrades it too [5].

So, there is lot of research going on in the direction of improving the classification accuracy of the text classifiers. Many authors have suggested many changes in the existing feature reduction methods or even in the whole classification process. But most of the improvements are demanding high computational cost and providing only marginal improvement in the classification accuracy in return and they even degrade the performance of the classifier for some cases as there is a dependency on the data sets too besides other factors, so deciding what improvement to incorporate requires a well thought decision and so one has to see the negative side of introducing those improvements as in most of the cases, the improvement is at the cost of high computational burden on the system and also in many cases, the improvement is not guaranteed and even degradation is possible. So, one has to carefully decide whether to incorporate the improvements or not.

# CHAPTER 4

# PROPOSED METHODOLOGY

We have seen that the classification accuracy of a classifier depends on several factors. The choice of a particular feature reduction method is one of the major factors affecting the classification accuracy of text classifiers. We have also observed that no particular feature reduction method works best in all cases as there are other factors too that govern the classification accuracy. But many times, it has been noted that among other feature selection techniques, IG method gives a good classification performance in many cases. Although, it is not guaranteed that IG method will always give good results only as there is dependency on data sets too [3].

However, IG measure has been widely used for reducing the size of the initial feature space and has given very good results sometimes. But, we observed that there are few shortcomings in this method and we have already highlighted those shortcomings earlier. Now, we will discuss the details of the new method devised by us for performing this feature reduction task. We have named our evaluation measure as Discriminative Power (DP) as it gives a score to each word in the document, which specifies the discriminative ability of that word.

Let us start with the formula for finding the Discriminative Power of a term t. The formula is as follows :

$$DP(t) = \left(\log\left(1 + \frac{N}{N_p}\right)\right)(1 + (P_{max} - P_{min})) + \left(\log\left(1 + \frac{N}{N_a}\right)\right)(1 + (A_{max} - A_{min})) \quad (4.1)$$

Here, we have :

N = total number of *distinct* classes

$N_p$ = no. of *distinct* classes where term t is *present* at least once

$N_a$ = no. of distinct classes where term t is *absent* at least once

$P_{max}$ = max. probability of term t being *present* in one of the $N_p$ classes

$P_{min}$ = min. probability of term t being *present* in one of the $N_p$ classes

$A_{max}$ = max. probability of term t being *absent* in one of the $N_a$ classes

$A_{min}$ = min. probability of term t being *absent* in one of the $N_a$ classes

Now, let us understand the meaning of each variable used in the above Equation (4.1).

**Example 1:** Suppose there are 5 categories or classes – A, B, C, D, and E and suppose there are 4 documents belonging to each of these classes. So, total number of documents is 20. Now, suppose there is a word $W_1$ which is present in 4 documents only and these documents belong to classes A, B, C and D respectively.

So, there are 4 distinct classes in which $W_1$ is present at least once, so $N_p = 4$. Also, there are 5 distinct classes in which $W_1$ is absent at least once, so $N_a = 5$. Also, N = 5, as total number of distinct classes is 5. Now, let us find the probabilities of $W_1$ being present and absent in these $N_p$ and $N_a$ classes.

Prob. of $W_1$ being present in class A = 1/4 (present in just 1 document out of 4 documents)

Prob. of $W_1$ being present in class B = 1/4

Prob. of $W_1$ being present in class C = 1/4

Prob. of $W_1$ being present in class D = 1/4

So, $P_{max} = 1/4$ and $P_{min} = 1/4$

Now,

Prob. of $W_1$ being absent in class A = 3/4 (absent in 3 documents out of 4 documents)

Prob. of $W_1$ being absent in class B = 3/4

Prob. of $W_1$ being absent in class C = 3/4

Prob. of $W_1$ being absent in class D = 3/4

Prob. of $W_1$ being absent in class E = 4/4 = 1

So, $A_{max} = 1$ and $A_{min} = 3/4$

Now, putting these values in Equation (4.1), we get :

$DP(W_1) = 0.725$

**Example 2 :** Suppose there are 5 categories or classes – A, B, C, D and E and suppose there are 4 documents belonging to each category. So, total number of documents is 20. Now, suppose there is a word $W_1$ which is present in every document belonging to class A and is not present in any other document belonging to other categories. So, we have following values for the variables used in Equation (4.1)

N = 5 (since there are 5 distinct classes only)

$N_p$ = 1 (since $W_1$ is present only in one class, i.e., class A)

$N_a$ = 4 (since $W_1$ is absent in 4 classes, i.e., classes B to E)

$P_{max}$ = 1 (since $W_1$ is present in all documents belonging to class A, so prob. = 4/4 = 1)

$P_{min}$ = 1 (since $W_1$ is present only in one class and so its max. and min. Prob. is same)

$A_{max}$ = 1 (since $W_1$ is absent in classes B to E and its probability of being absent in any of these is same as it is not present in any of the documents belonging to these classes and so max prob. of being absent in any of the $N_a$ classes = 4/4 = 1)

$A_{min}$ = 1 (since $W_1$ has same prob. of being absent in any of the $N_a$ classes and this prob. is equal to 1 as it is absent in *all* documents belonging to any of the $N_a$ classes)

So, now putting above values in Equation (4.1), we get :

$DP(W_1) = 1.13$

So, in both the examples, $W_1$ is present in exactly 4 documents only out of total 20 documents but in 1st example, $W_1$ is present across many classes of documents as well it is absent across many classes of documents whereas in 2nd example, $W_1$ is present in only one class of documents but is absent across many classes of documents. So, the discriminative ability of $W_1$ is more in 2nd example and the DP scores in both the examples also are in accordance with this as DP score for $W_1$ is more in 2nd example than in 1st example.

Now, we will show how our new method handles the shortcomings of the IG method as discussed earlier.

**(1) Gives "same" scores to words having same discriminative ability**

Let us consider there are 5 categories or classes – A, B, C, D and E and suppose there are 24 documents belonging to class A and each of the rest classes from B to E have 4

documents each belonging to exactly one of them. So, total number of documents is 40. Now, further suppose that there are 2 words – $W_1$ and $W_2$ - such that $W_1$ is present in every document belonging to category A and is not present in any other document belonging to other categories. Similarly, $W_2$ is present in every document belonging to category B and is not present in any other document belonging to other categories. So, it is clear that both $W_1$ and $W_2$ have same discriminative power as both are present in exactly 1 category out of total 5 categories but IG measures for both of them are different. Let us calculate the discriminative power measures for both $W_1$ and $W_2$ now.

For word $W_1$, we have :
$N = 5$ (since total distinct classes are 5)
$N_p = 1$ (since $W_1$ is present in only class A)
$N_a = 4$ (since $W_1$ is absent in 4 classes, i.e., classes B to E)
$P_{max} = 1$ (since $W_1$ is present in *all* documents belonging to class A, so prob. = 24/24 = 1)
$P_{min} = 1$ (since $W_1$ is present in only one class, so its max. and min. prob. are same)
$A_{max} = 1$ (since $W_1$ is absent in *all* documents belonging to any of the $N_a$ classes, so prob. = 4/4 = 1)
$A_{min} = 1$ (since $W_1$ is having same prob. of being absent in any of its $N_a$ classes and this prob. is equal to 1 as $W_1$ is totally absent in all documents belonging to any of the $N_a$ classes)

So, putting these values in Equation (4.1), we get :

$DP(W_1) = 1.13$

Now, let us calculate the DP value for word $W_2$. For word $W_2$, we have :
$N = 5$ (since total distinct classes are 5)
$N_p = 1$ (since $W_2$ is present in only class B)
$N_a = 4$ (since $W_2$ is absent in 4 classes, i.e., classes A, C, D and E)
$P_{max} = 1$ (since $W_2$ is present in *all* documents belonging to class B, so prob. = 4/4 = 1)
$P_{min} = 1$ (since $W_2$ is present in only one class, so its max. and min. prob. are same)
$A_{max} = 1$ (since $W_2$ is absent in *all* documents belonging to any of the $N_a$ classes, so prob. = 24/24 = 1 or 4/4 = 1)
$A_{min} = 1$ (since $W_2$ is having same prob. of being absent in any of its $N_a$ classes and this prob. is equal to 1 as $W_2$ is totally absent in all documents belonging to any of the $N_a$ classes)

Now, putting these values in Equation (4.1), we get:

$DP(W_2) = 1.13$

So, the discriminative power (DP) values for both words $W_1$ and $W_2$ have come out to be same (1.13 each) whereas the information gain (IG) values for both words $W_1$ and $W_2$ were different (0.29 and 0.14 respectively). Therefore, our new method (DP method) eliminates this shortcoming of IG method and provides "equal" scores to words that have "equal" discriminative ability.

Now, let us show that our new method also removes the 2nd shortcoming of IG method mentioned earlier.

**(2) The score of words "does not" change on increasing the count of documents belonging to a particular category only**

Suppose we have 5 categories – A, B, C, D and E and there are 4 documents under each of these categories or classes. So, total number of documents is 20. Now, further suppose there are 2 words – $W_1$ and $W_2$ - such that word $W_1$ is present in every document that belongs to category A and is not present in any other document belonging to other categories and similarly, $W_2$ is present in every document belonging to category B and is not present in any other document belonging to other categories. Then the discriminative power measures for $W_1$ and $W_2$ can be calculated as :

For word $W_1$, we have :
$N = 5$ (since total distinct classes are 5)
$N_p = 1$ (since $W_1$ is present in only class A)
$N_a = 4$ (since $W_1$ is absent in 4 classes, i.e., classes B to E)
$P_{max} = 1$ (since $W_1$ is present in *all* documents belonging to class A, so prob. = 4/4 = 1)
$P_{min} = 1$ (since $W_1$ is present in only one class, so its max. and min. prob. are same)
$A_{max} = 1$ (since $W_1$ is absent in *all* documents belonging to any of the $N_a$ classes, so prob. = 4/4 = 1)

$A_{min}$ = 1 (since $W_1$ is having same prob. of being absent in any of its $N_a$ classes and this prob. is equal to 1 as $W_1$ is totally absent in all documents belonging to any of the $N_a$ classes)

So, putting these values in Equation (4.1), we get :

$DP(W_1)$ = 1.13

Now, let us calculate the DP value for word $W_2$. For word $W_2$, we have :
N = 5 (since total distinct classes are 5)
$N_p$ = 1 (since $W_2$ is present in only class B)
$N_a$ = 4 (since $W_2$ is absent in 4 classes, i.e., classes A, C, D and E)
$P_{max}$ = 1 (since $W_2$ is present in *all* documents belonging to class B, so prob. = 4/4 = 1)
$P_{min}$ = 1 (since $W_2$ is present in only one class, so its max. and min. prob. are same)
$A_{max}$ = 1 (since $W_2$ is absent in *all* documents belonging to any of the $N_a$ classes, so prob. = 4/4 = 1)
$A_{min}$ = 1 (since $W_2$ is having same prob. of being absent in any of its $N_a$ classes and this prob. is equal to 1 as $W_2$ is totally absent in all documents belonging to any of the $N_a$ classes)

Now, putting these values in Equation (4.1), we get:

$DP(W_2)$ = 1.13

So, both words $W_1$ and $W_2$ have same DP scores as they possess “same” discriminative ability too. Now, suppose we add 20 more documents belonging to category A so that the total number of documents now becomes 40 (total number of documents now belonging to category A becomes 24). This increase in the number of documents in a particular category does not increase or decrease the discriminative ability of any word but still the IG measures of words will change now and that too by different factors as shown earlier (IG values for $W_1$ and $W_2$ were 0.22 before increasing the count of documents of class A and after increasing the count of documents of class A, the IG values for $W_1$ and $W_2$ changed to 0.29 and 0.14 respectively). But our new method (DP method) removes this shortcoming of IG measure and does not change the DP scores of words even after increasing the count of documents of any category. Let us calculate the DP values of words $W_1$ and $W_2$ to show that they have not changed.

For word $W_1$, we have :

N = 5 (since total distinct classes are 5)

$N_p$ = 1 (since $W_1$ is present in only class A)

$N_a$ = 4 (since $W_1$ is absent in 4 classes, i.e., classes B to E)

$P_{max}$ = 1 (since $W_1$ is present in *all* documents belonging to class A, so prob. = 24/24 = 1)

$P_{min}$ = 1 (since $W_1$ is present in only one class, so its max. and min. prob. are same)

$A_{max}$ = 1 (since $W_1$ is absent in *all* documents belonging to any of the $N_a$ classes, so prob. = 4/4 = 1)

$A_{min}$ = 1 (since $W_1$ is having same prob. of being absent in any of its $N_a$ classes and this prob. is equal to 1 as $W_1$ is totally absent in all documents belonging to any of the $N_a$ classes)

So, putting these values in Equation (4.1), we get :

$DP(W_1)$ = 1.13 (same as before)

Now, let us calculate the DP value for word $W_2$. For word $W_2$, we have :

N = 5 (since total distinct classes are 5)

$N_p$ = 1 (since $W_2$ is present in only class B)

$N_a$ = 4 (since $W_2$ is absent in 4 classes, i.e., classes A, C, D and E)

$P_{max}$ = 1 (since $W_2$ is present in *all* documents belonging to class B, so prob. = 4/4 = 1)

$P_{min}$ = 1 (since $W_2$ is present in only one class, so its max. and min. prob. are same)

$A_{max}$ = 1 (since $W_2$ is absent in *all* documents belonging to any of the $N_a$ classes, so prob. = 24/24 = 1 or 4/4 = 1)

$A_{min}$ = 1 (since $W_2$ is having same prob. of being absent in any of its $N_a$ classes and this prob. is equal to 1 as $W_2$ is totally absent in all documents belonging to any of the $N_a$ classes)

Now, putting these values in Equation (4.1), we get:

$DP(W_2)$ = 1.13 (same as before)

So, it is clear now that our new method is very much consistent with the actual discriminative abilities of the words and when the discriminative ability has not changed, the same is reflected by constant DP score of that word. Thus, we have shown that our new DP

method has eliminated both the shortcomings discussed earlier for IG method. Now, we will proceed with the Experimental design setup and will show that our new method improves the average classification accuracy of the classifier and also, the classification accuracy becomes much more stable and has much less deviations from the average value than those present when IG method is used.

# CHAPTER 5

# EXPERIMENTAL DESIGN

We will now discuss the experimental design setup required to obtain the results. The text classification process involves a series of steps like tokenization, removal of stop words, stemming, feature selection and then creating TFIDF matrix and finally creating a prediction model. So, we have implemented a tool in C++ that performs all these processes on the input data set that contains both category as well as description for each instance. Also, we have used Nearest Neighbour algorithm for the prediction of the category of the unknown instance present in the test data set. Although the classification accuracy of Nearest Neighbour classifier is not that great but still we have used it as it is easy to implement and moreover we are not concerned with the absolute accuracy but mainly with the comparison of accuracy of the classifier with respect to IG method and our own new discriminative power method.

## 5.1 DATA SETS

We have used PITS-A data set that has been supplied by NASA's software Independent Verification and Validation (IV & V) Program. The problems or challenges concerned with human rated systems and robotic satellite missions were collected for about more than 10 years and have been included in this data set. The data sets contain fault reports. A fault report includes the description of the faults, their ID and their associated severity level. According to NASA's engineers, the faults can be divided into 5 severity levels which are very low, low, medium, high, and very high.

The faults having severity level as very high are the most critical faults as they threaten the security and the safety. Also, such faults are impossible to recover. Therefore, in the empirical study, the severity level 1 has not been considered and only next 4 severity levels

have been considered, namely, severity 2 (high), severity 3 (medium), severity 4 (low) and severity 5 (very low).

The data set consists of total 960 faults which are of different severity levels (from 2 to 5). Their distribution into these 4 severity levels is as follows :

**Table I : Severity levels and associated fault count in Data Set**

| Severity Level | No. of faults |
|---|---|
| 2 | 320 |
| 3 | 375 |
| 4 | 239 |
| 5 | 26 |

## 5.2 VALIDATION METHOD

For validation purpose, we have used k-fold cross validation method in which the whole data is divided into k folds and one fold is used as the test data and remaining (k-1) folds are used as training data and then the same process is repeated k times such that every time a new fold is used as the test data. This ensures every fold or every record has been used for both training as well testing purpose. In our case, we have taken the value of k as 10. So, we have divided our data set containing 960 fault descriptions into 2 sets - training set and testing set and this division has been done 10 times so that each record gets used both as training and testing data. Since the data set contains faults belonging to 4 severity levels or classes, we have tried to maintain the ratio of faults belonging to these classes same in both training and test data sets. Therefore, the distribution of faults into 4 severity levels in the training and test data sets is similar to the distributions shown in table II and table III respectively.

**Table II : Severity levels and associated fault count in Training Data Set**

| Severity Level | No. of faults |
|---|---|
| 2 | 288 |
| 3 | 337 |
| 4 | 215 |
| 5 | 24 |

**Table III : Severity levels and associated fault count in Testing Data Set**

| Severity Level | No. of faults |
|---|---|
| 2 | 32 |
| 3 | 38 |
| 4 | 24 |
| 5 | 2 |

## 5.3 PERFORMANCE METRICS

The performance measures are required to evaluate the working of any model or method. Since our main aim is to improve the classification accuracy of the classifier by using our new feature selection method, which we have named as Discriminative Power (DP) method. The classification performance can be measured by a number of different performance metrics, like accuracy, precision, recall, F-measure, etc. We have used all these 4 measures to evaluate the performance improvement of the classifier – accuracy, precision , recall and F-measure.

Since F-measure combines the effects of both precision and recall (as it is harmonic mean of these two), we have shown results in terms of F-measure. We have also compared results on the basis of accuracy achieved (The "accuracy" is defined as the ratio of the number of *correct* classifications to the total number of classifications made by a classifier). The runtimes of both the methods (*IG* and our new method *discriminative power)* have also been compared. Also, since there are 4 classes in our data set, we have calculated the precision, recall and F-measure for each class separately and then the average of these values is taken as the final values of these measures. In finding the precision, recall and F-measure for a particular class (say, class A), we have treated all other classes as "Not Class A" and similarly, we have found these measures for all classes and then the average of those values has been taken as the final values of these measures.

# CHAPTER 6

# RESULTS AND ANALYSIS

We created 2 versions of our tool which were exactly the same except the feature selection method used was different in both of them. In one version, we implemented *IG* as the feature selection method whereas in the other version, we implemented our own method *discriminative power* as the feature selection method. We then tested both the versions on different training and test data sets and compared the performance in both the cases.

As discussed earlier, we have used accuracy, precision, recall and F-measure as the performance metrics to evaluate and compare both the methods. However, we will show results in terms of accuracy and F-measure only as precision and recall are already taken into account in F-measure as it is the harmonic mean of these two.

The feature selection methods usually provide scores to the features or words present in the documents and then we choose top N features out of them which are most useful for the classification purpose and ignore or remove the rest of the features. We have used the term *feature count* to represent the number of features that have been chosen as best N features by the used feature selection method. The performance of the classifier varies a lot with the change in the value of N, i.e., feature count, therefore we have calculated the performance metrics for both the methods over a range of N. In our experiment, we varied the value of feature count (N) from 5 to 100, with step-size of 1 and recorded the various performance metrics. The results we obtained are as shown below :

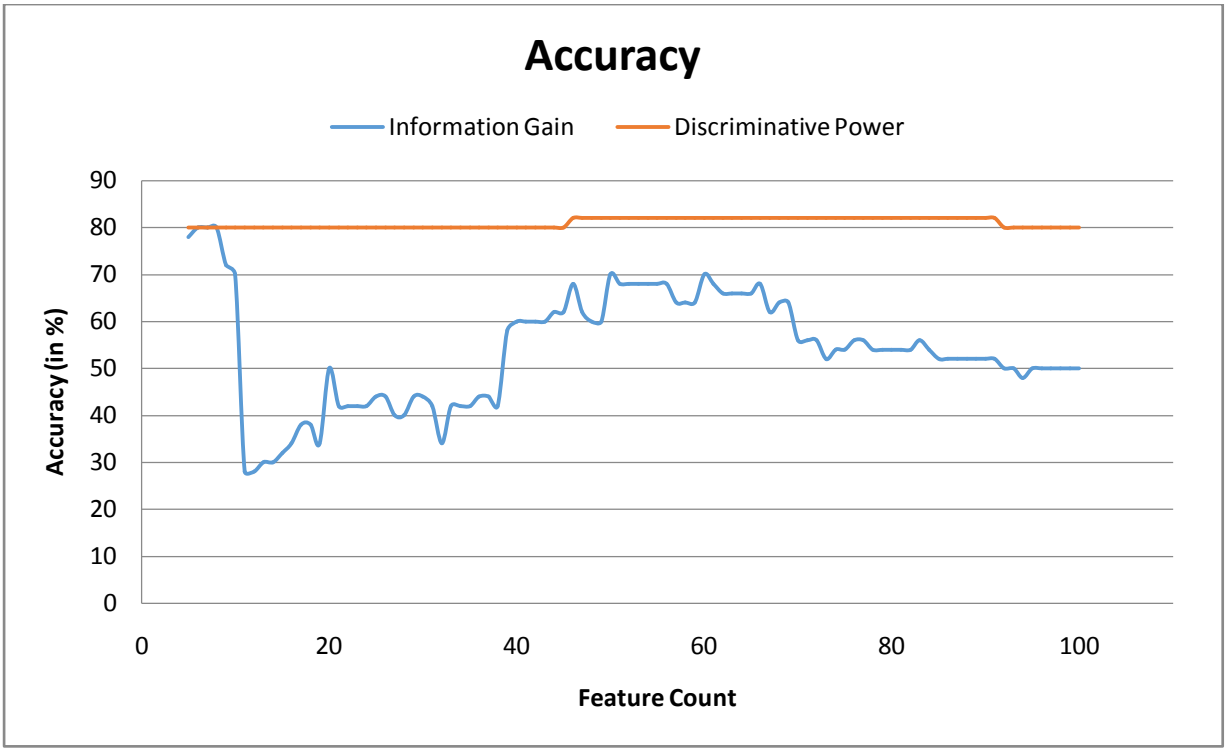


**Fig. 6.1 Comparison of feature selection methods in terms of Accuracy**

When we used *IG* method as the feature selection method, then the best accuracy achieved was 80% at the feature count value of 6. Also, the average accuracy achieved when feature count was varied from 5 to 100 was 54.08%. But when we used our *discriminative power* method as the feature selection method, the best accuracy achieved was 82% at the feature count value of 46. Also, the average accuracy achieved when feature count was varied from 5 to 100 was 80.96%. So, our method has improved the average accuracy of the classifier by 49.7% as it has increased from 54.08% to 80.96%. Also, the maximum accuracy achieved is more in our DP method than in IG method.

Moreover, the accuracy-graph is much more stable (less deviations) when our method is used. As Fig. 6.1 shows, the accuracy is varying a lot with the change in the feature count when IG method is used whereas when our discriminative power method is used, there is almost no change in the accuracy when feature count varied from 5 to 100, so our method gives much more stability in the accuracy of the classifier than that given by IG method.

However, when we changed the feature count value from 5 to 100 at a step-size of 5 and calculated the accuracy given by each of the two methods at different values of feature count, then we observed that for some data sets, the highest accuracy achieved using *IG*

method was slightly more than that achieved using our *discriminative power* method. But in most of the cases, the average accuracy achieved using IG method could not beat the average accuracy achieved using our method. Therefore, we can say that except for some cases, our *discriminative power* method gives improvements in best accuracy achieved as well as average accuracy achieved in the classification process. Let us now see the comparison of both methods in terms of F-measure.

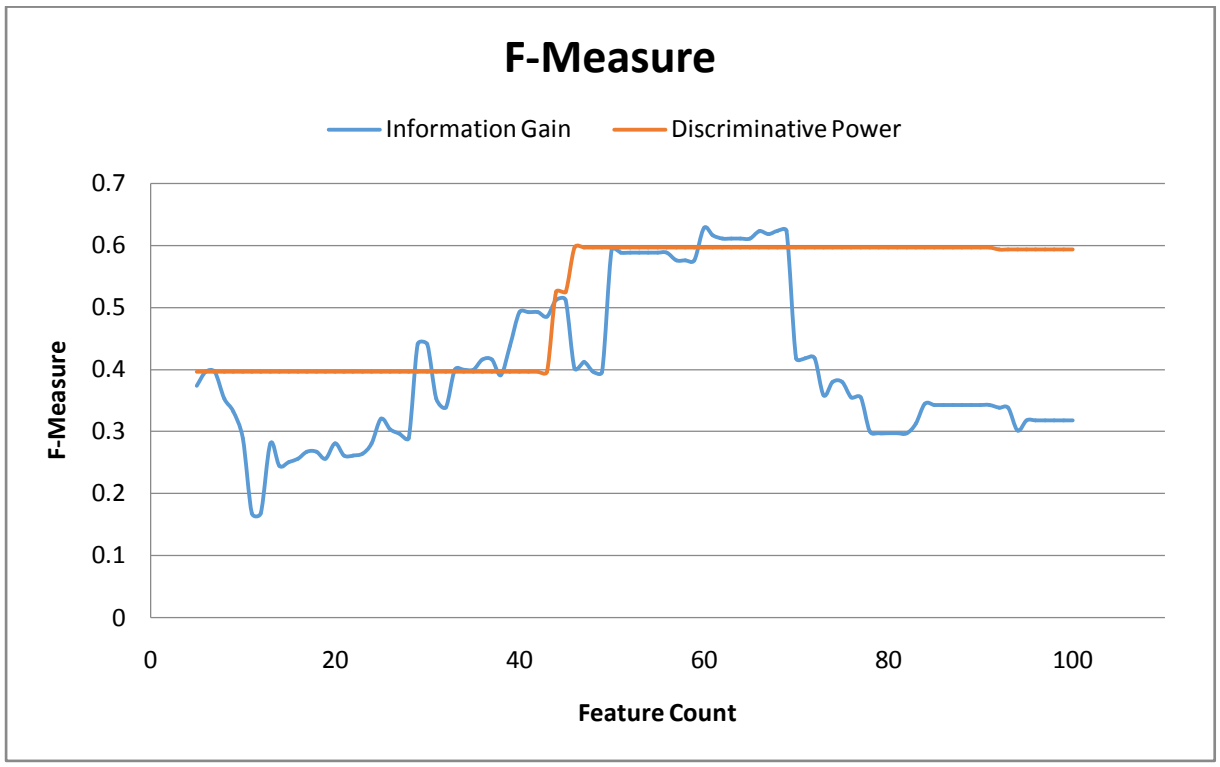


**Fig. 6.2 Comparison of feature selection methods in terms of F-measure**

As also mentioned earlier that there are 4 categories or classes in our data set, so we have calculated the precision, recall and then F-measure for each of the classes separately and then the average of these values has been taken as their final measures and we have compared the *IG* method and our *discriminative power* method based on those average values. So, the F-measure values shown in Fig. 6.2 are the average values of F-measure taken over all the 4 classes present in the data set.

When we used *IG* method as the feature selection method, the highest F-measure achieved was 0.63 at the feature count value of 60. Also, the average value of F-measure when feature count was varied from 5 to 100 came out to be 0.40. But when we used our *discriminative power* method, the highest F-measure achieved was 0.60 at the feature count

value of 46. Also, the average value of F-measure when feature count was varied from 5 to 100 came out to be 0.51. So, our method has again performed better than IG method in terms of F-measure also as the average value of F-measure has been improved by our method from 0.40 to 0.51. Thus, the improvement in terms of F-measure is 27.5%. As Fig. 6.2 shows, the F-measure is varying a lot with the change in the feature count when IG method is used whereas when our discriminative power method is used, there is almost no change in the F-measure when feature count varied from 46 to 100, so our method gives much more stability in the F-measure of the classifier than that given by IG method.

However, when we changed the feature count value from 5 to 100 at a step-size of 5 and calculated the F-measure given by each of the two methods at different values of feature count, then we observed that for some data sets, the highest F-measure achieved using *IG* method was slightly more than that achieved using our *discriminative power* method. But in most of the cases, the average F-measure achieved using IG method could not beat the average F-measure achieved using our method. Therefore, we can say that except for some cases, our *discriminative power* method gives improvements in best F-measure achieved as well as average F-measure achieved in the classification process.

Also, many researchers have attempted some kind of improvements in the existing algorithms or methods to improve the classification accuracy of text classifiers but most of the times, those improvements have been complex to implement and so the run time of the improved process or method increases by a great margin which is not desirable. However, our new method (discriminative power) is not complex to implement and can be easily implemented with similar run time as the earlier unimproved version. We have calculated the run time of the classification process in both of the methods (*IG* and our own method *discriminative power*) and then we compared both the run times with each other and the results are as shown below :

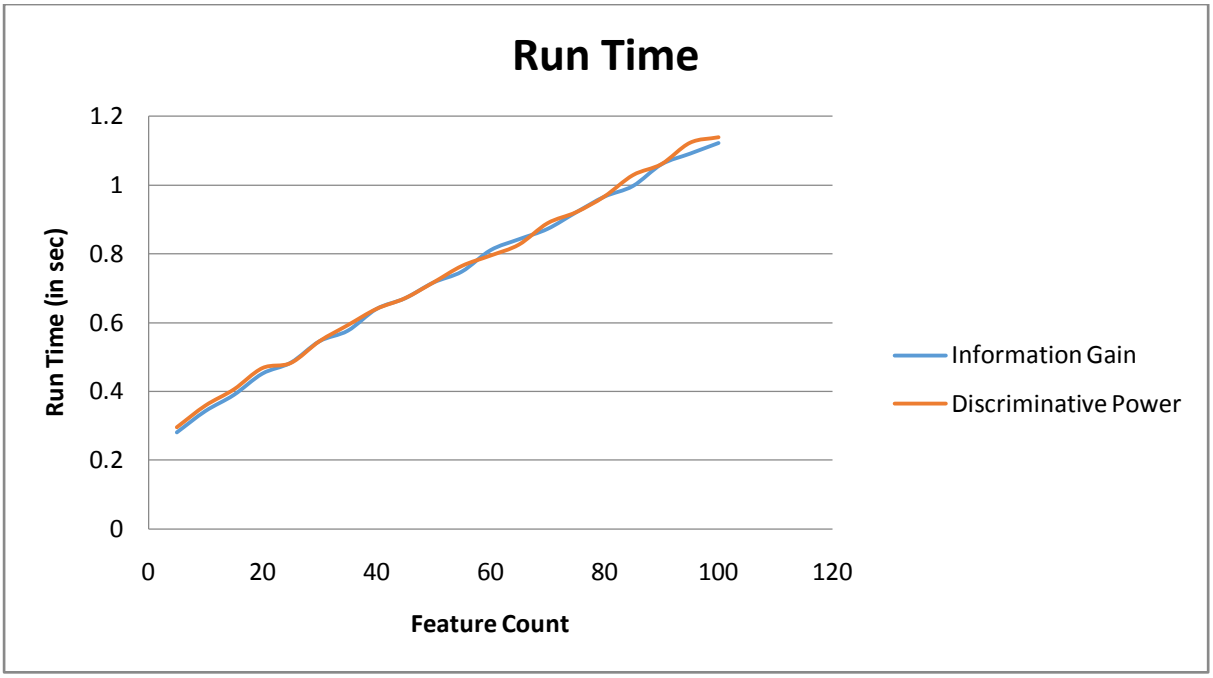


**Fig. 6.3 Comparison of feature selection methods in terms of Run Time**

As can be seen in Fig. 6.3, the run times in both the methods are almost overlapping, which confirms that our new method is not at all complex to implement and takes almost same time as that taken by the existing IG method. However, as the Fig. 6.3 shows, the time taken by our method is slightly more than that taken by IG method at some places whereas at other places, the reverse is happening, i.e., the time taken by our method is slightly less than that taken by IG method. But overall, we can say that both methods take almost same run times for their feature selection process, and so our new method provides improvements in the classification accuracy of the text classifiers at no additional cost of run time or anything.

# CHAPTER 7

# CONCLUSION

There is a strong need of automated text classification in today's scenario as the data is growing at an alarming rate and this huge data cannot be analyzed manually and so automated processes are required to analyze and explore this large amount of data. But then the classification accuracy of automated classifiers becomes the next challenge and as we have discussed earlier, there are many factors that affect the classification accuracy of a text classifier. One such important factor is the feature reduction method used. Different feature reduction methods have different effects on the classification accuracy and run time of the classification process. Some methods promise better accuracy but take much longer runtimes whereas some methods are quite efficient in time but do not guarantee good accuracy always and may give very degraded classification performance sometimes [1], [3].

However, Information Gain (IG) method is one of the most popular and widely used feature selection method but we observed few shortcomings in this method and thus devised a new method which eliminates these shortcomings completely. Our new method provides scores to features or words in accordance with their discriminative ability only and that is why we have named our method as Discriminative Power (DP) method. We have also shown how our DP method eliminates the shortcomings of the IG method. We have also tested both IG and our DP method on different data sets and evaluated the performance of the classifier in terms of accuracy, and F-measure and we found that our DP method improves the classification accuracy of the classifier and the improvement percentage is also good as we have observed that there has been about 50% improvement in terms of accuracy and about 28% improvement in terms of F-measure. However, in few cases, IG method performs slightly better than our DP method. This is mainly due to dependency of the classifier on data sets too. Also, we observed that our DP method provides much more consistent (or stable) classification performance than that provided by IG method which gives much more varying

classification accuracy when feature count is changed. Also, our DP method is easy to implement and takes almost same run time as that taken by IG method.

Therefore, we can say that our new method (Discriminative Power method) is a better feature selection method than IG method and provides improvement in the classification accuracy of a text classifier at no additional cost of run time or any other thing.

# 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 23-25, 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, 2008, pp. 90-105.

[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, (AAAI-1991), pp. 547-552.

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

[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.

[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, 2003, pp. 1157-1182.

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

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

[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. O'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", in *Proceedings of the Eleventh International Conference of Machine Learning*, 1994, pp. 121-129, Canada.