OPTIMIZATION OF TEXT CLASSIFICATION USING SUPERVISED AND UNSUPERVISED LEARNING APPROACH

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Suresh Kumar

(Roll No. 851232009)

Under the supervision of:

Dr. Shivani Goel

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

Certificate

I hereby certify that the work which is being presented in the Thesis Report entitled, **"Optimization of Text Classification Using Supervised And Unsupervised Learning Approach"**, submitted by me in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering department of Thapar University, Patiala is an authentic record of my own work carried out under the supervision of Dr. Shivani Goel and refers other researchers work which are duly listed in reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other University.

Suresh Kumar
(Suresh Kumar)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Shivani Goel
(Dr. Shivani Goel)
Assistant Professor
Computer Science and Engineering Department
Thapar University,
Patiala

Countersigned by

(Dr. Deepak Garg)
Associate Professor & Head
Computer Science And Engineering Department
Thapar university,

(Dr. S.S. Bhatia)
Dean (Academic Affairs)
Thapar University,
Patiala

i

Acknowledgement

I express my sincere and deep gratitude to my guide Dr. Shivani Goel, Assistant Professor in Computer Science & Engineering Department, for the invaluable guidance, support and encouragement. she provided me all resource and guidance through thesis work.

I am thankful to Dr. Deepak Garg, Head of Computer Science & Engineering Department, Thapar University, Patiala for providing us adequate environment, facility for carrying thesis work.

I would like to thank to all staff members who were always there at the need of hour and provided with all the help and facilities, which I required for the completion of my thesis.

I would also like to express my appreciation to my best friends Arvind, Lalit for motivation and providing interesting work environment, it was great pleasure in work with them during thesis work.

At last but not the least I would like to thank God and mine parents for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

ABSTRACT

With the rapid growth of the Internet and the raise in on-line information, the technology for effective retrieval and categorization of large amounts of text data plays a vital role in text mining. In the 1990s, the concert of computers enhanced harshly and it became feasible to handle huge amount of text data. This has led to the utilization of machine learning approach, which is a method of exploring the structure and learning of algorithms that can be trained from and make predictions on data given in a category label. This approach provides brilliant precision, reduces effort, and ensures traditional utilization of resources. Due to rapid spread and high dimensionality of online information, efficient retrieval of some exact information is complicated without good indexing and summarization of document content. Therefore document categorization or classification may be the result to successfully handle and manage such large amount of text.

Text Classification, also known as text categorization, is the task of automatically allocating unlabeled documents into predefined categories. Text Classification means allocating a document to one or more categories or classes. The ability to accurately perform a classification task depends on the representation of documents to be classified. Text representation transforms the textural documents into a compact format. Text Classification plays an important role in information mining, summarization, text recovery and question-answering. It uses several tools from information retrieval (IR) and Machine Learning. Here we are reviewing the effectiveness of different supervised and unsupervised learning approaches in text classification.

TABLE OF CONTENTS

LIST OF FIGURES

Caption **Page No.**

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Machine Learning(ML)

ML is a study of systems that can learn from data. Machine learning is employed in a wide range of computing tasks like search engines, computer vision, optical character recognition (OCR) etc. The tasks under machine learning can be of several forms i.e. supervised or unsupervised. If the computer is presented with example inputs and their desired outputs, given by a "teacher", then it is an example of supervised learning. In case of email filtering onto spam messages or ham messages, the supervised learning algorithm is presented with email messages labeled for an example as "spam" or "not spam". Then the machine learning produces a computer program that can correctly label new messages as either spam or not. If the examples are not provided to the program as training examples, then it is unsupervised learning. It is used for applications where one is to discover hidden patterns in data. No labels are given to the learning algorithm, leaving it on its own to groups of similar inputs.

1.2 Text Mining

In text mining process, initially a number of documents are gathered and a tool is used to extract the particular information or document and preprocess it. In text analysis phase, technique used is repeated until the relevant information is extracted. Data mining tools organize the structured data from the databases only whereas text mining extract information from semi-structured and structured datasets such as HTML files, e-mails etc. Therefore text mining is better option to handle or organize online data for companies.

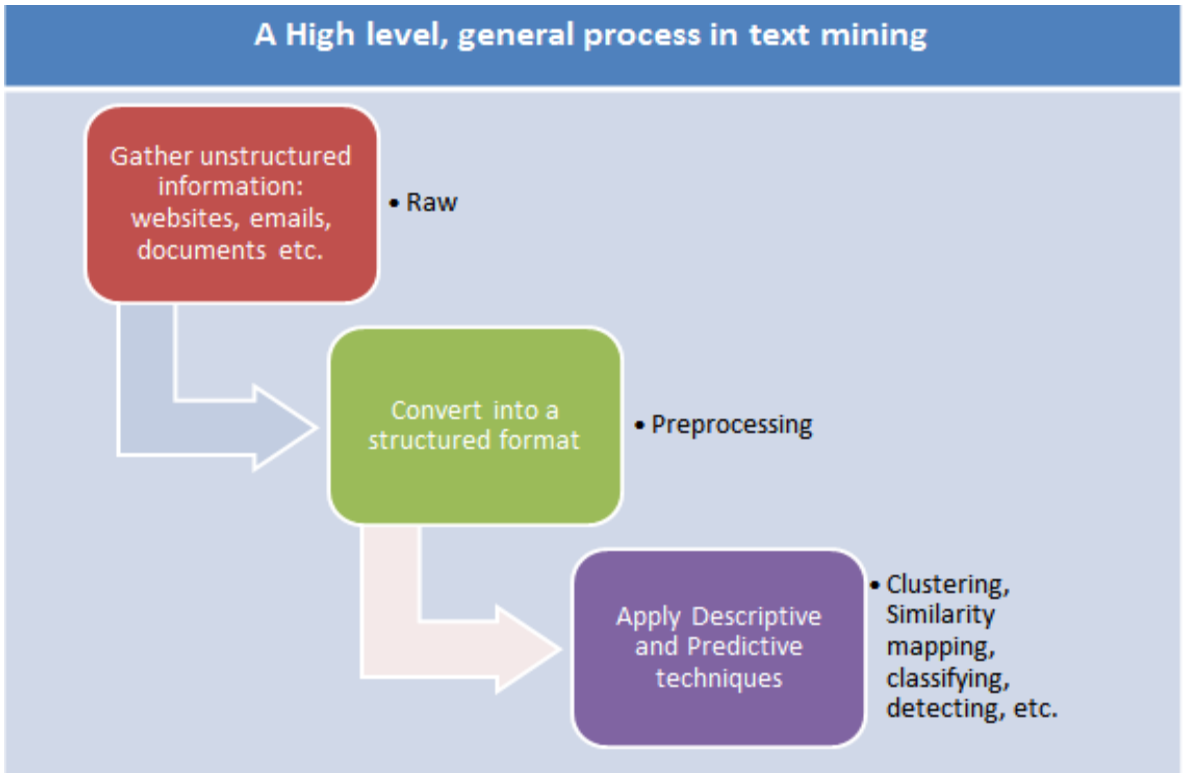

Fig 1.1: Text Mining

Text Mining tasks consist of:-

- Text Clustering
- Concept/Entity Extraction
- Text Summarization
- Text Classification

1.2.1 Text Clustering:- Text Clustering (or Document Clustering) is defined as a procedure of analysis of cluster to textual documents. It has function in automatic document association, topic extraction and fast information recovery. Text clustering can be of two types, online and offline. Online is generally controlled by efficiency problems as compared to offline applications.

1.2.2 Concept/Entity Extraction:- Concept/Entity Extraction is a subtask of structured data mining that aims to organize the elements in text into pre-defined categories. The categories could be the names of persons, areas, organizations etc.

1.2.3 Text Summarization:- Text summarization (or Automatic summarization) is the procedure of sinking a document from the data to generate a synopsis that contains the main

significant points of the real document. With the rapid growth of online text or data, text summarization plays a vital role in text mining. A search engine such as Google is an example of text summarization technology.

1.3 Text Classification

Text classification, also known as text categorization, is defined as the task of defining unlabeled documents into predefined classes automatically.

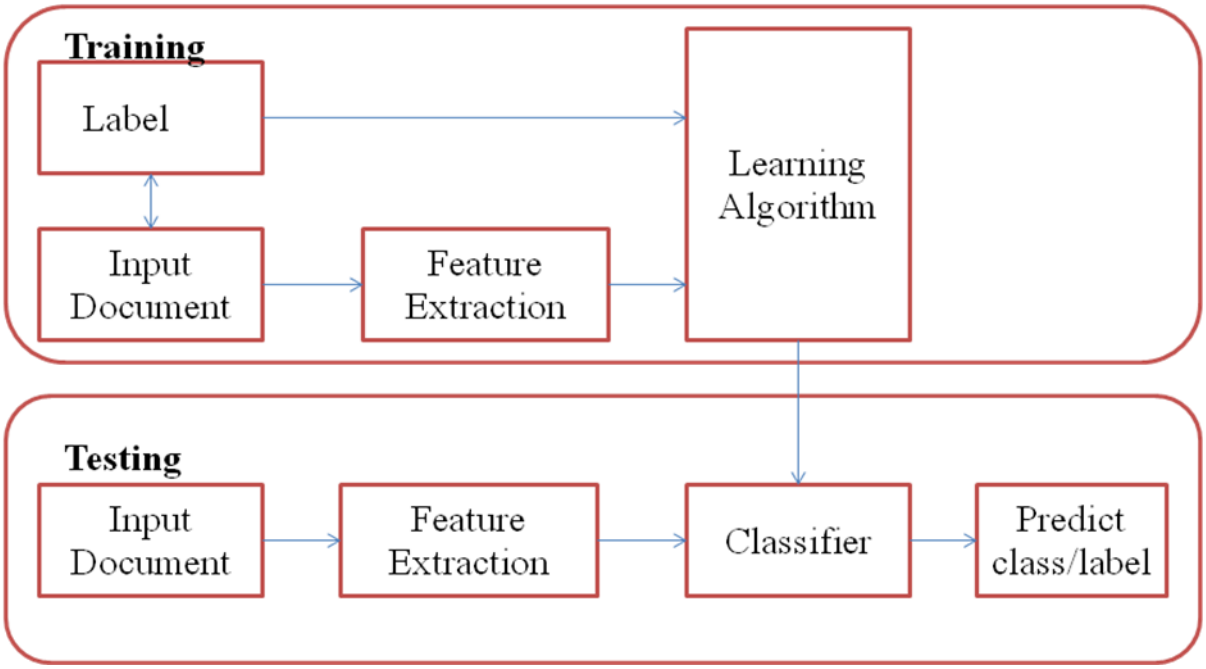

Fig 1.2: Text Classification

1.3.1 Mathematical Definition for Text Classification

Text Classification, also known as text categorization, is the task of defining unlabeled documents into predefined classes automatically. The capability to correctly execute a categorization task depends on the demonstration of documents to be categorized. In general, text classification may be defined as

$$x = (x1, x2, - - - - - xn)$$

Each document is expressed as vector in text classification and each document vector has two feature values: Frequency of a term in a document i.e. number of times the term appears in a document and frequency of term in collection of documents. For example, consider the two documents given below:-

"Olympics will be held in Chicago" (document 1)

And "Cricket is the champion of games" (document 2)

are expressed as $x1$, $x2$ using the four word features "Olympics", "Chicago", "Cricket" and "games".

In the given example the document is expressed by a four dimensional features. However, it is desirable to use at least 10,000 features, or as many as possible, to classify various documents at a high accuracy. Feature selection is a typical process for dimensionality reduction.. One of the most important theme sustaining data mining is transformation of textural text into numerical vectors i.e. text representations. Generally text representations include two types of works: indexing and term weighting. Indexing is done to allocate indexing terms for documents whereas term weighting is done to allocate weight to every term of the document which measures the significance of that term. Presently, for text classification, there is a lot of term weighting methods which are used for allocating weight to every term. Text categorization has borrowed the term weighting schemes from IR field, such as term frequency, term frequency-inverse document frequency(TF-IDF) and its variants. Feature representation is a conversion method that allows documents to be interpreted by classifiers and this method is also called as Term Weighting.

Table 1.1 Term Weighting Methods

Method	**Description**
Binary	Boolean Logic Representation 1= Present, 0 = Not Present
TF(Term Frequency)	Frequency of a term in a document i.e no of times the term appears in a document.
DF(Document Frequency)	Frequency of term in collection of documents.
D	Number of all documents
d_i	Document containing item i

In text categorization, documents are classified into a fixed number of predefined categories. Each document can be in exactly one category, multiple categories or no category at all. Using machine learning, we can learn classifiers from examples which perform the category assignments automatically[20]. This is a supervised learning problem. In order to maintain a small number of features, vector representation for the text in a document can be used. Various measures are used for feature retrieval for text stored as vector space:

Term frequency(TF) : It is the number of times a given term(n_i) appears in that document.
Inverse document frequency(IDF) : It is defined as the logarithm of the number of all documents divided by the number of documents containing that term.

Then TF-IDF = TF * IDF

A high weight in tf–idf is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents. For each document d_j and keyword k_i "tf-idf" is defined by the weight "w". Presently, there are many term weighting methods which are used for text classifications. Feature representation is a transformation method that allows documents to be interpreted by classifiers and this method is also called as Term Weighting.

CHAPTER 2

LITERATURE SURVEY

There are two tasks of text representation i.e term weighting and indexing as described by Bijalwan et.al [1]. The main objective of this paper is to study the efficiency and effectiveness of different text representation methods. For text representation two main tasks are indexing which is mainly concerned with statistical and semantic quality and weighting which is mainly concerned with term frequency (TF) and inverse document frequency (IDF).

The main goal of this paper is to study the efficiency of different indexing methods in text categorization. In this paper the author conducted the experiments to check the performance of the document representation methods i.e TF_IDF, Latent Semantic Indexing(LSI) and multi-word for text representation. Two documents i.e Chinese and English were used to evaluate the text representations method for text classification. The experimental results demonstrated that in text classification, LSI performed very well than other methods in both document collections. Also, while retrieving English documents LSI showed the best performance. The results has shown that LSI has both positive semantic and statistical quality and is different with the assert that LSI cannot produce discriminative power for indexing.

Text categorization is the task of automatically allocating unlabeled documents into predefined categories. If a document or text belongs to exactly one class or category, it is known as single-label classification task and if a document or text belongs to more than one or more class or category, it is known as multi-label classification task. In KNN based learning approach used for text categorization by Zhang et al. , first the documents are classified using KNN based machine learning approach and then compared with Naïve Bayes and Term-graph approach by returning the most relevant documents[2]. In this paper authors concluded that KNN showed maximum accuracy as compared to the Naive Bayes and Term-Graph.

The disadvantage of KNN classifier is that its time complexity is high but it gives enhanced accuracy than others. In this paper the authors rather than implementing the traditional Term-Graph used with AFOPT used Term-Graph with other methods. This hybrid approach showed a better result than the traditional combination. Finally author made an information

retrieval application using Vector Space Model to give the result of the query entered by the client by showing the relevant document.

A simple, scalable and non-parametric approach for short text classification was proposed by Sun[3]. In general short texts are much nosier, shorter and sparser therefore to improve short text representation author proposed to trim a short text categorization. This approach mimics human classification process for a piece of short text like tweets, status updates, and comments. Short text classification is optimized by learning multi-granularity topics by Chen et al.[4].

The authors proposed a new algorithm using multi-granularity to create features for short text. Due to sparsity and shortness of short text it was different from usual documents. Two major approaches used for short text classification to improve the representation of short text were:-

i) Fetch the appropriate information of short text to directly add more text.

ii) Drive latent topics from existing large corpus.

Second approach was found to be more well-designed and well-organized in most cases. In this paper author proposed a new method at leverage topics at multiple granularity using LDA (Latent Dirichlet Allocation) for short text classification. The author compares the proposed algorithm with the state of the art baseline over web-snippet data set (one open data set) through two type of classifiers: MaxEnt (Maximum Entropy), SVM(Support Vector Machine). An experimental result showed that proposed algorithm performed better and appreciably reduced the classification errors by 16.68% and 20.25% in the same way.

A new feature selection method for text classification using a supervised term selection approach was proposed by Basu et al.[5]. In this paper term significance (TS) a feature selection technique was compared with CHI, IG & MI. The proposed approach derived a similarity score between a term and a class and then ranked the terms according to their scores over all the classes. The experimental results showed that the proposed TS can produce better classification accuracy even after removing 90% unique terms.

A new scheme was proposed for multi class text classification by Ko et. al [6]. The main purpose was to improve text classification by efficiently applying class information to a term

weighting scheme. Then it was compared to the TF-IDF and previous methods. As a result the proposed scheme utilized class information for term weighting for text classification and performed consistently on the data sets and KNN and SVM classifiers.

In a paper by Kiritchenko and Matwin, a learning technique was introduced that decreases the effort needed in applying machine learning[7]. Main Problems in text classification are lack of labeled data and the cost required for labeling the unlabeled data. In this paper Classification is done on E-mail domain with Co-training algorithm that uses unlabeled data along with a small number of labeled examples. In this paper, the authors firstly tested SVM classifier on a Labeled edition of unlabeled data and then Naive Bayes classifier was tested. As a result SVM performed very well in comparison with Naive Bayes. Experimental results also showed that the performance of containing depends on learning method that it uses.

Pang and Jiang purposed a generalized cluster centroid based classifier(GCCC) to use KNN and Rocchio via a clustering algorithm[8]. In this paper, an algorithm was combined with Rocchio and KNN to make a generalized cluster centriod based model respectively to ensure the scalability and applicability of the GCCs model. Experimental results showed that GCCC showed stable and favorable performance than KNN and Rocchio classifier. One drawback of GCCC was that it was more time-consuming than KNN and Rocchio.

Samuel Danso et al. have done a relative study for the categorization of verbal autopsy text in three ways i.e. feature representation, effect of reducing features and Machine learning algorithms[9]. The author exhibited that normalized TF and standard TF-IDF achieved comparable performance across different classifiers. Finally author demonstrated the effectiveness of applying semi supervised feature reduction approach to increase accuracy and SVM (Support Vector Machine) algorithm found to be the best algorithm than other algorithms.

In this paper by Larochelle et al., the authors used an individual non-linear Classifier (RBM) for the classification[10]. Firstly the classifier RBM (Restricted Boltzmann Machine) is trained through different strategies and then tested with two classifiers i.e. LOG and NNet. In

this paper RBM was compared with two different classifiers on multitask datasets. As a result RBM classifier gave best performance on all datasets than other classifiers.

Baccianella et al. presented a method for performing feature selection using micro documents for ordinal text classification[11]. Most probably used methods for feature selection are information gain, chi square, mutual information etc. which are based on binary information. i.e. the term is either presented or absent in document. These methods don't provide information about how frequently the term occurs in a document. To overcome this, author presented a method for feature selection using micro documents i.e. Dividing each training document of length into "n" training documents. The author used ordinary text classification in which four feature selection methods were used along with two learning algorithms. As a result the proposed method showed substantial accuracy improvements for all combinations of data set, feature selection function, and learning algorithms. In this paper the author presented four novel feature selection techniques proposed for ordinary classification. Further the author test these in two data sets i.e. Trip advisor- 15763 and Amazon – 83713. In this paper, the author tested proposed methods with two different SVM- based learning algorithms for ordinal regression: - SVR and SVOR. The experimental results showed that Amazon- 83713 dataset gave stable performance across the reduction levels range.

Xia et.al made a comparative study of efficiency[12]. Firstly for sentiment classification, 2 types of feature sets are selected. These features sets were based on part of speech and word relation feature set. After that SVM, maximum entropy and Naïve Bayes classifiers were used on these features sets. Then for three ensemble strategies three types of ensemble methods were allocated. Finally experiments were done on 5 data sets which are movie, book, DVD, electronics and kitchen as they all have 1000 positive and negative reviews. After that experiments were conducted on data sets using 5 cross validation methods. In which 4 folds were used for training, one fold was used for testing; the overall performance was reported on the basis of accuracy. The experimental results showed that ensembling of the feature sets and classification algorithm gave better results.

Nigam et.al proposed an algorithm based on combination of EM and naïve based classifiers from labeled and unlabeled documents[13]. The paper showed the precision and accuracy of learned text classifiers. The algorithm firstly trained the classifiers using labeled documents

and tested the classifiers through unlabeled documents. The experiments were conducted on three data sets: 20 newsgroups, web KB, Reuters. Results showed that combination of labeled and unlabeled documents using EM algorithm gave better accuracy than Naïve Bayes. In this way combining the both labeled and unlabeled documents reduced the errors by 30%.

Irani et al. perfomed the study of trend stuffing i.e. spammers who post unrelated tweets to pictures on twitter[14]. The author studied the use of TC over 600 trends which contains 1.3 millions tweets and their related pages. In this paper features were reduced by using information gain method. The authors have done a comparative study of naïve Bayes and decision spumes classifiers over individual feature set. After that the authors combined the classifiers for tweets, web pages associated with tweets. As a result classifier J48 achieved high accuracy and F1 measure. The limitation of this classifier was that it's required training time was significantly smaller. According to proposed approach, the authors divided the trend stuffing into 3 steps:- firstly to study the text categorization based only on 140 or less characters of tweets which helped to determine how useful the text of tweets were to create a classifier model. Secondly, author investigated the text categorization based on web pages associated with tweets which helped to determine that whether the trend belonged to tweet or not. Thirdly, combining text classification predictions from tweets was compared using AND, OR operators.

Nizamani et al. did a comparison of advanced topics likes multi-objective and ensemble-based evolutionary clustering[16]. A novel approach was proposed for large scale hierarchical text classification by Ha-Thuc et al. [17]. It didn't require any labeled data. In the proposed approach, meaning of category was not defined by human labeled documents but by its descriptions, ascendants and descendents. In this paper the author exploited the hierarchy to construct a query for category. The query must be enriched and context aware. Firstly, we used its ancestors to define a context and resolve possible errors and the query was submitted to the web search engine to get relevant information. Secondly, author extracted a language model for each category and excludes noise i.e. non relevant information. Finally using language models test documents were classified into categories. The author proposed a novel approach for hierarchical text classification instead of using human labeled documents. Authors have done experiments on IPTC taxonomy having 1131

categories to check the effectiveness of proposed approach. As a result, language model extracted by proposed approach outperforms than models extracted maximum likelihood. It was able to reduce noise and was able to identify the relevant information.

In their work, Xu et.al applied the KNN method for classification of spam messages. In order to determine the subjects of spam messages, clustering was used on a multi-document summarization method [18].

CHAPTER 3

PROBLEM FORMULATION

3.1 Gap Analysis

- In previous works text data is huge due to which sparsity of features is high which increases the overhead of process. This problem is solved by Linear SVC with results optimized.
- In previous work classifier use very less data set as these classifiers are not able to give boundary condition. This problem is solved by Support vector machine (SVM) because it gives extra margin in boundary condition entities.
- In previous work classifiers are not able to properly classify multi class data because class imbalance is not removed. This problem is solved by SGD classifier.
- In previous work optimization methods are used without any previous information. Therefore optimization methods (SVC and SGD) are used with machine learning to get better results.

3.2 Problem Statement

With the rapid growth of online information, effective retrieval of some particular information is difficult. Hence text classification plays an important role in effective handling and organizing such huge text collections or data. The proposed work will use supervised and unsupervised learning approaches for optimization of classification techniques to get more optimized results.

3.3 Objectives

I. To reduce the error of text classification by optimization methods in proposed algorithm (e.g. using linear Support Vector Classifier (SVC)).
II. To compare the results of classification approaches (KNN, K-means, Naïve Bayes, SVM, SGD).
III. To validate the proposed approach on 20 newsgroup dataset.

4.1 Text Classifier Methods

Various classifier methods are applied to sample data set based on proposed methodology.

4.1.1 K-means Classifier

It aims to partition n observations into k clusters. Here each partition belongs to the cluster with the nearest mean, which serves as a prototype of the cluster. Given a set of observations $(x_1, x_2 \ldots, x_n)$, where each observation is a *d*-dimensional real vector. The aim is to partition the *n* observations into *k* ($\leq n$) sets $S = \{S_1, S_2 \ldots, S_n\}$ so as to minimize the within-cluster sum of squares (WCSS). In other words, its objective is to find:

$$\arg\min \sum_{i=1}^{k} \sum_{x \in Si} \|x - \mu i\|$$

where μi is the mean of points in S

A set of n objects are classified into k clusters where k is to be inputted by the user. All the data must be available in advance for the classification [15].

4.1.2 Stochastic Gradient Descent(SGD)

SGD is a simple and efficient approach to discriminative learning of linear classifiers. It comes under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. It has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing.

4.1.3 K-Nearest Neighbors Method (KNN)

Neighbors-based methods are known as non-generalizing machine learning methods. The aim is to predefine a number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning), or these can vary based on the local density of points. KNN classifier finds k nearest neighbors on the basis of Euclidean distance.

$$\sqrt{((x_2-x_1)^2-(y_2-y_1)^2)}$$

The value of k is very crucial because the right value of k will help in better classification. [16].

4.1.4 Support Vector Machine (SVM)

A SVM is a discriminative classifier defined by a separating hyper plane. For a linear independent set of 2D-points which fit in to one of two classes, find a separate straight line.

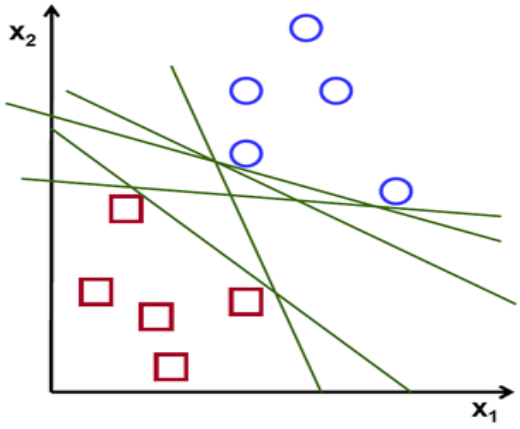

Fig 4.1 Hyper-plane separating two classes [17]

Figure 4.1 shows that there exist multiple lines that given a solution to a problem. A line is said to be bad if it is passed too close through the points due to the noise sensitivity. Therefore, it is very necessary to find the line that should passed through points as far as possible.Then, the methodology of the support vector machine algorithm is based on finding the hyper plane that offer the smallest distance to the training examples. Again, the smallest distance receives the significant name of boundary within SVM's theory. Therefore, the optimal solution for separation of hyper plane increases the boundary values of the training data.

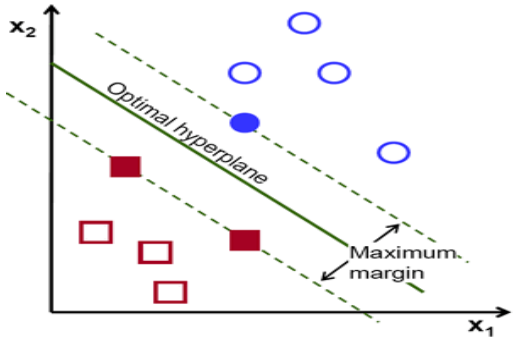

Fig 4.2 Optimal Hyper plane [17]

The given notation is used to describe a hyper plane:

$$f(x) = \beta_0 + \beta^T x,$$

Where β = Weight vector & β_0 = Bias

The numerator is equal to one and the distance to the support vectors, for the canonical hyper plane, is

$$distance_{support\ vector} = \frac{|\beta_0 + \beta^T x|}{||\beta||} = \frac{1}{||\beta||}$$

The boundary values introduced in figure 4.2 is represented as M here, and is twice the distance to the nearby examples:

$$M = \frac{2}{||\beta||}$$

Finally, the problem of maximizing M is equivalent to the problem of minimizing a function $L(\beta)$ subject to some constraints. Formally,

$$distance_{support\ vector} = \frac{|\beta_0 + \beta^T x|}{||\beta||} = \frac{1}{||\beta||}$$

4.1.5 Naïve Bayes Classifier

Naive Bayes methods are based on Bayes' theorem with the "naive" assumption of independence between every pair of features[19]. Given a class variable y and a dependent feature vector x_1 through x_n, Bayes' theorem states the following relationship:

$$P(y|x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n|y)}{P(x_1, \ldots, x_n)}$$

Using the naive independence assumption that for all i, this relationship is simplified to

$$P(y|x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \ldots, x_n)}$$

Since $P(y|x_1, \ldots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y|x_1, \ldots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

$$\hat{y} = \arg\max P(y)\prod_{i=1}^{n} P(x_i|y)$$

and we can use Maximum A Posteriori (MAP) estimation to estimate $P(y)$ and $P(x_i|y)$; the former is then the relative frequency of class y in the training set.

The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i|y)$.

4.1.6 Multinomial Naive Bayes(NB)

Multinomial NB implements the Naive Bayes algorithm for multinomially distributed data. The distribution is parameterized by vectors $\theta_y = (\theta_{y1}, \theta_{y2}, \ldots . \theta_{yn})$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θ_{yi} is the probability $P\left(\frac{x_i}{y}\right)$ of feature i appearing in a sample belonging to class Y.
The parameter θ_y is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\theta_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

The smoothing priors $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing.

4.1.7 Bernoulli Naive Bayes(NB)

This methods implements the Naive Bayes training and classification algorithms for data which is distributed according to multivariate Bernoulli distributions. This means each one is assumed to be a binary-valued variable(true, false). Therefore, this class requires samples to be represented as binary-valued feature vectors. If it handed any other kind of data, a

Bernoulli NB instance may binarize its input. Word occurrence vectors (rather than word count vectors) may be used to train and use this classifier In the case of text classification.

4.2 System Design

4.2.1 Python

Python is an easy to learn, powerful object oriented programming language. It has efficient high-level data structures. It is an ideal language for scripting and rapid application development in many areas on most platforms due to dynamic typing. The Python interpreter is easily extended with new functions and data types implemented in C or C++. It is also suitable as an extension language for customizable applications.

Natural Language Processing with Python provides a practical introduction to programming for language processing. One can easily write Python programs, work with corpora, categorize text using natural language toolkit (NLTK).

4.2.2 Natural Language Toolkit (NLTK)

NLTK is used for building Python programs. It provides easy-to-use interfaces to over 50 corpora and lexical resources. It also has text processing libraries for many function like classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

4.2.3 Data Set

A 20 newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups is used for experiments. It was originally collected by Ken Lang[19].

4.3 Proposed Solution

The proposed solution for text classification using natural language processing involves following steps:

Step 1: Tokenization & Stemming
Step 2: Representation of text using Vector Model
Step 3: Feature Selection
Step 4: Application of Method

Different methods are applied for 3 steps:

a) Feature extraction

b) Training

c) Testing

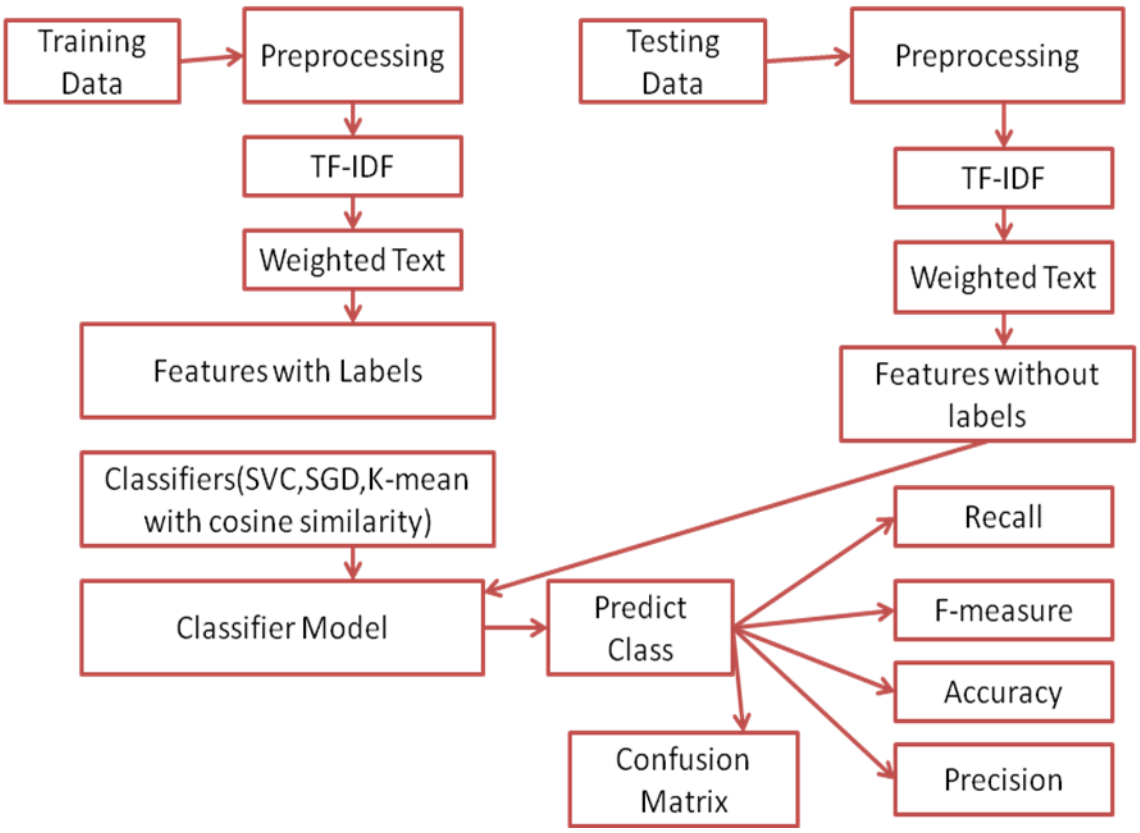

Fig 4.3 Proposed Methodology for text classification

Proposed Algorithm (Algorithm for Text Classification)

Input: unstructured text without label

Output: Labeled text

For I = 0 to I = <length (DOC. Training)

Begin:

Tokenization (DOC [I])

Stop word removal (DOC [I])

TF-IDF (DOC [I])

End

Document with weighted vector

For I= 0 to I<length (DOC.features)

Begin

Put in Linear SVC Classifier

End.

Linear SVC Model

For I= 0 to I <length (Doc .test)

Begin:

Tokenization (DOC [I])

Stop word removal (DOC [I])

TF-IDF (DOC [I])

End

Document test features

Put in Linear SVC model

Check the results in labeled text Precision, Recall, Accuracy, F-measure.

In above algorithm unstructured text is firstly preprocessed and TF-IDF is used to give weight to the text. After preprocessing features are extracted from the text. As features are extracted from the text now the text is converted into a trainable classifier model using SVC classifier. After training, using SVC a model is generated and testing will be done on it. In testing module also, tokenization and features are extracted as before will be extracted and is tested on trained SVC model that whether it predicts class as trained or not.

Step 5: Application of Probabilistic model

A probabilistic model is used to estimate the probability of an event occurring again on the basis of past (historical). Then different models are used instead of SVC for the text classification.

4.4 Performance Measures

In order to compare the performance of various classifiers applied, many performance measures are used:

Precision: Precision of a retrieval algorithm is defined as the fraction of retrieved documents that are relevant to the find.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Recall:

Recall of a retrieval algorithm is defined as the fraction of the documents that are relevant to the query that are successfully retrieved.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

F-measure:

F-measure is a measure of the accuracy of any test. It considers both the recall *r* and the precision *p* of the test to compute the score.

The F_1 score can be interpreted as a harmonic mean of the precision and recall. The best value of F_1 score is at 1 and its worst score is at 0.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Accuracy:

The accuracy of a measurement system is defined as the degree of closeness of measured value of a quantity to the actual (true) value of the quantity.

The results of algorithm implementation and performance measures are summarized in chapter 5.

CHAPTER 5

RESULTS

In this thesis the algorithm Spam Detection is implemented in Python language. Seven different algorithms are used, and their performance is compared. The algorithms used are:

- Linear SVC
- KNN (K Nearest Neighbors)
- SGD Classifier
- Naive Bayes (Multinomial NB)
- Naive Bayes (Bernoulli NB)
- SVM (Support Vector Machine)

Ham message is one which is non a spam message. A Python script was written to process these messages and create a feature vector out of each message. The script divided the generated feature vectors into training set and test set. A ham-spam ratio is maintained in each set.

Actually, the script randomly creates 90 different pairs of training set and test set as follows:

- 9 different "training fractions" were used : 0.1, 0.2... 0.9 i.e. 10%, 20%,….90% of total sample size.
- For each training fraction, randomly 10 different pairs of training set and test set were created.

In the following tables, various parameters for various machine learning algorithms have been shown. The training fraction used is 0.5. The results show that best two algorithms are K-means and Naïve Bayes(Multinomial NB). The values of F-measure, Accuracy, Precision, Recall of different classifiers are given in tables 1-11 for different machine learning algorithms.

TABLE 5.1 COMPARISON OF F-MEASURE FOR VARIOUS CLASSIFIER METHODS

Classifier	F-measure
KNN	85.61
Linear SVC	82.84
SGD Classifier	82.64
K-means	72.23
Multinomial NB	85.77
Bernoulli NB	87.14
SVM	86.78

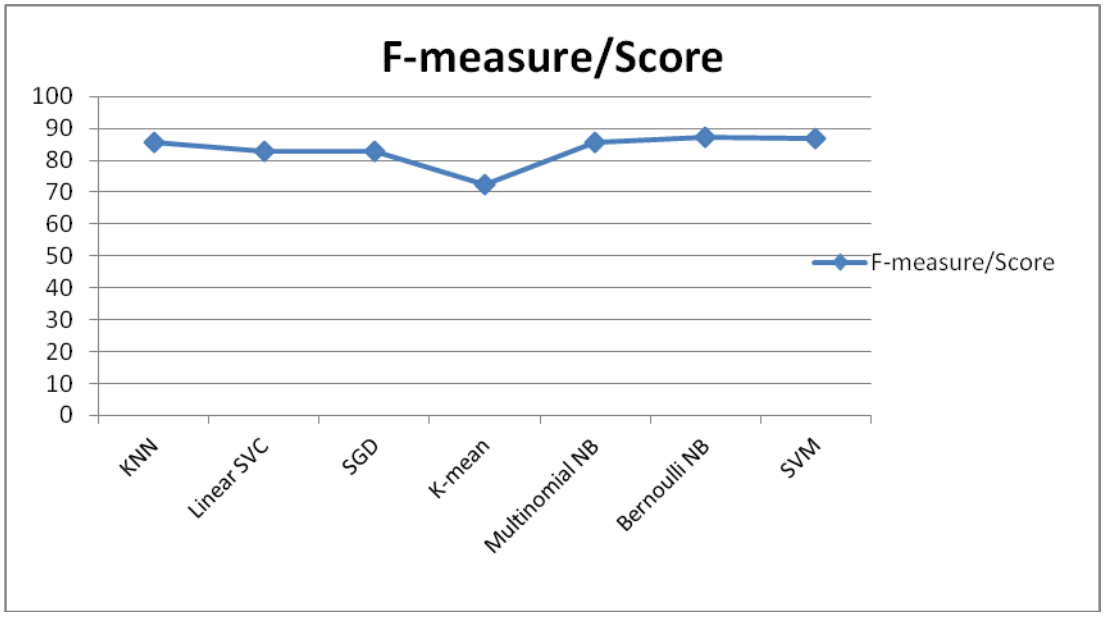

Fig 5.1 A graph showing the comparison of F-Measure for various Classifier Methods

TABLE 5.2 COMPARISON OF ACCURACY FOR VARIOUS CLASSIFIER METHODS

Classifier	Accuracy
KNN	78.27
Linear SVC	86.24
SGD Classifier	86.27
K-mean	87.6
Multinomial NB	84.23
Bernoulli NB	83.4
SVM	84.23

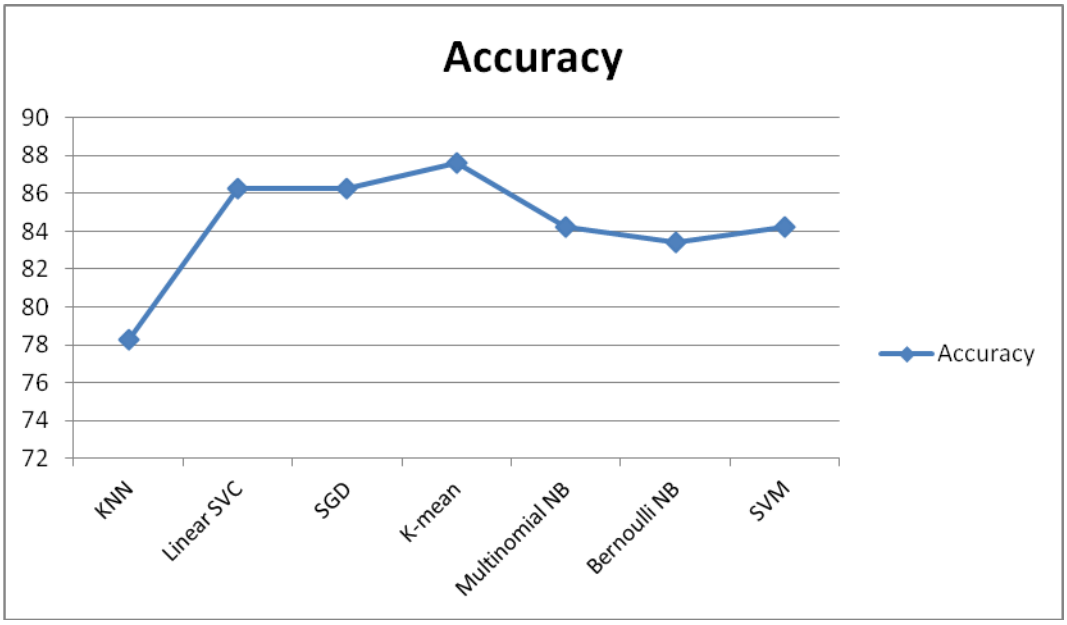

Fig 5.2 A graph showing the comparison of accuracy for various Classifier Methods

TABLE 5.3 COMPARISON OF PRECISION FOR VARIOUS CLASSIFIER METHODS

Classifier	Precision
KNN	89.23
Linear SVC	91.36
SGD Classifier	91.24
K-mean	83.6
Multinomial NB	94.23
Bernoulli NB	88.17
SVM	89.92

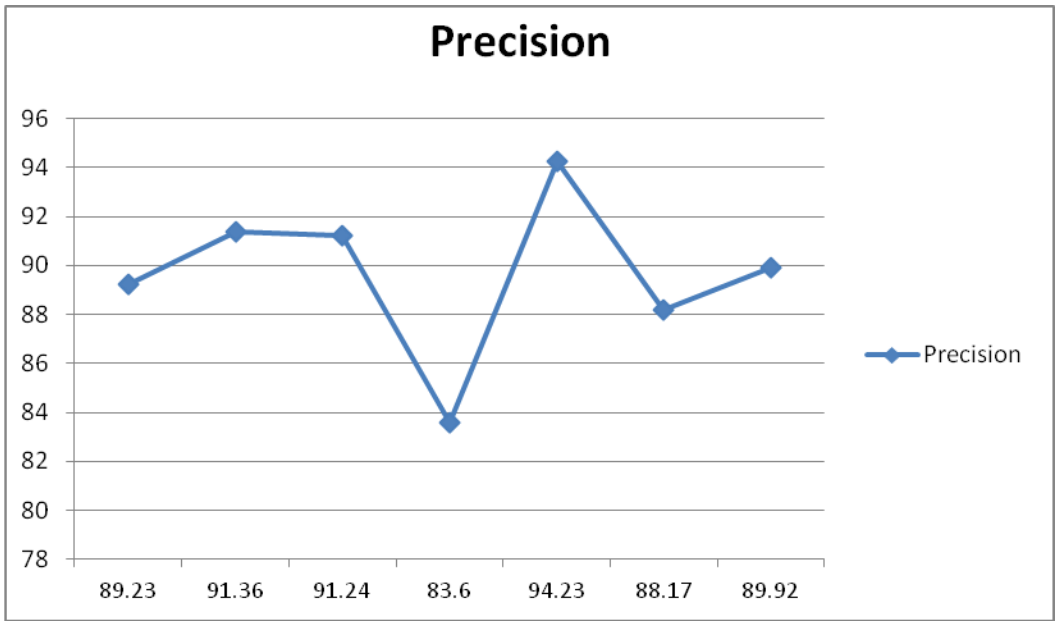

Fig 5.3 A graph showing the comparison of Precision for various Classifier Methods

TABLE 5.4 COMPARISON OF RECALL FOR ALL TEXT CLASSIFIER METHODS

Classifier	Recall
KNN	89.29
Linear SVC	95.64
SGD Classifier	95.64
K-mean	96.01
Multinomial NB	92.34
Bernoulli NB	93.7
SVM	84.23

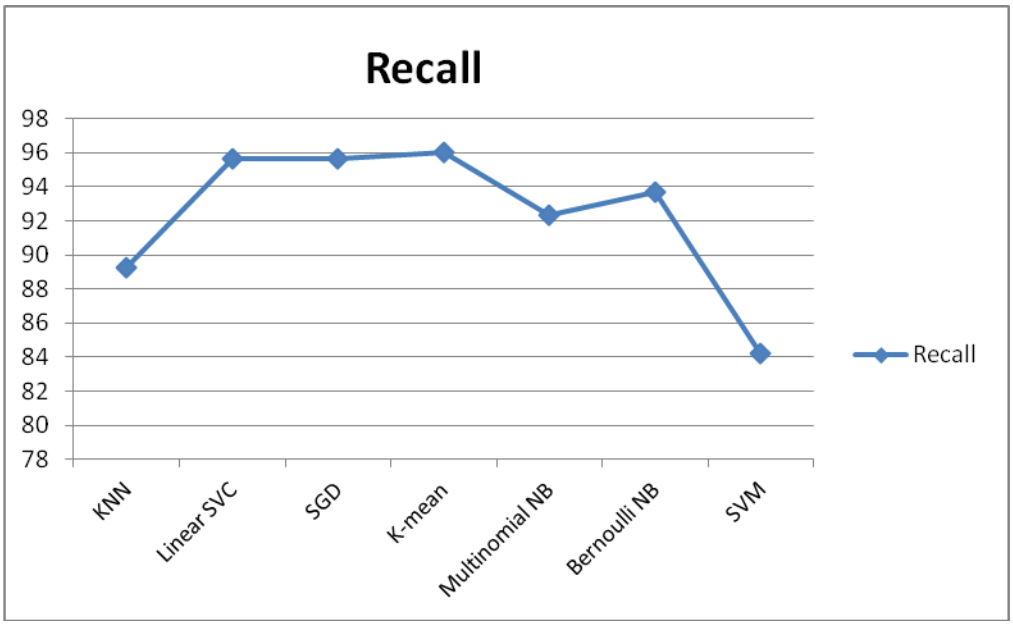

Fig 5.4 A graph showing comparison of Recall for text classifier

TABLE 5.5 RESULTS AFTER USING K-MEANS

Classifier	K-means
Train time	0.02
Test time	0.571
F-measure	72.23
Accuracy	87.6
Precision	83.6
Recall	96.01

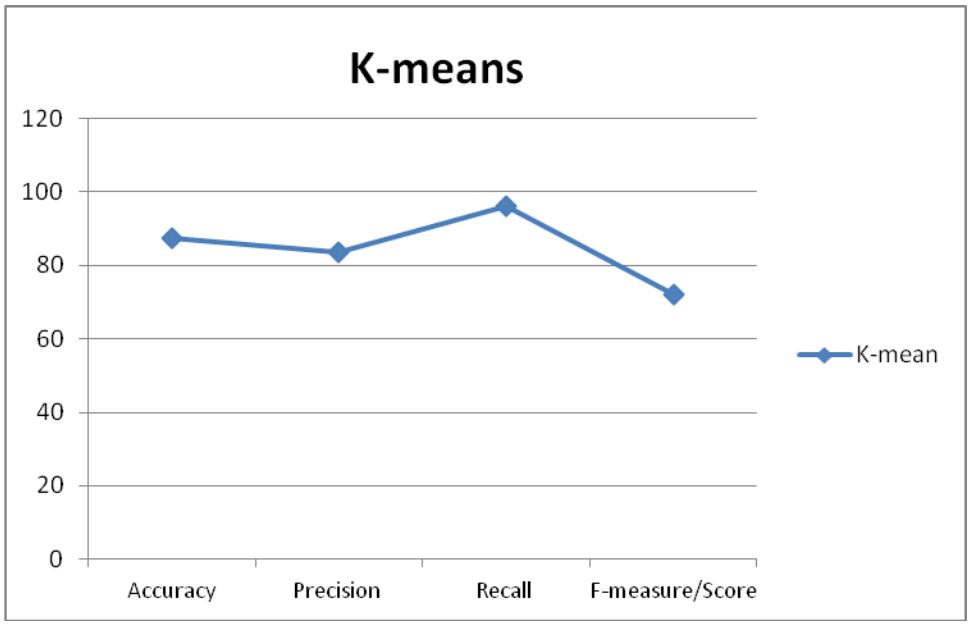

Fig 5.5 A graph showing the results after using K-means

TABLE 5.6 RESULTS AFTER USING MULTINOMIAL NAIVE BAYES

Classifier	Multinomial Naïve Bayes
F-measure	85.77
Accuracy	84.23
Precision	94.23
Recall	92.34

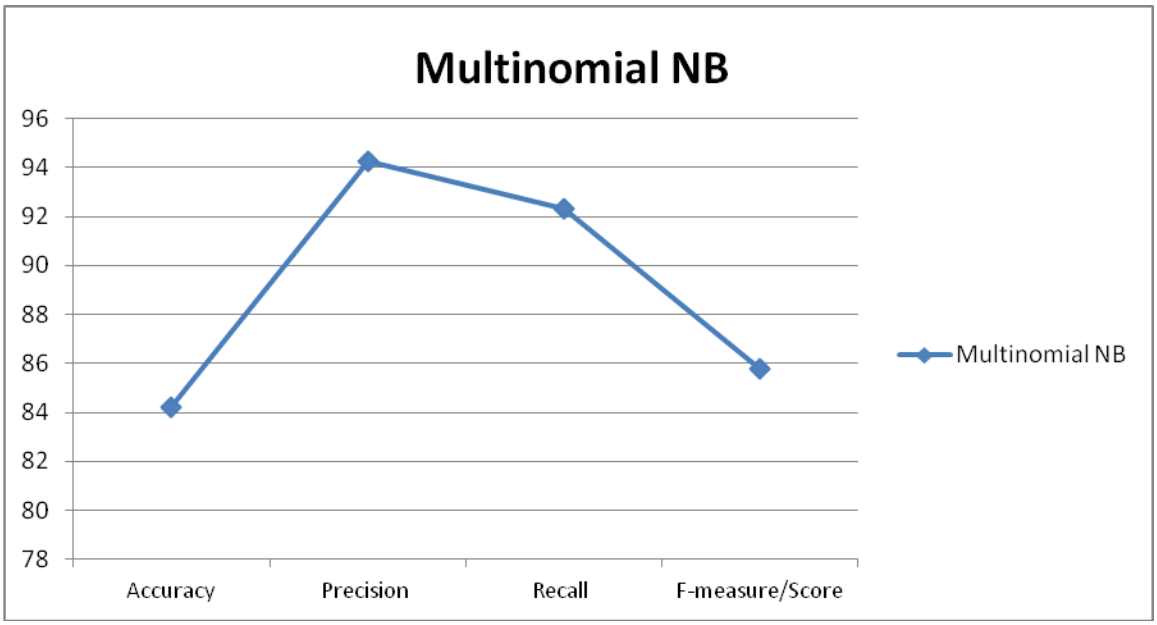

Fig 5.6 A graph showing results after using Multinomial Naïve Bayes Classifier

TABLE 5.7 RESULTS AFTER USING BERNOULLI NAIVE BAYES

Classifier	Bernoulli Naïve Bayes
F-measure	87.14
Accuracy	83.4
Precision	88.17
Recall	93.7

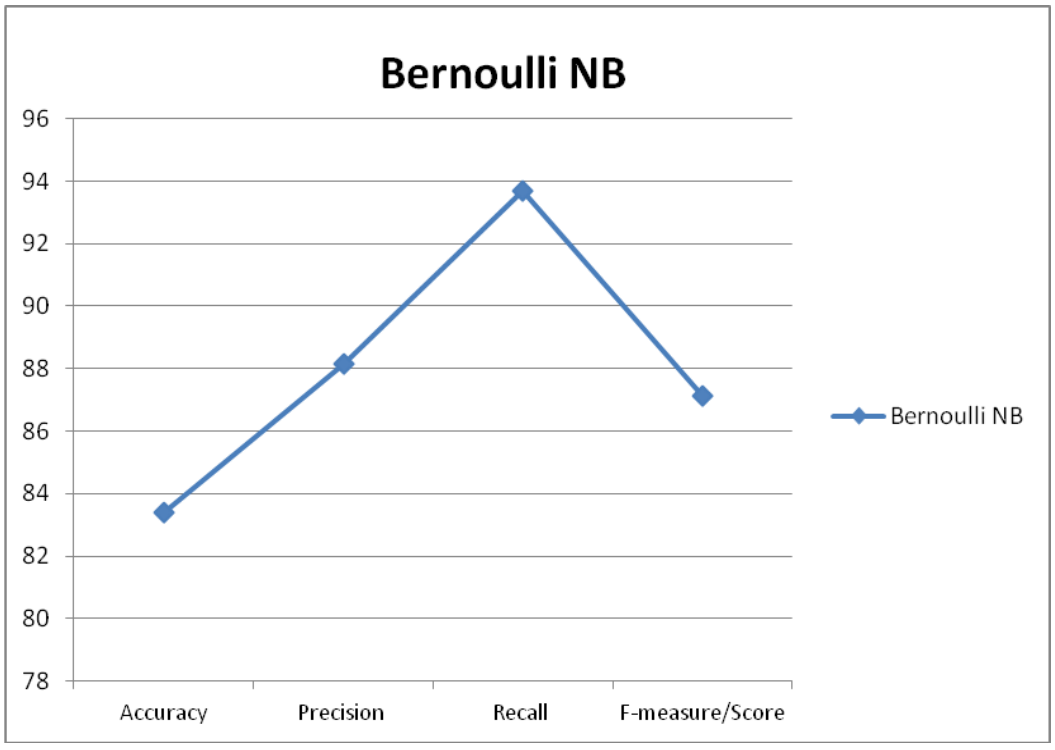

Fig 5.7 A graph showing results after using Bernoulli NB Classifier

TABLE 5.8 RESULTS AFTER USING SGD CLASSIFIER

Classifier	SGD Classifier
F-measure	82.64
Accuracy	86.27
Precision	91.24
Recall	95.64

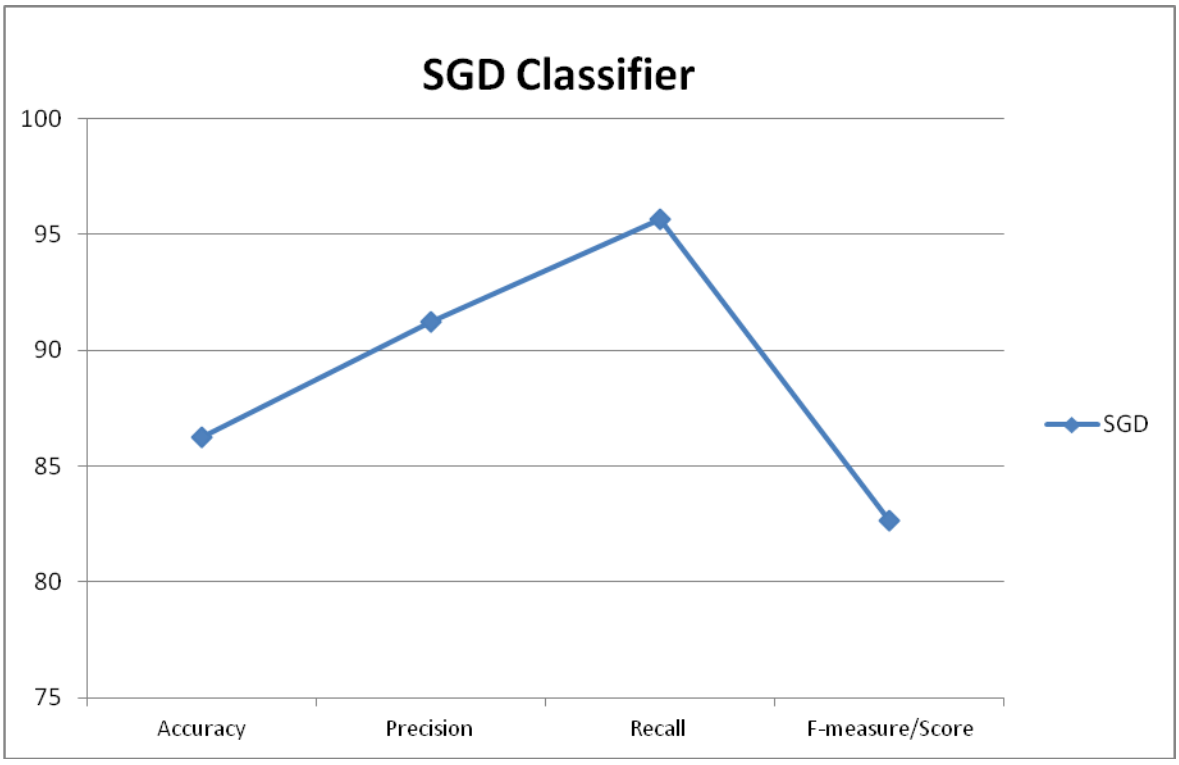

Fig 5.8 A graph showing results after using SGD Classifier

TABLE 5.9 RESULTS AFTER USING SVM

Classifier	SVM
F-measure	86.78
Accuracy	84.23
Precision	89.92
Recall	84.23

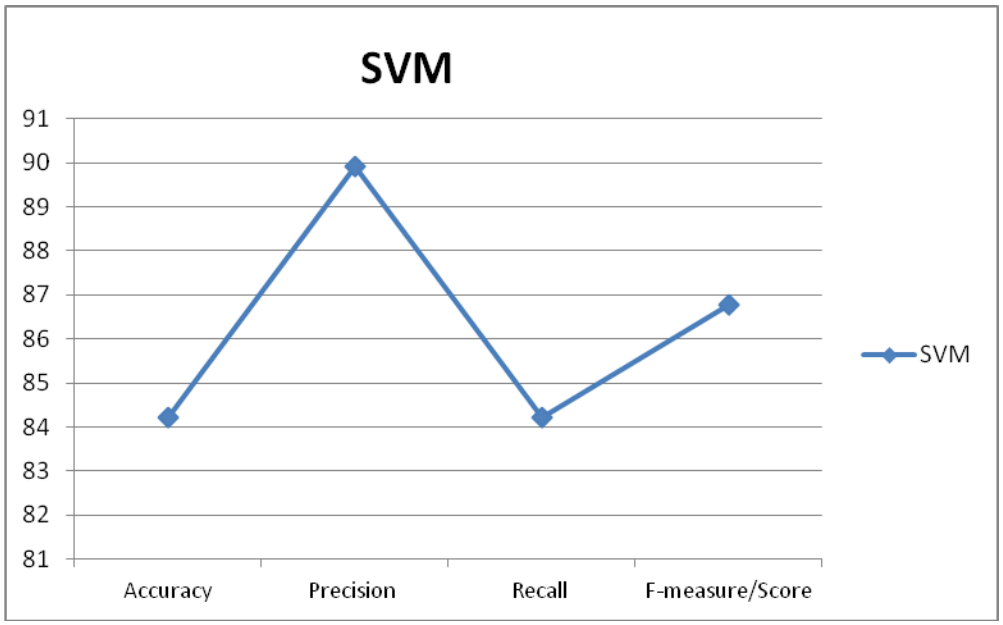

Fig 5.9 A graph showing results after using SVM

TABLE 5.10 RESULTS AFTER USING LINEAR SVC

Classifier	Linear SVC
F-measure	82.84
Accuracy	86.24
Precision	91.36
Recall	95.64

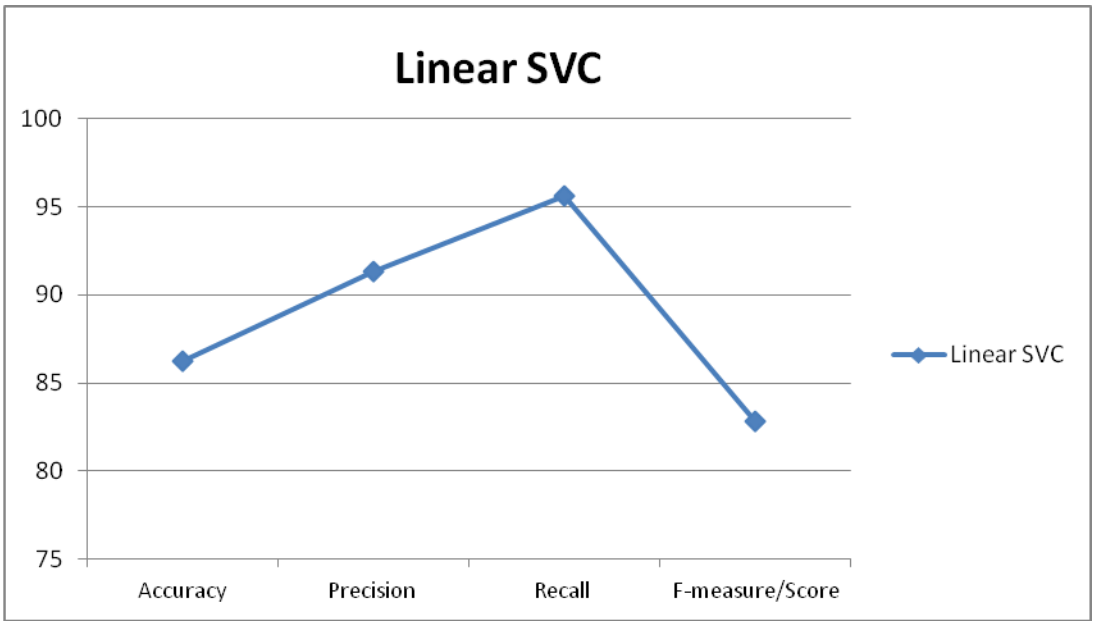

Fig 5.10 A graph showing results after using Linear SVC Classifier

TABLE 5.11 ANALYSIS OF RESULTS OF ALL CLASSIFIERS

Classifier	KNN	Linear SVC	SGD	K-Means	Multinomial NB	Bernoulli NB	SVM
F-measure	85.61	82.84	82.64	72.23	85.77	87.14	86.78
Accuracy	78.27	86.24	86.27	87.6	84.23	83.4	84.23
Precision	89.23	91.36	91.24	83.6	94.23	88.17	89.92
Recall	89.29	95.64	95.64	96.01	92.34	93.7	84.23

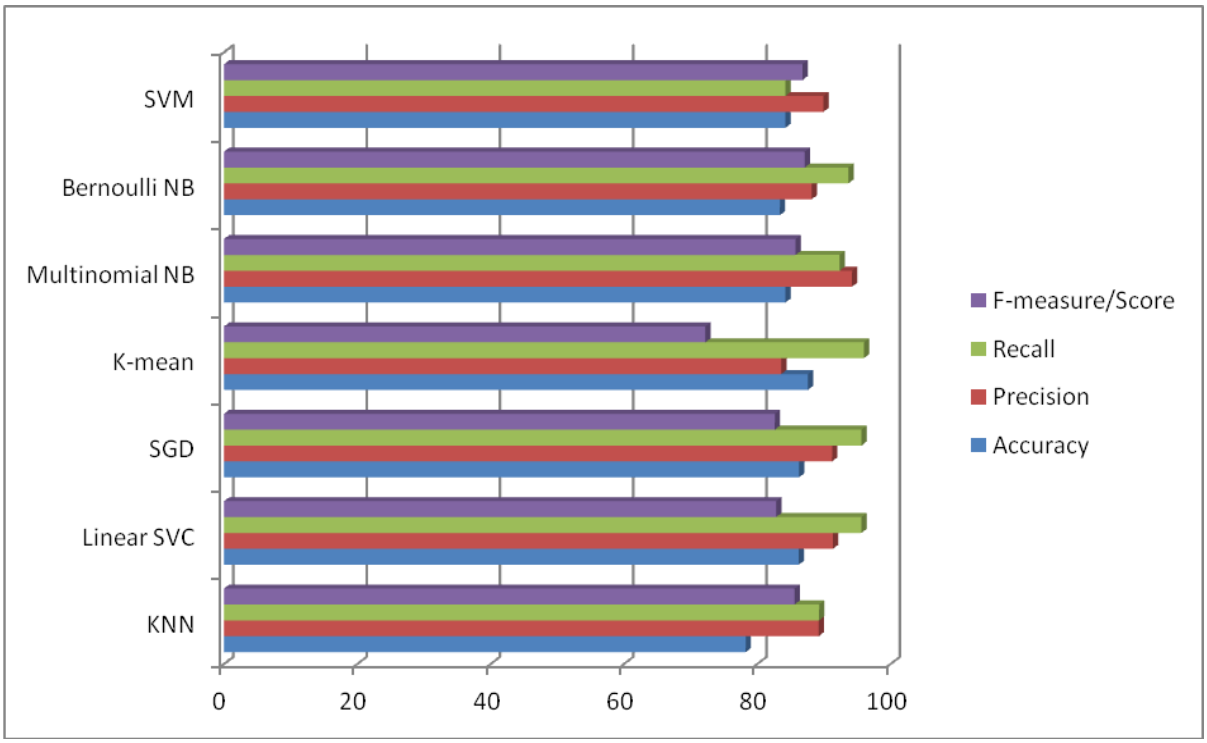

Fig 5.11 A graph showing analysis of results of all the classifier methods

The experiment result shows that F measure is best for Bernoulli NB followed by SVM. Accuracy and recall is best in K means. Precision is best with Multinomial NB. The highest value of precision is given by Multinomial NB followed by Linear SVC. SGD gives good value for recall.

CHAPTER 6

CONCLUSION & FUTURE SCOPE

The experiment results show the best result algorithms SVM and Naïve Bayes (Bernoulli NB). Naïve Bayes performs well as precision if highest with Multinomial NB and it is easier to implement also. It has low running time but its accuracy and recall are less than K means. Hence we conclude that optimization methods perform well and show better results than other classifiers.

In future, kernel function can be used for reducing the processing time.

[1] Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari, Jordan Pascual, "KNN based Machine Learning Approach for Text and Document Mining", *International Journal of Database Theory and Applications*, Vol.7, No.1, 2014, pp. 61-70.

[2] Wen Zhang, Taketoshi Yoshida, Xijin Tang, "A Comparative Study of TF*IDF ,LSI and multi words for text classification", *Expert Systems with Application Journal Elsevier*, Vol. 38, No. 3, 2011, pp. 2758-2765.

[3] Aixin Sun, "Short Classification using very few words", *In Proceedings of the 35th International ACM SIGIR Conference on Research and development in Information Retrieval*, 2012, pp. 1145-1146.

[4 Mengen Chen, Xiaoming Jin, Dou Shen, "Short Text Classification Improved by Learning Multi-Granuality Topics", *In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2010, pp. 1776-1781.

[5 Tanmay Basu, C. A. Murthy, "Effective Text Classification by a Supervised Feature Selection Approach", *Proceedings of the 2012 IEEE 12th International Conference on Data Mining Workshops*, 2012, pp. 918-925.

[6] Youngjoong Ko, "A Study of Term Weighting Schemes Using Class Information for Text Classification", In *Proceedings ACM Conference on SIGIR'12*, 12-16 Aug, 2012, pp. 1029-1030.

[7] Svetlana Kiritchenko, Stan Matwin, "Email Classification with Co-training", *In Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative Research, 2001, CASCON '01*, pp. 8.

[8] Guansong Pang, Shengyi Jiang, " A Generalized Cluster Centroid based classifier for text categorization", *Information Processing and Management Journal, Elsevier*, Vol. 48, No. 2, 2013, pp. 576-586.

[9] Samuel Danso, Eric Atwell, Owen Johnson ,"A Comparative Study of Machine Learning Methods for Verbal Autopsy Text Classification", International Journal of Computer Science Issues, Volume 10, Issue 6, No 2, November 2013 , pp. 1-10.

[10] Hugo Larochelle and Yoshua Bengio, "Classification using Discriminative Restricted Boltzmann Machines", In Proceedings of the 25 th International Conference on Machine Learning, Helsinki, Finland, 2008, pp. 1-8.

[11] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani , "Using micro-documents for feature selection: The case of ordinal text classification", *Expert Systems with Applications*, Vol. 40, 2013, pp. 4687-4696.

[12] Rui Xia , Chengqing Zong, Shoushan Li, "Ensemble of feature sets and classification algorithms for sentiment classification", *Information Sciences*, Vol. 181 ,2011, pp. 1138–1152.

[13] Kamal Nigam, Andrew Kachites Mccallum, "Text Classification from labeled and Unlabeled Documents using EM", *Machine Learning*, 1999, pp. 1-34.

[14] Danesh Irani, Steve Webb, Calton Pu and Kang Li, "Study of TrendStuffing on Twitter through Text Classification", *CEAS 2010 Seventh annual Collaboration, Electronic messaging, AntiAbuse and Spam Conference* July 1314, 2010, Redmond, Washington, US, pp. 1-10.

[15] Sang Min Lee, Dong Seong Kim,Ji Ho Kim,Jong Sou Park, "Spam Detection Using Feature Selection and Parameters Optimization", *IEEE International Conference on Complex, Intelligent and Software Intensive Systems*, Feb. 15-18, 2010, Krakow, Poland, pp. 883-888.

[16] Sarwat Nizamani, Nasrullah Memon, Uffe Kock Wiil, Panagiotis Karampelas, "Modeling Suspicious Email Detection using Enhanced Feature Selection", *International Journal of Modeling and Optimization*, Vol. 2, No. 4, April 2012, pp. 371-377.

[17] Viet Ha-Thuc and Jean-Michel Renders "Large-Scale Hierarchical Text Classification without Labeled Data", In proceedings, *WSDM'11,* February 9–12, 2011, Hong Kong, China, pp. 1-10.

[18] Qian Xu, Evan Wei Xiang and Qiang Yang, "SMS Spam Detection Using Non-Content Features", *IEEE Intelligent Systems*, Vol. 27, No. 6, Dec. 2012 , pp. 44-51.

[19] http://qwone.com/~jason/20Newsgroups/

[20] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Second Edition, The Morgan Kauffman Series in Data Management System.

LIST OF PUBLICATIONS

Suresh Kumar, Shivani Goel, "Enhancing Text Classification by Stochastic Optimization method and Support Vector Machine", International Journal of Computer Science and Information Technology, July 2015 (accepted).