# Chapter 1

# Introduction

Day by day the amount of information is increasing with the development of the technology, human lifestyle and factors of social. 80% of data are in textual form. So, test classification is an important task for organization of data. Text classification is a set of actions or procedures to categorize text in certain organized group. Now a day, many classification algorithms are established like k nearest neighbors, naïve Bayes, svm and association classification etc. Feature is one of the most important terms in machine learning and data mining. Noisy and high amount of feature can cause problems in classification. There is a few work for feature selection for Bangla test classification.

### 1.1 Background

Text classification is the process of categorizing text files into organized groups. Text classification offers a good framework for getting familiar with textual data processing without lacking interest. In recent year, proposed text classification methods use many models to classify different languages text documents. Most of the classifiers identify a document in any class with all unique feature's term value i.e. TF-IDF values of each unique words in documents training in any classifier model. But it has faced dimensionality problem. So dimension should be reduce using feature selection. There are four main types of feature selection: filter method, wrapper method, embedded method and hybrid method. In filter method, there is no need to concentrate on classification model. It is only focused on features importance factors. On the other hand, wrapper method wraps classification method. Embedded method where feature selection is considering a part of the classification model. Finally, hybrid method is combination of filter method and wrapper method. Our proposed model combines those two method for get batter outcomes. Here we introduce genetic algorithm in wrapper method. Genetic algorithm is an evolutionary algorithm based on Darwin's natural selection theory.

### 1.2 Motivations

Textual data's features space can contain huge number of features, and performing text classification with such a high dimensional feature space influences the quality of the separation of different class of test. Noise also is a concerning issue. When the data features number is huge than quantity of documents. Then there is a probability of training model incorrectly. On the another way, huge quantity of features consumes more memory and take more time to run the training phase. It calls computational overhead to the memory distribution. For so, features selection is a imperative part of machine learning. We know that, unique terms are treated as

feature of documents in documents classification. If a single document contains at least ten unique terms, then 2000 documents have at least 20k features. But our dataset contains more document and more features. Then the features quantity becomes as more as it is hard for a computational model to cope with the huge amount of features. It can call curse of dimensionality problem. Most of the Bengali text classifier has modeled without considering dimensionality problem. Day by day Bengali documents vault is continuously rising. Besides there are a few works for the Bengali text documents. So it feels that Bengali text classifier should to be developed for proper utilization of data. This problem become very complex regarding features distribution problem region. Higher quantity of features means higher dimensionality. Model cannot perform well in higher dimension of features. As the search space of features is increased, it consumes more memory and time. Features selection is a complex problem. Firstly, selection of most effective features is a difficult task. It is because of various features ranking measurement metrics. Different method can rank features differently. Which one is accurate and correct form? It is the one of the deceptive questions. The answer is very from person to person. Second concern is features reduction effect on model evaluation result. Sometimes features reduction can reduce the evaluation outputs result like precision, recall, accuracy etc. Finally, time and memory is another concern. Feature selection take more memory and time with model computation time. SO, it is said that feature selection is complex problem. But features selection should be done before model training. Most of the time it increases the better outcomes probability.

### 1.3 Objectives

This objective of this work is to reduces dimensional problems of Bengali text classifier. For this purpose, here will be introduced enhanced GA for features optimization and select a standard number of features rather all unique features. Finally, this proposed method supposed to be the one of the best solution for Bengali text classifier. Our proposed model is supposed to reduces the problem described previous section. Our model is designed based on hybrid method. In our method we use two simple filter method: Term frequency Inverse Document Frequency and Log-TFIDF-Cosine (LTC) methods. Firstly, features subset is generated using filter method and finally genetic algorithm does further feature reduction. For wrapper part we use genetic algorithm with enhanced procedures. Those procedures are supposed to reduce most of the problems. In genetic algorithm here we make few change in genetic function: crossover, mutation and selection. For selection we use rws method with three variants. In crossover we use Mendelian first low where child always takes parents fitness. As a result, there is no way that feature reduction reduces evaluation metrics outcomes. Summary of the objectives:

- Reduces dimensional problems of Bengali text classifier.
- Build more efficient hybrid model to feature selection.
- Introduced enhanced GA for features optimization.
- Select optimum number of features for classifier.
- Make a classifier model that is more accurate than before.

# Chapter 2

# Related Work

Many researchers had contributed in the feature selection and classification. Most of the work I have seen is filter method or wrapper based. Few a work is based on hybrid method. Now a day's evolutionary algorithms are used in feature selection. They give better result than local feature selection algorithm. Papers I have seen in Bangla language, mostly use only filter approaches to select feature for text classification. Few of the papers review is given below.

Jasmina NOVAKOVIĆ et al. [1] presented a comparison between several feature ranking methods implemented on two different dataset. They analyzed with six ranking methods that can be classify into two main categories: statistical based and entropy-based. Four supervised learning algorithms are used for building models: IB1, Naive Bayes, C4.5 decision tree and the RBF network. They get 75-80% classification accuracy in classification stages after features selection.

Laith Mohammad Abualig et al. [2] proposed the genetic algorithm (GA) to solve the unsupervised feature selection problem, namely, (FSGATC). They experiment on four different text dataset. Their main target is to reduce sparse and uninformative features. There was used vector space model to represent documents in dataset. They combine TF-IDF method and regular genetic algorithm for features reduction. They compare result with K-Mean Clustering.

A.K. Uysal et al. [3] proposed IGFSS method with target to improve the classification performance using global feature selection methods by extracting a features set viewing all classes equally. One local feature selection method was used to identify features on discriminative power on classes. The labels are used to generate the final feature sets. They used SVM and Naïve Bayes They evaluate the model in term of Micro-F1 and Macro-F1 metrics. Their scores varies from 50-90% on different algorithms.

Bing Xue [4] presented the first study on multi-objective particle swarm optimization (PSO) for feature selection generating a Pareto front of non-dominated solutions (feature subsets). They analyzed two PSO-based multi-objective feature selection algorithms: one introduces the idea of non-dominated sorting into PSO to address feature selection problems and another algorithm applies crowding, mutation, and dominance to PSO for the Pareto front solutions. They have showed that they get around 60% to 85% F1 scores.

A.S. Ghareb et al [5] introduced hybrid feature selection with EGA. They made change in GA's crossover and mutation to reduce feature multidimensionality. They used Arabic news dataset for the experiment analysis. They gives weight on features for being participated in crossover. They divided an individual chromosome into two part. Then calculated the relative weight of each part of two chromosomes. Best part generated a child and worst part generated another child. This is their variation in GA. There they used two classifiers: NB and associative

classifier. They calculated fitness in term of Macro F1 score. Their precision result varies from 60-92%, recall varies from 50-88% and F- measure varies from 50-90%.

Thomas et al. [6] presented a method for filtering span emails. There they have used ten filter method for feature selection. They trained selected features data in Support Vector Machine and Random forest algorithm. They have got about 97.8% F1 score. They said that they are successful to reduce 80% of main length of features set.

Ankita Dhar et al. [7] proposed an automated text classification for Bangla language with recent feature selection method TF-IDF-ICF. They use inverse class frequencies with TF-IDF method. The have shown that they get accuracy around 98.87%. They use multinomial Naïve Bayes classifier for classification.

Yan Xu et al. [8] proposed a text categorization method with mutual Information. They have shown two different MI based features selection laws in text categorization reviews. They differentiated between pointwise mutual information(PMI) and mutual information(MI). They also compared among document frequency(DF), information gain(IG), PMI and MI.

Md Mofijul et al. [9] proposed a Bangla text classification method and named it BARD. There they develop a web application that can properly classify text. They have used ward2vec and TF-IDF method to represent documents and classification. They have used five supervised classifiers to classify. They get result around 70- 96%.

# Chapter 3

# Proposed method

## 3.1 System Overview

At first, we collect data from Bangla News Portal Prothom alo using python *Beautifulsoup* modules. Then our procedural works are done step by step. After creating dataset, we preprocess out text dataset. After that, hybrid features selection will be happened. Finally, we make classifier modes to classify our dataset. Fig 3.1 is the total model of our system.

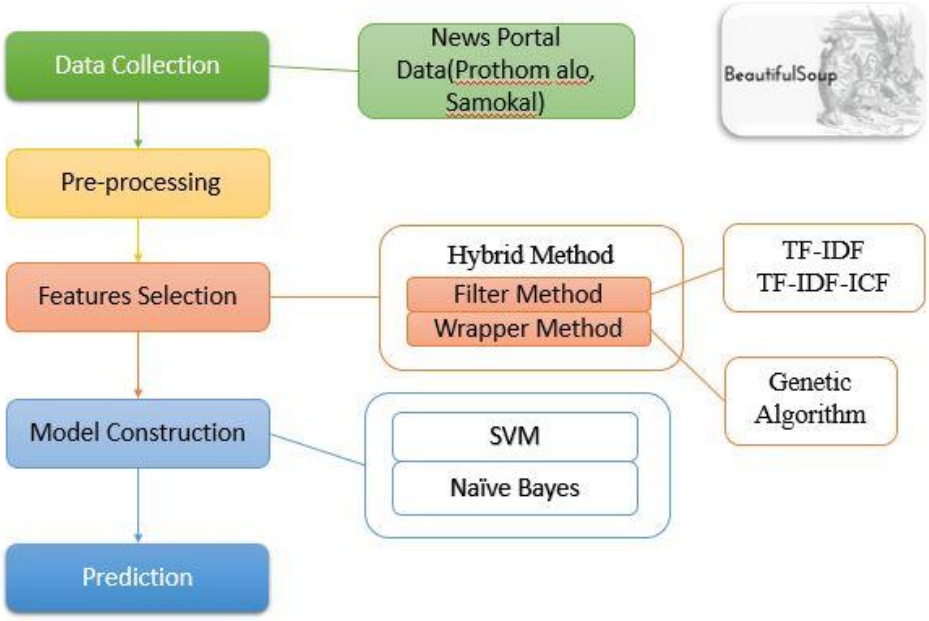


Fig. 3.1: System Diagram.

## 3.2 Data Collection

We collect data from Bangla News Portal site (Prothom Alo) using web scarping. For web scarping we use Python web scarping library BeautifulSoup and request library for https request responses. We are able to collect about 12K Bengali text documents.

## 3.3 Proposed Methodology

The proposed methodology has three steps to select features and texts classification. Those steps are: A. Pre-processing, B. Feature selection and C. Model construction and prediction. Step A describes data preprocessing steps. B describes about our hybrid feature selection method combining filter and wrapper methods. In wrapper method we use Genetic algorithm(enhanced).

After feature selection, in step C, Bengali text documents will be trained in three classification models separately: Support Vector Machine, Naïve Bayes and Decision Tree. Finally, we will analysis our outputs with previous established methodologies.

### 3.3.1 Preprocessing

Preprocessing is the data cleaning stage of our methodology. We know that Bangla sentence contains complex grammatical structure. Words used in a passage more firmly with suffix, prefix or with other word's variants. Also sentences have different grammatical structure. As a result, sentences contain words those are only needed for sentences formation. Those elementary words are called stop word. Sentences also contains different punctuation marks. In the preprocessing step, the punctuations, stop words and suffixes get removed. The whole document should be preprocessed first for getting batter result in training stage. Briefly preprocessing steps are described below.

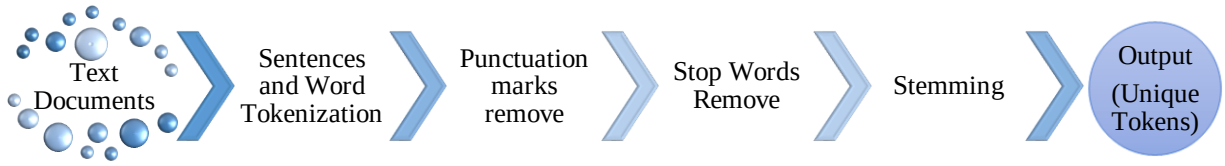


Fig. 3.2: Preprocessing Procedures

*a) Text and Word Tokenization:* Firstly, the sentences and words should be tokenized. Sentences are tokenized with sentences completing punctuations points. Here, sentence tokenized sets are generated. The number of the set elements are the total length of a documents. Words are tokenized by white space of a word's end. Each sentence of documents are represented containing words tokenized. Tokenized words help to remove unnecessary data from documents.

*b) Punctuations Removal:* Punctuations can make noise in the word scoring process of filter stages. Bangla language has more than eight punctuation marks. So, punctuations should be removed.

*c) Stop Words Removal:* Briefly, stop words stands for the common and the mostly used words that don't have any meaning. Stop words are chosen for a certain purpose. In Bengali language, words those are used only for structural purpose are considered as stop words. So, stop words are unnecessary item for scoring words and sentences. Stop words elimination will reduce the analysis complexity and save time. So, stop words should be removed. We collect stop words from [12].

*d) Stemming:* Bangla texts have interesting complex structure. A single word can be used in text with suffix or prefix items. Without suffix items, the meaning of a certain word cannot very a lot. So, word having different suffixes create redundancy in sentences segments. As a result,

suffixes behind a words create difference identity words. Those will score differently. If suffixes are eliminated, this score redundancy will be solved. So, all word should be stemmed before going to next stages. Our Stemmer removes suffix part from word. After stemming, the word is divided in two parts: root and suffix. Root part is main part of word meaning. The method only works with stem (root) part.

**Illustration:** Preprocessing task

**Initial Text:** আজ রাত ৮ টায় স্টেশন থেকে বাস ছেড়ে যাবে।

**Transition Steps:**

| | |
|---|---|
| Sentence Tokenization: | 'আজ রাত ৮ টায় স্টেশন থেকে বাস ছেড়ে যাবে।' |
| Words Tokenization: | 'আজ', 'রাত', '৮', 'টায়', 'স্টেশন', 'থেকে', 'বাস', 'ছেড়ে', 'যাবে', '।' |
| Removing Punctuation: | 'আজ', 'রাত', '৮', 'টায়', 'স্টেশন', 'থেকে', 'বাস', 'ছেড়ে', 'যাবে', |
| Removing Stop Words: | 'আজ', 'রাত', '৮', 'টায়', 'স্টেশন', 'বাস', 'ছেড়ে', 'যাবে' |
| Stemming Words: | 'আজ', 'রাত', '৮', 'টা', 'স্টেশন', 'বাস', 'ছেড়ে', 'যাব', |

Fig. 3.3: Illustration of preprocessing steps.

### 3.3.2 Features Selection

A subset of features S, which can be selected using any filter method, the problem is still the high dimensionality of the features that can be revised with the wrapper approach. The GA to reduce text dimensionality. However, the GA is impractical for high dimensional text because it takes a long time to locate the relevant feature subset. To avoid these problems and achieve better performance in terms of text classification, we propose four filter approaches with GA. There are two stages in FS process: filter stage and GA stages.

### 3.3.2.1 Filter Stages

This stage is performed for three purposes, first to perform an initial reduction of text dimensionality, second, to further minimize the effect of randomization of the initial population generation in EGA and third to speed up the feature subsets generation process with EGA. The following equations present these methods mathematically, as computed for each feature $f_i$ in category $c_i$.

*1.Term Frequency Inverse Document Frequency (TF-IDF):* The tf-idf weight is a weight often used in information manipulation and text mining. This weight is a statistical calculation used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of appearances a word appears in the document but is offset by the frequency of the word in the corpus.

*TF:* **Term Frequency**, which is the measurement of generalized frequently a term in a document. Since every document is varied in length, it is not impossible that a term would appear much more times in long documents than shorter ones.

$$TF(f_i) = \frac{N_{f_i}}{N}$$

*IDF:* Inverse Document Frequency measures how essential a term is. In Term frequencies(TF), all words are seemed equally essential. Inverse document frequencies down the weight of frequent terms while increase rare terms weight.

$$IDF(f_i) = log\ (\frac{Total\ Number\ of\ Documents}{Number\ of\ documents\ with\ f\ term})$$

TFIDF is the multiplication term frequency and inverse document frequency.

$$TFIDF\ = TF(\ f_i\ , c_i)\ \times IDF(f_i)$$

*2. TF-IDF-ICF* : TF-IDF-ICF is the new extended version of the TFIDF feature selection method. TF-IDF-ICF is the multiplication of TFIDF and inverse class frequency(ICF).

$$ICF(\ f_i) = log\ \frac{C}{CF(f_i)}$$

Here C is the number of class domain and $CF(f_i)$ is frequency of a feature in a distinct class.The TF-IDF-ICF formula is:

$$TF - IDF - ICF\ = TF(\ f_i\ , c_i)\ \times IDF(f_i) \times ICF(\ f_i)$$

In the stage, N number of features are filtered to features subset S. S contains features in several number according to their importance factor.

### 3.3.2.2 Genetic Algorithm Stage

This is the main section of our work. In this stage, the input is feature subset S. The output is our resultant reduced important features set. In this stage, there are five major operations. Here we use genetic algorithm's procedures to select the most important features subset. Now following section will be described about the five procedures briefly.

*1. Initialize Population:* Here, a number of population is generated from feature subset randomly at beginning. Every features is called the gene of a chromosome or individuals. An individual is collection of genes. Basically chromosome is a set of binary array. Where 1 means the selected feature and 0 means feature is not selected.

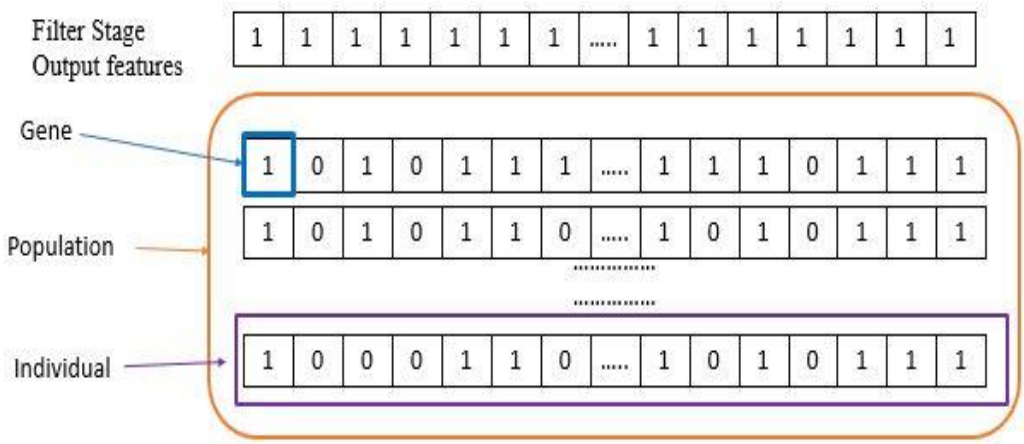


Fig. 3.4: Gene, population and Individuals.

*2. Fitness Function:* The feature subsets performances are calculated using fitness function. It plays major role for individual selection proceed to next generation. Higher scored subsets have higher probability of being selected for next generation generating. In our work we used Naïve Bayes classifier in wrapper method. Here we use macro average F1-measure for fitness evaluation. Following function is used for fitness calculation:

$$Fitness(S_i) = Z * C(S_i) + (1 - Z)\frac{1}{Size(S_i)}$$

Here $S_i$ is selected subset of features. $C(S_i)$ is macro average f1- score. Z is a random variable ranging between 0 and 1.

It holds the relation between feature subset performance vs subset size. It gives importance to feature performance.

*3. Selection:* After fitness calculation has finished, selection takes places. Selection is done according subsets relative finesses. Most probable subsets are selected for generation of next generation. Here every time two subset of features are selected for production of next generation. For selection we will use two methods for two condition. If the feature relative fitness's mean deviation is point to zero then we use Rank Selection method, otherwise we will use Stochastic Universal Selection (same as RWS).

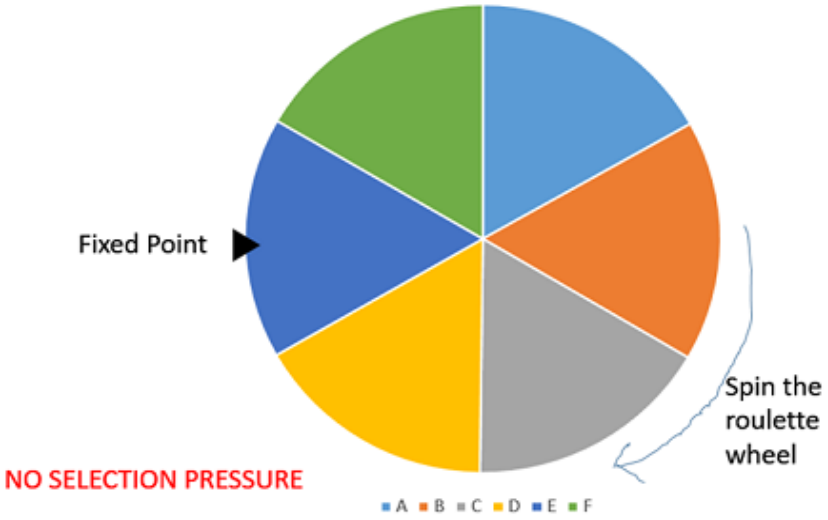


Fig. 3.5: Rank selection

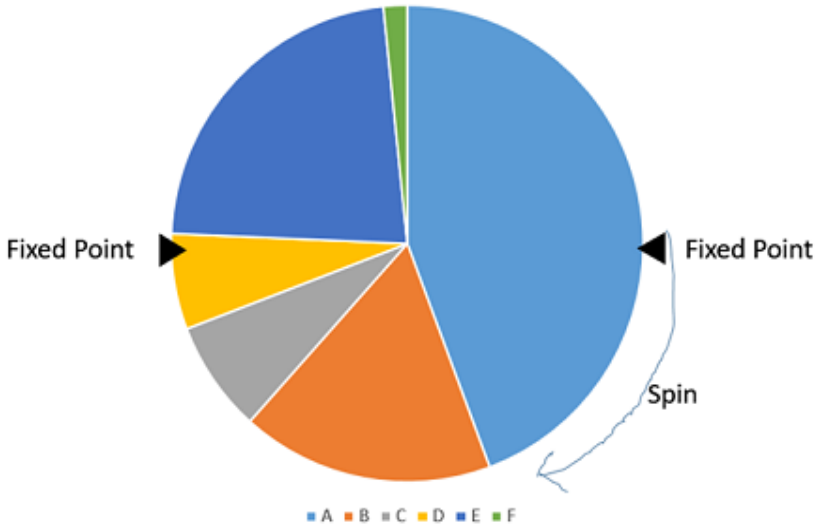


Fig. 3.6: Roulette wheel selection

In rank selection all subset has equal selection probability. Stochastic Universal selection method elements has their own selection probability. Here all probability is unequal. This selection method also known as Roulette wheel selection. Selection probability of a subset is:

$$P(S_i) = \frac{Fitness(S_i)}{\sum_{i=1}^{n} Fitness(S_i)}$$

*4.* Crossover: In the step two parents generate child. Here we modify usual crossover methodology. We divide each two parent in two part. Then calculate cumulative fitness of each part is calculated. Then we apply Mendel's first law to every parent sub chromosome. Then There will generate four children from two different parents. Here we use two crossover point.

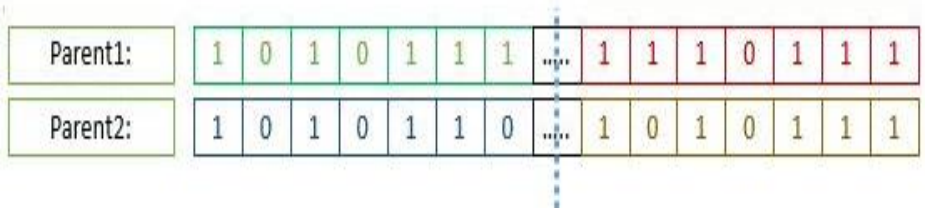


Fig. 3.7: Example parents

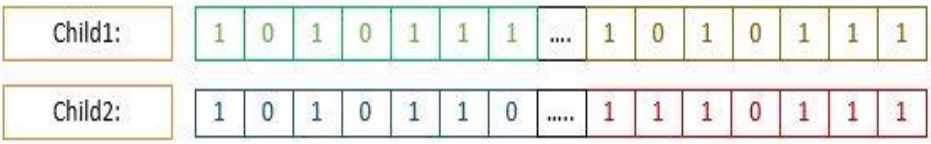


Fig. 3.8: Example Childs from parent

*5. Mutation:* Children gene should have verity than parents. This method is applied to a single child at a time. A random number of genes is selected of a child then flip the gene's values. In our method, we mutate best chromosomes children. In proposed method, here specific number of genes(features) are selected if child cumulative fitness is less than parent. Replace specific number less important feature with highest important features (not in subset before). Mutation method make child different from parent. We will use two-point mutation. Here we randomly flip less fitted genes status.

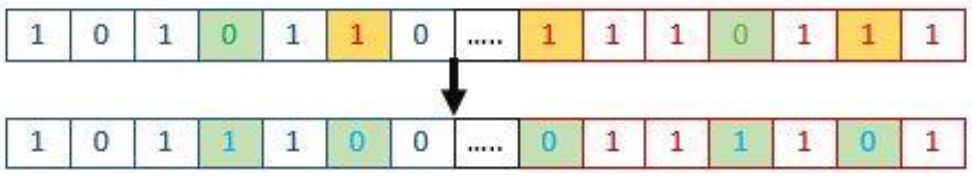


Fig. 3.9: Mutation

Those described process will be continuing iteratively until termination criteria is met. Here in our method the process will be stopped when there finds same best fitness score through few sequential generations. Crossover, selection and mutation is called genetic function in combine form.

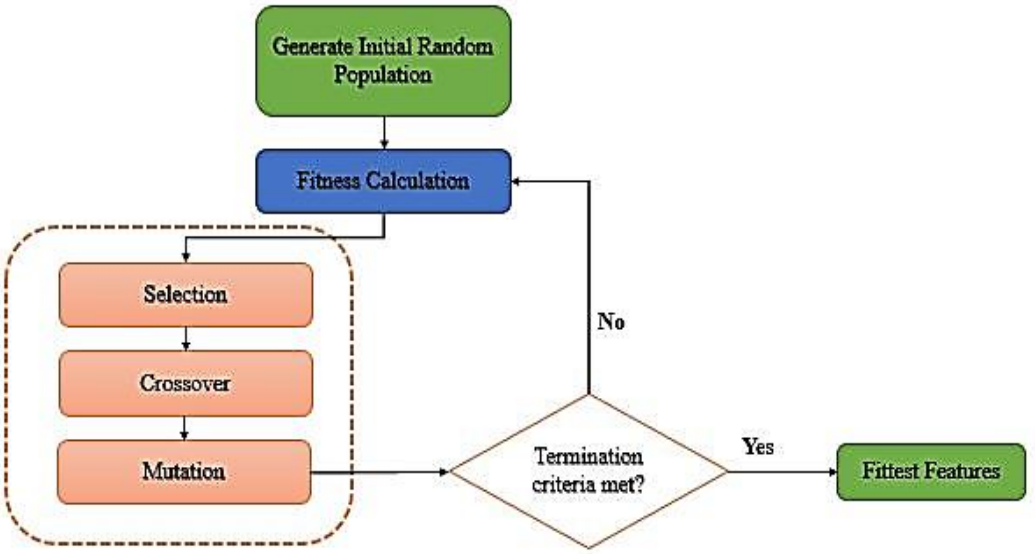


Fig. 3.10: Genetic state

After all previous steps are finished, there will get best selected features from huge set of features. Our reach to our final outcomes. In the next stage, documents will classify with this selected features.

### 3.3.3 Model construction and prediction

Classification methods are employed to measure the strength of the proposed method. This step also for showing the improving classification performance. Here we will use three classifier model to classify our text. Those model are: Naïve Bayes, Support Vector Machine and Decision tree. Finally we will check performance of each model separately.

#### 3.3.3.1 Naïve Bayes classifier

Naïve Bayes classifier is a probabilistic classifier model based on Bayes theorem. It is naïve because all features are thought to be independent or equally independent to target variable. For So, this model is called Naïve Bayes model. It is one of the most used classifier model. I can also do regression work.

This classifier has three types: Gaussian, Bernoulli and Multinomial classifier. Here we have used Gaussian Naïve Bayes classifier for our dataset. We also use this at the time of evolutionary fitness calculation. Finally, we also use this in final classification stages.

A document is said to in a group which have highest posterior probability. This formula of the Naïve Bayes model is like below.

$$P(Class|document) = \frac{P(documents|class)P(class)}{P(document)}$$

Sometime the probability of document is avoided. Then our equation will be like below:

$$P(Class|document) = P(documents|class)P(class)$$

The document final class will be the max value of each class conditional probability. Here "class" is called hypothesis and "Document" is called evidences.

$$Class = Argmax\ P(Class|Document)$$

Naïve Bayes basically classify document with the value of likelihood probability of documents and prior probability of classes.

### 3.3.3.2 Support Vector Machine

Support vector machine is one of the most effective classifier model. It is suited for both linear and nonlinear model. It is basically known for its kernel trick. Kernel trick is a method where data are transformed to n- dimension. There are three types of kernel: linear, polynomial and radial basis function(RBF). In our work we have used linear SVM. Because of its accuracy and distinctness to find out the class of text documents.

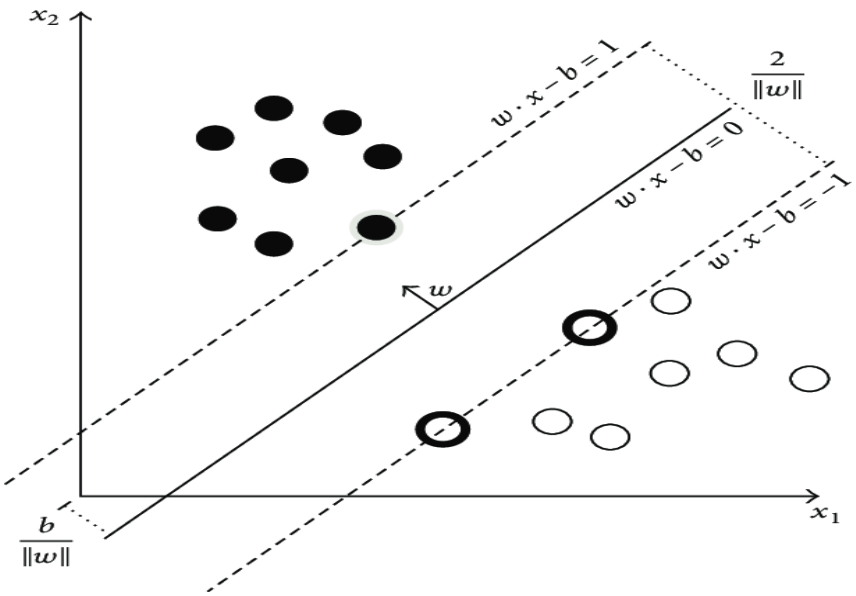


Fig. 3.11: Support Vector Machine

SVM target is to find maximum marginal hyperplane. Margin is difference between nearest support vector differences of two classes. fig.3.11 is the simple illustration of Support Vector Machine. Solid black line is the hyperplane where two classes are divided. The equation of hyperplane is :

$$w.x - b = 0$$

Here, w is random weight, x is support vector. Support vector is composed of the nearest class's points. Support play a vital role in classification and also in model construction. In the equation b is split from y identified. Equation of marginal boundary:

$$Boundary = \frac{2}{||w||}$$

It is calculated like below:

$$w.x_1 - b = 1$$
$$w.x_2 - b = -1$$

Those two equation for two data points of two class: $x_1$ is for first class and $x_2$ for the second class. After substation of two equations we get:

$$x_1 - x_2 = \frac{2}{||w||}$$

Here, $x_1 - x_2$ is the boundary between two classes. This is show in dotted line in fig. 3.11.

#### 3.3.3.3 Decision Tree Classification

Decision Tree is a tree based model where dataset points are divided in another segments where every sub points are think homogenous. Decision Tree Classifier, iteratively divides the data points into sub part by identifying discriminative line lines. It is also discriminative classifier model. It has the ability to show logic like human. Basically it can mimic human logical thinking behaviors.

There are few types of Decision tree classifier: CART, Hunt algorithm, ID3, C4.5 etc. Now question is which one we take first as root. Answer is, find the attribute that best classifies the training data; use this attribute at the root of the tree. Repeat this process at for each branch. So It can be called top down greedy search approach. Now which one is best classifier attribute? Best attributes are identified using few calculations. In CART algorithm there is used Gini impurity calculation. Attribute that has lowest Gini impurity is taken as best to split. In ID3, there was use information gain calculation. Attribute having high information gain is taken as best for split.

Finally, Decision tree classifier is a non-parametric method, easy to understand and one of the most effective algorithm. It has overfitting problems. Besides it needs less data cleaning and not depending on data type of dataset.

# CHAPTER 4

# Implementation Details

Implementation strategy plays a great role for successful experiment of proposed model with preferred dataset. We implement our proposed model on a Bangla Newspaper dataset collected from Prothom alo web site. In our dataset there are two working feature including target class. Following section, we will discuss about how we collect data, create dataset and analyze dataset with visualization.

## 4.1 Data Collection Procedures

We have collect out textual data from one of the well known Bangla News portals website: Prothom Alo. Here we use Python library Beautyfulsoup[10]. Firstly, there was made a HTTP request to Prothom Alo distinct website. The HTTP response was pursed by html purser. Beautiful-soup convert HTTP response to HTML markup text. Then using get_all() function of text tag we gather textual content of html pages. After processing text with regular expression we get clean text of contents. We save out data in csv format. Data collection summary:

- Make a HTTP request using python library **"*requests*"**.
- Catch HTTP response.
- Purse HTTP to HTML using ***BeautyfulSoup***.
- Make a Date range for making request date wise.
- Make HTTP request date wise.
- Purse HTML from response date wise.
- Get text content using <p> html tag.
- Apply regex on text content.
- Preserve the clean text data.

```
import requests
from bs4 import BeautifulSoup

url = 'https://www.prothomalo.com/sports'
response = requests.get(url)


soup = BeautifulSoup(response.content, "html.parser")
```

Fig. 4.1: Code Snip of making HTTP request using Python

## 4.2 Data Processing

Data processing is initial set of works to making out dataset fit for further analysis. We already discuss data processing procedure in previous chapter in preprocessing section. Now I discuss about how those procedures are made. Firstly, tokenization is make up with the help of termination punctuation mark and white spaces between words. We run split function of python with those marks. We get tokenized sentences and words as the outputs. Then we remove uninformative data and token from tokenized outputs. First we remove punctuation marks and unnecessary keywords from tokens using punctuation mark remover function. Then we remove Stop words from out tokens with stop word remover function. Those all functions are made with Python languages. Here we follow Natural Language Processing procedures.

## 4.3 Data Visualization

Data visualization is an important preprocessing task. It help to understand data graphically. What procedure will be taken is dependent on dataset patterns and data points distribution. In our dataset we have 12180 text documents. There are used six categories form all categories of news portal. Following table show the count of each classes.

| **Class** | **Quantity of data** |
|---|---|
| বাংলাদেশ | 6348 |
| আন্তর্জাতিক | 1456 |
| খেলা | 2247 |
| দূর পরবাস | 853 |
| বিনোদন | 789 |
| অনান্য | 487 |
| **Total** | 12180 |

Table 4.1: Class Data Count

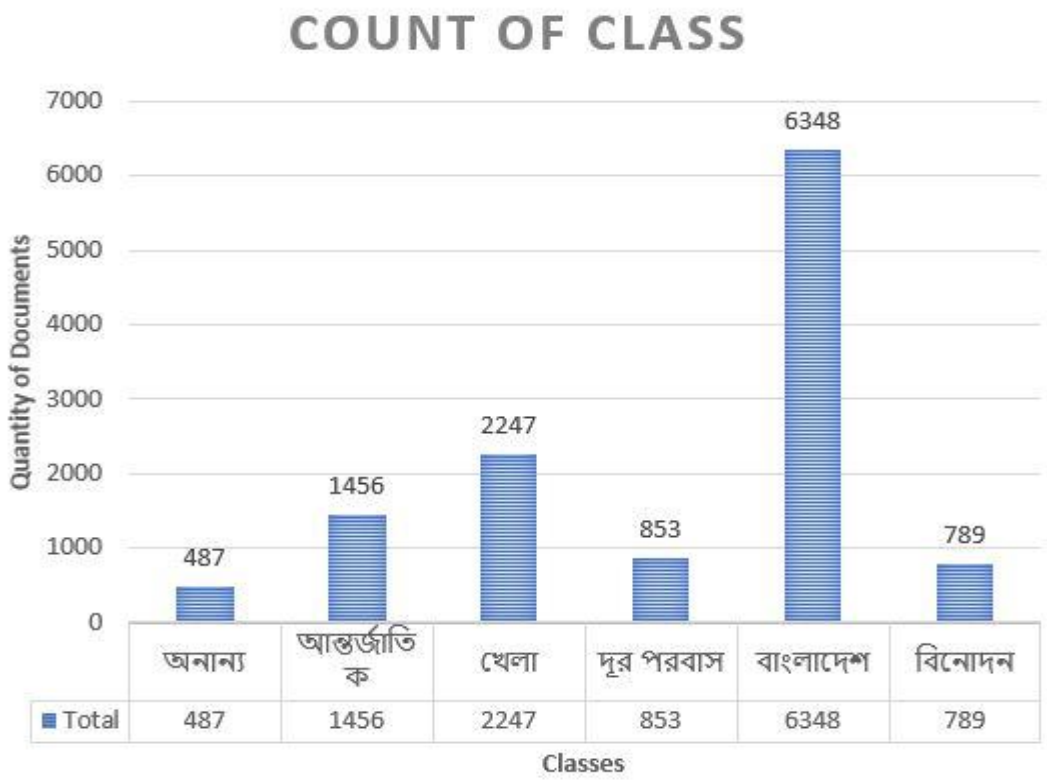


Fig. 4.2: Counts of class Bar diagram

Bar diagram (fig. 4.2) shows the data point distribution for our dataset. Here we see that there are imbalanced data in our dataset. Average length of every documents is 30-40 lines. As a result work with whole dataset is time consuming and memory overload will be hold. For so, we hope to work with representative data. After seeing performance of representative dataset we see there is a pin point fluctuation in performance measurements. As we use 1200 data for our whole procedures.

| **Dataset Type** | **Quantity of Documents** | **Word Counts** | **Unique Features** |
|---|---|---|---|
| Full Dataset | 12180 | 3008504 | 177464 |
| Representative Dataset | 1200 | 334901 | 45450 |

Table 4.2: Words and Feature counts

Here in table 4.2 we see that there is a huge quantity of data features. It is seemed high dimensionality of dataset. Our main focus is to reduce the dimension of the dataset.

## 4.4 Feature Selection and Classification

We previously discussed that our feature selection method is hybrid model where it combination of filter and rapper method. We Use two filter method. At a time one is used. Then in wrapper method we use genetic algorithm. We modify genetic algorithm. We include Mendel's first law in crossover. Feature selection procedure:

- Firstly, filter features with TF-IDF or TF-IDF-ICF.
- Make sorted feature list.
- Form few feature subsets with several size. Here we make four features subsets. Subset sizes are 6K, 10K, 15K and 30K.
- Apply wrapper method on feature subset. One at a time.
- Collect reduced feature.

In genetic algorithm there are few parameters. We tune our parameter like table 4.3.

| **Description** | **Enhance Genetic algorithm Setting** |
|---|---|
| Population Size | 35 |
| Selection Technique | RWS/Rank Selection |
| Crossover type | One Point |
| Crossover rate | 0.9 |
| Mutation rate | 0.02 |
| Generation Number | 200 |

Table 4.3: EGA parameter setting

After feature selection we make classifier model with Naïve Bayes, Decision Tree and SVM classifier. Finally, we evaluate our model with different metrics.

## 4.5 Environment

We have done our experiment on PC having Windows 10 pro operating system. The Pc has Intel Core i5 processor, 4GB ram, 6MB cache. We done all the coding work in Python. Basically we have done raw coding in few steps. We use Sklearn[11] library for classification model and data splitting. All we use are in Anaconda Python Package.

# Chapter 5

# Experimental Result and Discussion

Experimental result is one of the vital part in any type of research. Researchers always try to get the best output from their proposed method. Results can vary from algorithm to algorithm. Researcher select the best one for future experiment. In our work we implement three algorithms: SVM, Naïve Bayes and Decision tree. We calculate Precision, Recall and F1 measure with Accuracy measurement of each algorithm.

## 5.1 Evaluation Metrics

In our experimental work, we calculate Precision, Recall, F1 measure and Accuracy score. We use Macro f1 measure in genetic algorithm fitness calculation. We also calculate confusion matrix for each algorithm.

| | | **Actual Data** | |
|---|---|---|---|
| | | **Positive** | **Negative** |
| **Predicted Data** | **Positive** | True Positive (TP) | False Positive (FP) |
| | **Negative** | False Negative (FN) | True Negative (TN) |

Fig. 5.1: Confusion Matrix

a) Confusion Matrix: Confusion matrix is a tabulated form of performance of test data. Vertically shows predicted values and horizontally shows Actual outputs. It has four sections: True Positive, True Negative, False Positive and False Negative.

- True Positive: This means a model can predict positive class correctly. E.g. Cancer having people is correctly identified.
- False Positive: This means a model predict negative class correctly. It says negative class positive. E.g. People without cancer is incorrectly identified as cancer patient.
- False Negative: This means a model predict positive class correctly. It says positive class negative. E.g. People having cancer is incorrectly identified as without cancer patient.
- True Negative: This means model can predict negative class correctly. E.g. People not having cancer is correctly identified as without cancer patient.

b) Accuracy Score: Accuracy means how correctly our classifier classify the data points. It is basically calculated using division of sum of correctly classify data points by all data samples. It is calculated with following equation:

$$Accuracy = \frac{TP + TN}{Total\ Data}$$

c) Precision: Precision mean how precise our model to correctly predict positive with regard to model positive prediction. E.g. precision is ratio of who has actual cancer and predict having cancer.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

d) Recall: Recall tell how correct our model to identify actual cased with regard to total actual data. It is also called True Positive Rate(TPR).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

e) F1 – measure: Sometime precision and recall are not enough then F1 score calculates enough details. It is the harmonic mean of Precision and Recall.

$$F1\ Measure = \frac{2(Precision \times Recall)}{(Precision + Recall)}$$

## 5.2 Results without Features Selection

Firstly, our dataset is as huge as it can take a whole day for running our features selection operation. So, without processing whole data, we work with representative data. Then we evaluate two datasets without features selection.

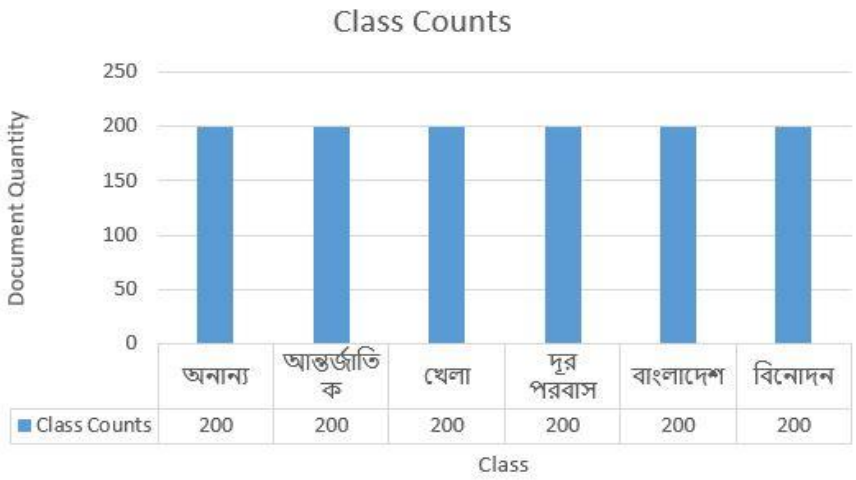


Fig. 5.2: Bar diagram of Representative Data

We get around 73- 87% accuracy, 75-85% precision, 74-87% recall and 74-86% F1-measure scores. We know that we have 12180 documents in our dataset. We create representative dataset for our calculation simplification. Table 5.1 represents the evaluation outputs. Here we see that

there is a little deviation between Sample dataset and whole dataset. So, we can conclude that we can proceed to next stages with representative sample dataset.

| Documents | Classifier Model | Precision | Recall | F1-measure | Accuracy |
|---|---|---|---|---|---|
| Sample Document(1200 documents) | Naive Bayes | 82% | 76% | 78% | 75.23% |
| | SVM | 85% | 87% | 86% | 87.88% |
| | Decision Tree | 75% | 74% | 74% | 74.22% |
| All documents(12108 documents) | Naïve Bayes | 81% | 75% | 78% | 74.81% |
| | SVM | 78% | 78% | 78% | 85.32% |
| | Decision Tree | 75% | 74% | 74% | 73.27% |

Table 5.1: Result without features selection

## 5.3 Result with Features Selection (Filter methods)

As we said before, we use two filter methods for feature selection: TF-IDF, TF-IDF-ICF. We basically give importance on TF-IDF filter method. Because it is simple for ranking words and sentences. We have calculated TF-IDF of each feature terms at first. Then we sort the term in descending pattern with the value of TF-IDF. We then make features subsets with several sizes. We take first 6K feature key make 6K word vector for each sentence. We have done same things for each subset with size 10K, 15K and 30K. Then with train our proposed three model for classification. Finally, we evaluate our classification model.

Table 5.2 shows the evaluation output. Here we see that Support Vector Machine has done best result than other result. For 6K features precision lies between 75% -87%, Recall 74%-87%, F1 measure as Recall and Accuracy lies between 74%-88% . One things is noted that Support Vector Machine gives same performance for all features subset and the evaluation values is around 87% for precision, recall, F1 measurement and accuracy score. On the other hand, Decision tree gives lowest performances among the three classifier model. Its evaluation values is like: 74-77% for precision, 73-76% for recall, same for f1 measure and accuracy lies between 74-76%.

| Filtered Feature Quantity | Classifier Model | Precision | Recall | F1-measure | Accuracy |
|---|---|---|---|---|---|
| 6K | Naive Bayes | 82% | 76% | 79% | 76.2% |
| | SVM | 87% | 87% | 87% | 87.8% |
| | Decision Tree | 75% | 74% | 74% | 74.22% |
| 10K | Naïve Bayes | 83% | 77% | 77% | 77.23% |
| | SVM | 87% | 87% | 87% | 87.02% |
| | Decision Tree | 74% | 73% | 73% | 73.27% |
| 15K | Naïve Bayes | 83% | 77% | 77% | 76.67% |
| | SVM | 87% | 87% | 87% | 87.08% |
| | Decision Tree | 74% | 73% | 73% | 73.33% |
| 30K | Naïve Bayes | 82% | 76% | 76% | 75.84% |
| | SVM | 87% | 87% | 87% | 87.08% |
| | Decision Tree | 77% | 76% | 76% | 75.83% |

Table 5.2: Experiment result with filter features selection.

## 5.4 Result with Features Selection (Wrapper method)

We use genetic algorithm in wrapper method. In filter method, we use four features subsets. Now in wrapper method each features subset is input of wrapper method. We get 1956 features from 6K features, 2305 features from 10K features, 2975 features from 15K features, 3043 features from 30K features set and 3155 features from all features set with 46450 elements. We have make word vector form for each sentences with those final features set. Then we train our classifier models. Finally, we calculate performances measurement. We get result like table 5.3.

Here in table we see that our highest accuracy we get with SVM classifier. The score is 94.08%. There is use 3043 features. We get highest precision score 89% in Naïve Bayes with features 1956 and in SVM with 3155 features. We get the highest recall values 89% in SVM classifier model with 2035, 3043 and 3155 features respectively. We also get the highest F1 measurement in those points. Decision tree give the lowest performance in our dataset and features

set. Its accuracy lies between 75-89.96%. It gives average result in precision, recall and f1 measure. Here one things is noted that we get about 10-18% more accurate result than without features selection dataset evaluation result.

We can say from this table that Support Vector Machine give the best performances in evaluation stages. Naïve Bayes is in second position. Decision tree give the lowest performance.

| Final Feature set | Classifier Model | Precision | Recall | F1-measure | Accuracy |
|---|---|---|---|---|---|
| 1956 | Naive Bayes | 89% | 87% | 88% | 93.75% |
| | SVM | 87% | 87% | 87% | 92.21% |
| | Decision Tree | 80% | 82% | 81% | 89.96% |
| 2305 | Naïve Bayes | 88% | 88% | 88% | 90.09% |
| | SVM | 88% | 89% | 89% | 88.65% |
| | Decision Tree | 81% | 80% | 80% | 91.13% |
| 2975 | Naïve Bayes | 87% | 87% | 87% | 91.877% |
| | SVM | 87% | 87% | 87% | 89.46% |
| | Decision Tree | 84% | 83% | 83% | 88.11% |
| 3043 | Naïve Bayes | 85% | 88% | 86% | 89.590% |
| | SVM | 88% | 89% | 89% | 94.08% |
| | Decision Tree | 87% | 86% | 86% | 75.94% |
| 3155 | Naïve Bayes | 86% | 86% | 86% | 88.578% |
| | SVM | 89% | 89% | 89% | 89.08% |
| | Decision Tree | 85% | 85% | 85% | 87.81% |

Table 5.3: Final Evaluation Result (TF-IDF EGA)

## 5.5 Genetic Algorithm Performances

We enhanced genetic algorithm's performance including Mendel's law and modifying crossover method. We get our satisfactory output in enhanced genetic algorithm. There about 10% -15% fitness is improved from first generation to final generation. Reminding from the previous

chapter, we run evolutionary genetic algorithm through 200 generations. As a result, our proposed method can able to reduce till 93% of full subset size. Fig. 5.3 represent the evaluation graph of each TF-IDF filtered features subsets.

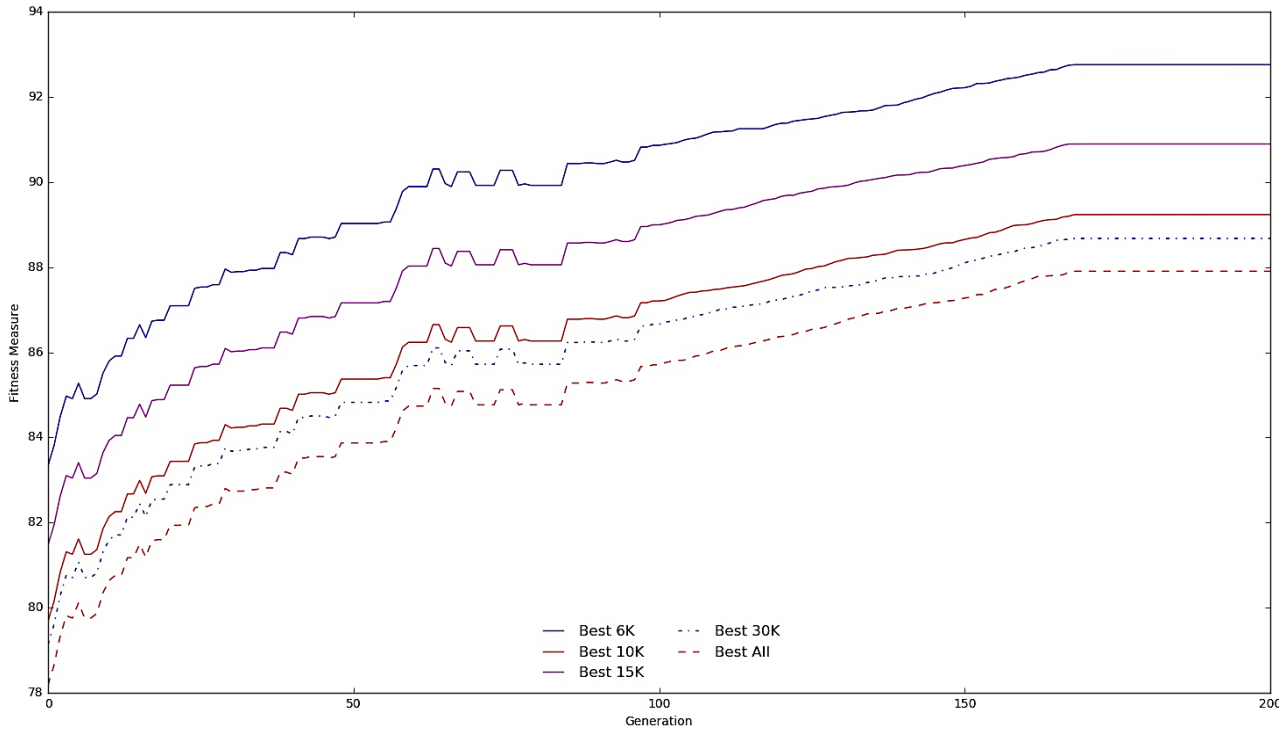


Fig. 5.3 Evaluation Graph

In table 5.4, we can see that the finesses of population is increasing till 200$^{th}$ generation. In fig 5.3, we see that the fitness is changing till around 175 generation. But after the generation, finesses values are not changing at all or simple change is happened. In graph, it shows linear line. Let alone we run our generation till 200$^{th}$ generation. We get about 8% more fitness generation in mid-point generation. Here after 100$^{th}$ generation fitness is increasing till around 175$^{th}$ generation. We can name this section from 100$^{th}$ generation to 175$^{th}$ generation is the golden season for generation. After 175$^{th}$ generation finesses are not changing, thus it shows linear line. We can name this section general stages of population. No, improvement is happening there. It also means the termination of the generation. Because we know that "Survival for the fittest".

| Filtered Features Quantity | Initial Fitness (Fitness before 1st generation) | Mid Generation Fitness(100th generation) | Final Fitness(200th generation) |
|---|---|---|---|
| 6K | 83.33359% | 90.82005% | 92.21412% |
| 10K | 79.67714% | 87.16360% | 88.64943% |
| 15K | 81.46713% | 88.95359% | 90.38799% |
| 30K | 79.13045% | 86.61691% | 88.0964% |
| All Features | 78.17714% | 85.6636% | 87.27342% |

Table 5.4: Population finesses.

## 5.6 Dimension Reduction Analysis

Throughout the research, out main target is to reduce the features size namely dimension reduction. Our result show that we are successful to reduce the dimension of feature search spaces. It also preferable that we are able to reduce computational cost of classification. Table 5.5 show the reduction rate of feature. If we have N features and reduce to M features. Then reduction rate(RR):

$$Reduction\ Rate\ (RR) = 1 - \frac{M}{N}$$

| Total Features | Filtered Features | Final Features | Reduction Rate(Filter) | Reduction Rate(EGA) |
|---|---|---|---|---|
| 46450 | 6K | 1956 | 87.08% | 68.40% |
| | 10K | 2305 | 78.47% | 76.95% |
| | 15K | 2975 | 67.71% | 80.17% |
| | 30K | 3043 | 35.41% | 89.86% |
| | All Features | 3155 | 0% | 93.21% |

Table 5.5: Reduction rate of features

# Chapter 6

# Conclusion and Future Scopes

## 6.1 Conclusion

Our proposed work is mainly focused on feature selection in Bangla text classification. Here we implement hybrid feature selection method for features selection and dimensionality reduction. Firstly, we calculate TF-IDF and TF-IDF-ICF. And sort the key with result and makes few different sized features subset(S). Then we apply genetic algorithm in features subset and get final redacted features. Finally, we classify our text dataset with different classifier using the selected features. Here our main target is to reduce dimension and select the most important features set that can classify text data more accurately. We can get to our goal state but result can be more batter that now.

Our work proposed the enhanced version of genetic algorithm (GA) for features selection in Bangla text classification. The genetic functions are modified to reduce the side effects of randomization and to guide search for the best feature subsets and create population diversity with the useful knowledge. It can be useful to get dimensionality reduction, time and classification performance. The effectiveness of the hybrid FS based on GA will be explored, two hybrid FS approaches are proposed that incorporated two filtering methods with the EGA. The results of the hybrid FS approaches will show that the approaches are more effective in reducing dimensionality and they could produce a higher reduction rate and higher classification precision in most situations compared to single filter method and GA individually. It also be seemed the first approach for Bengali text classification using GA.

As the experimental results indicate that the potentiality of proposed enhanced genetic algorithm was proven individually when it was utilized in the second stages of hybrid features selection approaches. However, the usage of enhanced genetic function is limited in the construction stage in term of its effect on improving feature selection and simplifying the classification process that is performed by another classification algorithm. Thus, the enhanced genetic algorithm as classification algorithm can be used to create a rule based text classifier that combines several advantages instead of using it as preprocessing tool for another algorithm. In addition, the enhancement of genetic algorithm was applied to three operations (selection, crossover and mutation). So it can be possible to make another enhanced version of genetic algorithm modifying other function like fitness function.

As we have investigated on different researches, we come in one conclusion that there is less amount of work in Bangla text classification. Now the quantity of work is increasing. But few of them are fully applicable in term of cost count. Another thing of those work is generalized procedure. In Bangla text classification, we cannot find any effective paper for perfect feature

selection. We also cannot find Bangla text feature selection with genetic algorithm. As a result, our proposed is seemed more effective than those proposed work in term of feature selection.
Finally, we can say that our proposed model is seemed more effective that other Bangla text classification methods with features selection. There are some limitation in my work. One of them are general filter method use. In future we are supposed to model more accurate model for Bangla text classification regarding all limitations. In next section, we will talk about future scope of our proposed model.

## 6.2 Future Scopes

In future, there are a few scope to modify the work. On modification can be done in randomization process. To enhance genetic algorithm functionality dynamic probability for crossover and mutation can be introduced. There will be no probability distribution function that distributes the probabilities of crossover and mutation statically. Rather there will be a random function for probability distribution for crossover and mutation. Another major concerning term in genetic algorithm is fitness function that should be investigated. Fitness measurement can be enhanced with different fitness functions. In our work we have used macro f1 measurement as metrics. Here another function can be used for fitness calculation. For fitness measurement we use Naïve Bayes model. Here another discriminative model like SVM can be used. There can be another modification that can change the evaluation result. That is filter stage's function. In filter stages we use two most effective and simple model to crate features subset. There another functions like mutual information, CDM, pointwise mutual information, LTC can be used. Those are time consuming. As we avoid those function in filter stages. Another suggestion for future to improve enhanced genetic algorithm (EGA) to generate a complete set of useful rules for all text portions for text classification based only on rules. Moreover, performance of the approach could be looked in term of application to healthcare dataset in which the features are more dependent and correlated.