# Enhanced Classification Accuracy on Naive Bayes Data Mining Models

Md. Faisal Kabir &
Chowdhury Mofizur Rahman
Department of CSE, United International University, BD

Alamgir Hossain
Department of Computational Intelligence , Northumbria University,UK

Keshav Dahal
School of Computing, Informatics and Media, University of Bradford, UK

## ABSTRACT

A classification paradigm is a data mining framework containing all the concepts extracted from the training dataset to differentiate one class from other classes existed in data. The primary goal of the classification frameworks is to provide a better result in terms of accuracy. However, in most of the cases we can not get better accuracy particularly for huge dataset and dataset with several groups of data . When a classification framework considers whole dataset for training then the algorithm may become unusuable because dataset consisits of several group of data. The alternative way of making classification useable is to identify a similar group of data from the whole training data set and then training each group of similar data. In our paper, we first split the training data using k-means clustering and then train each group with Naive Bayes Classification algorithm. In addition, we saved each model to classify sample or unknown or test data. For unknown data, we classify with the best match group/model and attain higher accuracy rate than the conventional Naive Bayes classifier.

## General Terms

Artificial Intelligence, Machine Learning, Data Mining.

## Keywords

Classification; Naive Bayes; Clustering; classification accuracy

## 1. INTRODUCTION

Data mining (DM) often referred as knowledge discovery in databases (KDD), is a process of nontrivial extraction of implicit, previously unknown and potentiality useful information from a large volume of data [1][2]. DM is a multi-disciplinary approach comprising of database technology, high performance computing, machine learning, numerical mathematics, statistics and visualization.

The DM algorithms should be computationally feasible for data analysis but takes low human intervention. As mentioned, DM can be performed by using several techniques [3]. Among those techniques, classification [4] is very popular and this technique is being intensively used in many real business applications now-a-days [5]. We first explain classification using theorem of reasoning [6] [7]. According to theorems of reasoning, two main inference techniques that a human uses in reasoning are deduction and induction. Classification involves both induction and deduction. Classification starts with the process of induction reasoning from cases in datasets and later applying these reasoning to predict decisions.

The naïve Bayesian classifier (NB) is one of the most popular data mining techniques for classifying the large dataset. It has been successfully applied to the different problem domains of classification task such as weather forecasting, intrusion detection, image and pattern recognition, medical diagnosis, loan approval and bioinformatics etc [7][8]. Naive Bayesian classifier also efficiently applied in feature selection [9] and web classification [10]. The classification task is to map the set of attributes of sample data onto a set of class labels, and naïve Bayesian classifier particularly suitable as proven universal approximates. Naive Bayesian classifier is a statistical classifier based on the Bayes' Theorem and the maximum posteriori hypothesis. The earliest description of Naive Bayesian classifier is found in [8]. Some of the reasons this classifier is so common and simple that it is easy to implement and fast since the naive assumption of class conditional independence reduces it's computational cost [11]. Experiential studies comparing classification algorithms have found that the naive Bayesian classifier is comparable in performance with decision tree and selected neural network classifiers [12] [13]. Naive Bayes based classifier had helped in large deal to solve many complex DM problems but there is still lot more to be done to improve its performance and accuracy. In our paper we address this issues and showed that our proposed models (ECNBDMM-I and ECNBDMM-II) provide better result in terms of accuracy than the conventional Naive Bayes classifier.

This paper is comprised of three important sections after introduction. Section 2 provides the background study where concepts and theorems will be discussed which will help understanding of our proposed solution to enhance the accuracy in the Naive Bayesian based classifiers. Section 3 is for our proposed solution section, where our proposed framework as the solution will be discussed in details. Section 4 will contain our experimental results. Our proposed framework will be evaluated using benchmark datasets and their results will be presented in this section. Section 5 is summary section of this paper. We conclude our paper and suggest possible future research directions in this section.

## 2. BACKGROUND

### 2.1 Classification

The classification is considered as the discovery of a machine learning model which partitions the given data $D$ (known as dataset or training dataset) into disjoint sets $C$ (class set), where $c_i$ is $i^{th}$ set or class of $C$ and $i \le n, n = \text{total number of classes.}$

In other words, the classification is to map a training dataset into one of several predefined categorical classes. This is supervised learning in nature. Mathematically a class $c_i$ ($i^{th}$ Class) is defined as follows:

$$c_i = \left\{ r_j \in D \middle| Cond_k^i(r_j) \right\}$$

where instance $r_j, j \le m, m = \text{number of training instances,}$ is drawn from the training dataset $D$ after evaluation of a condition $Cond_k^i(\ )$ for $r_j$ being a member of the class $c_i$. In a typical classification problem, $|D| \ge |C|$, which means every class $c_i$ has one or more instances in $D$.

#### *2.1.1* Bayesian Classifier

Naive Bayesian classifier is a statistical classifier based on the Bayes' Theorem and the maximum posteriori hypothesis. The earliest description of Naive Bayesian classifier is found in [14]. Some of the reasons this classier is so common and simple that it is easy to implement and fast since the naive assumption of class conditional independence reduces it's computational cost[11]. Experiential studies comparing classification algorithms have found that the naive Bayesian classifier is comparable in performance with decision tree and selected neural network classifiers [12][13].

In order to explain naive Bayesian classifier, Let $X = \{x_1, x_2, \cdots, x_n\}$ be a case from a dataset, whose feature values made on a set of $n$ attributes. Let $H$ be some hypothesis, such as that the data $X$ belongs to a specific class $C_i$ i.e. $H = X \in C_i$. In naive Bayesian classification, we calculate the probability that sample $X$ belongs to class $C_i$, given that we know the feature values of $X$. Using Bayes' theorem, we write this as follows:

$$P(H \mid X) = \frac{P(H)P(X \mid H)}{P(X)}$$

$$\text{i.e. } P(X \in C_i \mid x_1, x_2, \cdots, x_n) = \frac{P(C_i)P(x_1, x_2, \cdots, x_n \mid C_i)}{P(x_1, x_2, \cdots, x_n)}$$

In plain English, the equation (4) is written as

$$\text{Posterior Probablity} = \frac{\text{Prior Probablity} \times \text{likelihood}}{\text{evidence}} \quad (5)$$

The evidence part in (5) may be ignored since it is a scaling factor i.e. a constant if the values of the feature values are known and the final posterior probability calculation becomes as follows:

(1)

$$\text{Posterior Probablity} = \text{Prior Probablity} \times \text{likelihood} = P(C_i)P(X \mid C_i) \quad (6)$$

Equation (6) can be expressed using conditional probability theory and conditional independence as follows:

$$\text{Posterior Probablity} = P(C_i)P(x_1 \mid C_i)P(x_2 \mid C_i)P(x_3 \mid C_i)\cdots = P(C_i)\prod_{j=1}^{n} P(x_j \mid C_i) \quad (7)$$

In equation (7), $P(C_i) = \frac{1}{\text{number of } C_i \text{ class instances}} = \frac{1}{|C_i|}$ which can be calculated from training dataset easily? However, the calculation for the rest of the part needs is bit difficult. If $x_i$ is discrete variable then this can be calculated from frequency distribution and in the case of continuous then it is calculated from Gaussian probability which is the area under the bell curve. The Gausian probability calculation follows the following formula $g(X, \mu, \sigma) = \frac{1}{\sqrt{2\Pi\sigma}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$

(8)

where $\mu$ is mean and $\sigma$ is estimated standard deviation which and calculated from $\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \mu)^2$.

There could be many special cases which need to be handled, specially, to calculate posterior probability. For example, in the case of any $P(x_j \mid C_i) = 0$ due to $\left| x_j \right| = 0$ i.e. frequency of value $x_j$ in class distribution $C_i$ is zero, which will cancel the effects of all of the other posterior probabilities $P(C_i)$. To avoid this problem, we can assume that our training set is so large that adding one to each count that we need would only make a negligible difference in the estimated probabilities. This technique is known as Laplacian correction & $P(x_j \mid C_i)$ then become $\frac{|1|}{|C_i| + number\ of\ classes}$.

In order to predict the class label of $X$, equation (7) is evaluated for each class $C_i$. The classifier predicts that the class label of $X$ is $C_i$ if and only if it is the class that maximizes equation (7). The classification framework which we will propose in the next section will need to apply equation (7) to find the class label information for the representative instances or cases for the whole dataset. The rational to choose equation (7) is to make the proposed framework scalable since the naïve Bayesian classifier is one of the fastest classifier that we have already mentioned earlier in this section.

### *2.1.2* K-means

The goal of clustering is to reduce the amount of data by categorizing or grouping similar data items together. One of the motivations for using clustering algorithms is to provide automated tools to help in constructing categories or taxonomies. Clustering problems arise in many different applications, including data mining and knowledge discovery data compression. There are various algorithm exists used for clustering. Over various methods k-means is one of the most popular and widely studied clustering methods for points in Euclidean space. The different clusters are formed in such a way that substances in the same cluster are very similar and objects in different clusters are very distinct. Measures of similarity depend on the application. Given a set $D$ of n data points in real n-dimensional space, $R^n$, and an integer K, the problem is to determine a set of k points in $R^n$, called centers, to minimize the mean squared Euclidean distance from each data point to its nearest center. This measure is often called the squared-error distortion [15] and this type of clustering falls into the general category of variance based clustering. Clustering based on k-means is closely related to a number of other clustering and facility location problems. These include the Euclidean k-median and the Weber problem in which the objective is to minimize the sum of distances to the nearest center, and the Euclidean k-center problem [16] in which the objective is to minimize the maximum distance. Clustering based on k-means is closely related to a number of other clustering and location problems. These include the Euclidean k-medians (or the multisource Weber problem) in which the objective is to minimize the sum of distances to the nearest center and the geometric k-center problem [17] in which the objective is to minimize the maximum distance from every point to its closest center. An asymptotically efficient approximation for the k-means clustering problem has been presented by Matousek [18].

In general K-means clustering is a partitioning method. K-means partitioned the data into k mutually exclusive clusters, and returns the index of the cluster to which it has assigned each observation. Unlike hierarchical clustering, k-means clustering operates on actual observations (rather than the larger (9) set of dissimilarity measures), and creates a single level of clusters. K-means clustering is often more suitable than hierarchical clustering for large amounts of data. K-means uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. The result is a set of clusters that are as compact and well-separated as possible as shown in Fig. 1.

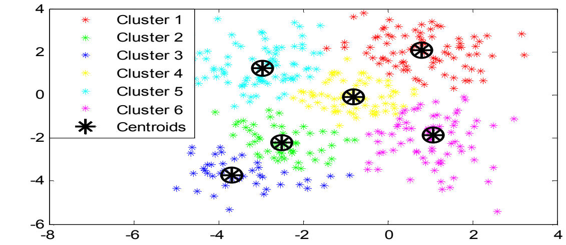

Fig. 1: K-means clustering.

**K-means Algorithm**

```
1. begin initialize n, k, $\mu_i, \ldots \mu_k$
2.     do classify n samples according to nearest $\mu_i$
3.         Recomputed $\mu_i$
4.     until no change in $\mu_i$
5.   return $\mu_1, \mu_2, \ldots, \mu_k$
6. End
```

## 3. PROPOSED ENHANCED CLASSIFICATION ACCURACY ON NAÏVE BAYES DATA MINING MODEL (ECNBDMM)

Our proposed ECNBDMM is shown in Fig.2. This framework has the following main constructs.

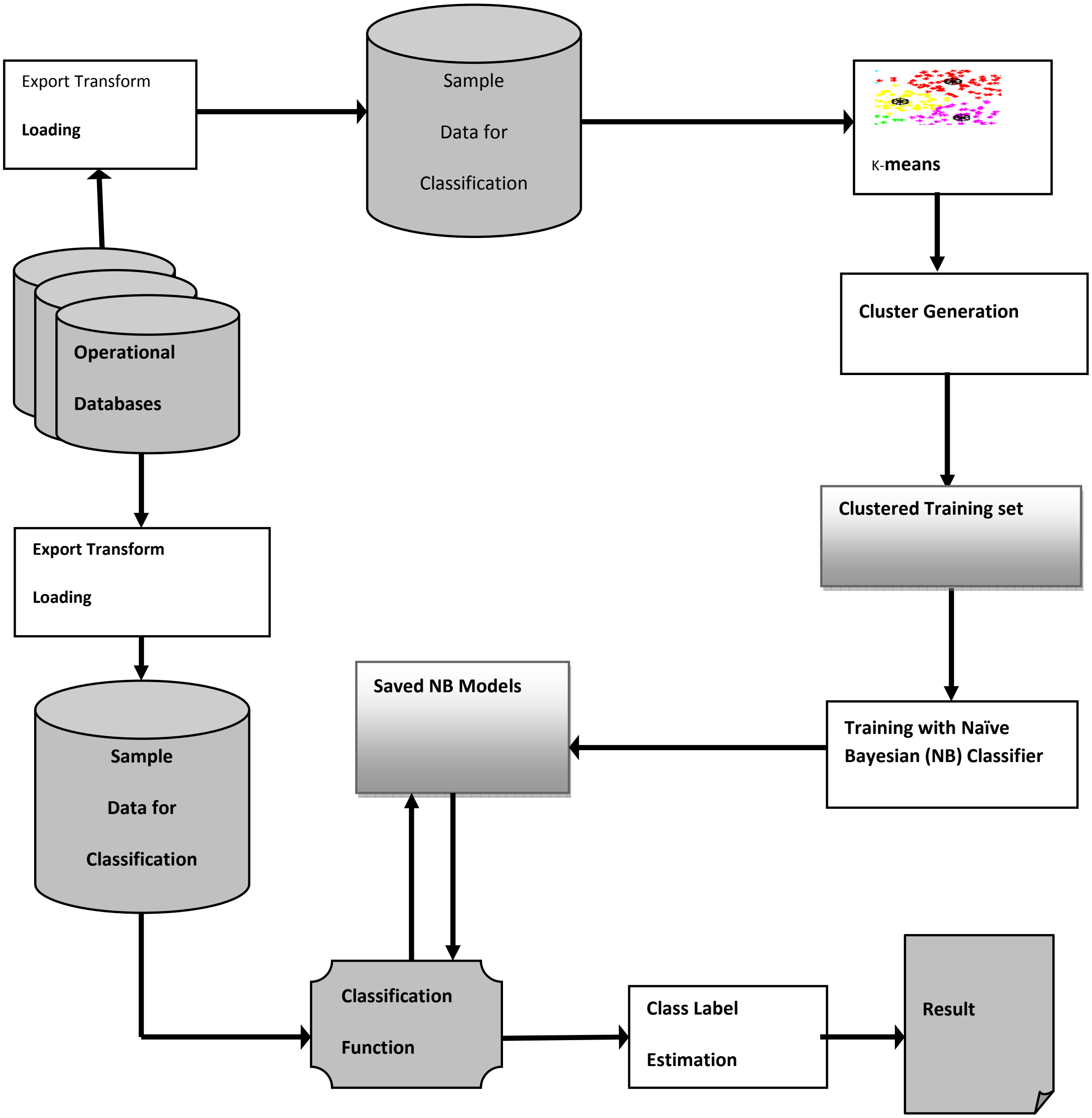

Figure 2. Steps involved in ECNBDMM

i) Operational Databases

These are the operational databases that the business or organization captures from the business flows and transactions. These databases are the primary source for our data selection for both classification model data and sample data for classification.

ii) Classification Model Data and Sample Data for Classification

ETL is a widely used technique for data warehousing. ETL starts with the extraction of data from operational data sources and after extraction it transforms the data to fit it at operational needs by enhancing data quality and integrity. ETL ends with the loading the data into the target data source. Using ETL we construct classification model data and Sample data for classification. The difference between the first set and the second set of the data is that the first one is precedent dataset where class information is known whereas the class information is unknown in the second dataset.

The class information is computed for the second dataset using the proposed ECNBDMM. To explain ECNBDMM consider the two datasets are:

$X = \{X^1, X^2, X^3, \cdots, X^p\}$where
$X^i = \{x_1, x_2, x_3, \cdots, x_{n+1}\}, i \le p$ and

$Z = \{Z^1, Z^2, Z^3, \cdots, Z^p\}$ where

$Z^i = \{z_1, z_2, z_3, \cdots, z_n\}, i \le p$

It should be noted that $(p+1)^{th}$ element of $X$ is the class information. Example of $X$ and z is given in Table 1 and Table 2.

**Table 1: Example of X data set(Training)**

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | Class |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 8 | 8 | 2 | 3 | 4 | 8 | 7 | 8 | 2 |
| 6 | 3 | 3 | 5 | 3 | 10 | 3 | 5 | 3 | 1 |
| 5 | 6 | 7 | 8 | 8 | 10 | 3 | 10 | 3 | 2 |
| 2 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| 5 | 3 | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 2 |
| 2 | 3 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| 5 | 5 | 5 | 2 | 5 | 10 | 4 | 3 | 1 | 2 |
| 10 | 10 | 10 | 3 | 10 | 8 | 8 | 1 | 1 | 2 |
| 4 | 1 | 1 | 3 | 1 | 5 | 2 | 1 | 1 | 2 |
| 1 | 2 | 3 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| 4 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| 5 | 1 | 3 | 1 | 2 | 1 | 1 | 1 | 1 | 1 |

**Table 1: Example of  Z data set(Testing)**

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | Class |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 2 | 1 | 3 | 1 | 1 | ? |

**Tabl3   data of Cluster 2**

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | Class |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| 5 | 3 | 3 | 1 | 3 | 3 | 3 | 3 | 3 | 2 |
| 2 | 3 | 1 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| 4 | 1 | 1 | 3 | 1 | 5 | 2 | 1 | 1 | 2 |
| 1 | 2 | 3 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| 4 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 |
| 5 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | 1 |
| 5 | 1 | 3 | 1 | 2 | 1 | 1 | 1 | 1 | 1 |

**Table4 :data of Cluster 2**

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | Class |
|---|---|---|---|---|---|---|---|---|---|
| 10 | 8 | 8 | 2 | 3 | 4 | 8 | 7 | 8 | 2 |
| 10 | 10 | 10 | 3 | 10 | 8 | 8 | 1 | 1 | 2 |

**Table5 :data of Cluster 3**

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | Class |
|---|---|---|---|---|---|---|---|---|---|
| 6 | 3 | 3 | 5 | 3 | 10 | 3 | 5 | 3 | 1 |
| 5 | 6 | 7 | 8 | 8 | 10 | 3 | 10 | 3 | 2 |
| 5 | 5 | 5 | 2 | 5 | 10 | 4 | 3 | 1 | 2 |

**Table 7: mean value of each cluster**

| A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|
| 3.444 | 1.6667 | 1.7778 | 1.222 | 2 | 1.667 | 2.333 | 1.2222 | 1.222 |
| 10 | 9 | 9 | 2.5 | 6.5 | 6 | 8 | 4 | 4.5 |
| 5.333 | 4.6667 | 5 | 5 | 5.3333 | 10 | 3.333 | 6 | 2.333 |

**Table 6: Centroid value of each cluster**

| Cluster | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3.444 | 1.6667 | 1.7778 | 1.222 | 2 | 1.667 | 2.333 | 1.2222 | 1.222 |
| 2 | 10 | 9 | 9 | 2.5 | 6.5 | 6 | 8 | 4 | 4.5 |
| 3 | 5.333 | 4.6667 | 5 | 5 | 5.3333 | 10 | 3.333 | 6 | 2.333 |

**Table 8: standard deviation of cluster**

| **A1** | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 |
|---|---|---|---|---|---|---|---|---|
| **1.5092** | 0.866 | 0.972 | 0.667 | 0.5 | 1.414 | 0.7071 | 0.6667 | 0.6667 |
| **0.0001** | 1.414 | 1.414 | 0.707 | 4.9497 | 2.828 | 0.0001 | 4.2426 | 4.9497 |
| **0.5774** | 1.528 | 2 | 3 | 2.5166 | 0.0001 | 0.5774 | 3.6056 | 1.1547 |

iii) K-means Clustering, Cluster Generation & Form Clustered Training set

This third step of the proposed framework is very important where we divide or cluster data with K-Means train using the description of Section 2.3 and construct the clustered training data for the classifier. The question is what would be the appropriate value of $k$ for a particular data set. the value of k is chosen by observig the weighted training error usig the following equation:

$$\text{training\_}error = \sum_{i=1}^{k} \left( cluster\_error\_of(\,c_i) \;\times \frac{\mathrm{n_i}}{N} \right)$$

where $\mathrm{C} = \{\mathrm{c_1}, \mathrm{c_2}, \mathrm{c_3}, \ldots \mathrm{c_k}\}$,

$\mathrm{n_i}$ = Number of data of the ith cluster and

N = Total number of training data

Depending on the differnt value of k the equation generate different weighted training error. The initial value of k is set to 2 and then increases untill we reach the stopping criteria. Stopping criteria is determined by continuous increase of weighted training error after few observation. Our goal is to chose the optimum value of k. After selecting the appropriate k value it generates $k$ groups of data from a given data set. Finally, Clustered training data sets are formed. Each clustered data (in this example 3) are shown in table 3 throwgh table 5. Note that, we also saved centroid value of each cluster (shown in Table 6) along with mean (shown in Table 7) and standard deviation (shown in Table 8) which will be used to select the appropriate cluster for unknown data.

iv) Training with Naïve Bayesian (NB) Model and Saved NB Model of each Cluster

The constructed clustered data in step (iii) of our proposed framework is used to build the Naïve Bayesian classifier model which is a statistical classifier based on the Bayes' Theorem and the maximum posteriori hypothesis. We train the each group of training data using NB model and saved the model for prediction of unknown (or test or sample) data.

v) Model Selection and Function Extraction

In this part, we first select the appropriate model from available models done at step (iv) for unknown data. For a particular data, we see which model it belongs to. In our paper we done this in following two ways (ECNBDMM-I & ECNBDMM-II):

a) ECNBDMM-I: We calculate Eucledian Distance (ED) of unknown data and each model's centriod data saved in step (iii). The appropriate model will be selected on the basis of lowest ED value for a particular unknown data. For example, we calculated Eucledian distance of unknown data (Table 3) with the centroid value of each cluster (Table 6) and found 2.8391, 16.9041 and 13.0084 for cluster 1, cluster 2 and cluster 3 respectively. The minimum distance is selected which is cluster 1. That means for this particular data to classify we will use NB model 1 only.

b) ECNBDMM-II: We calculate posterior of each cluster for a sample or unknown data using Gaussian probability. The appropriate model will be selected on the basis of highest posterior value from all calculated cluster. In details, here, we first calculate probabilty density function(pdf) of each attribute by using equation (8) in section 2.3 and then multiply each pdf value along with cluster's probabilty. The mean and standard deviation values are taken from table 7 and table 8 for pdf calculation. The posterior probabilities calculation for 3 clusters are shown below:

Cluster1: $\left(0.0712 * 0.3425 * 0.3999 * 0.3030 * 0.7979 * 0.2524 * 0.3617 * 0.5661 * (9/14)\right) = 4.4343 * \mathrm{e} - 005$

Cluster2: $\left(0.0001 * 0.0001 * 0.0001 * 0.4394\ 0.053 * 0.029 * 0.0001 * 0.0732 * 0.0628 * (2/14)\right) = 4.5525 * \mathrm{e} - 023$

Cluster3: $(0.0001 * 0.0146 * 0.0648 * 0.0807 * 0.065 * 0.0001 * 0.584 * 0.0423 * 0.1774(3/3)) = 4.7321 * \mathrm{e} - 017$

As posterior of cluster 1 is highest, so NB model 1 will be selected for this particular data to classify.

vi) Application and Results

Last part of our framework is to apply appropriate NB model and get class information of the given dataset for business uses.

## 4. EXPERIMENTAL RESULT

U This Section presents the detailed performance evaluation of ECNBDMM (ECNBDMM-I & ECNBDMM-II) in comparison with the conventional Naive bayesian model to enhance prediction accuracy. The primary performance metrics considered in this performance evaluation is accuracy which is also known as the correction rate expressed in percentage.

The thyroid benchmark dataset named as ann.data obtained from http://archive.ics.uci.edu/ml/datasets/ has been considered in evaluation of our proposed ECNBDMM. This dataset consists of 7,200 samples in which 3772 are training or learning examples and 3428 are testing examples (Table 9). The problem is to determine whether a patient reffered to the clinic is hypothyroid. Therefore 3 classes are built : normal (not hypothyroid), hyperfunction and subnormal functioning.

We observe the weighted training error for different value of k started from 2 untill stopping criteria reached. We limit the value of k to 10 as we can see that (Table 9) weighted training error is increasing. We see that the suitable value of k is 6 as weighted training error is minimum (2.9162) than the other k values which is shown in Figure. 5. We also get higher accuracy rate for this cluster for our proposed models.

For this data set we got accuracy rate 94.1365 % for conventional Naive Bayesian classifier where as we attained 95.59% and 96.003% for ECNBDMM-I and ECNBDMM-II respectively (table 9). It clearly shows that our both models provide higher accuracy rate than the conventional one.

We also tested other benchmark data with our models and attained higher classification accuracy rate than conventional naive bayesian classifier. The results are shown in Table 10.

**Table 9: Thyroid data set along with results**

| DATASET | TRAIN | TEST | CLUSTER | NB (%) | WEIGHTED TRAINING ERROR(%) | ECNBDMM –I (%) | ECNBDMM –II (%) |
|---|---|---|---|---|---|---|---|
| THYROID DISEASE (CLASS=3) | **3772** | **3428** | **2** | **94.1365** | **4.5599** | **94.5741** | **94.7116** |
| | | | **3** | | **3.9841** | **95.3617** | **95.2198** |
| | | | **4** | | **3.9767** | **94.9533** | **95.6616** |
| | | | **5** | | **3.4995** | **94.9825** | **95.3449** |
| | | | **6** | | **2.9162** | **95.59** | **96.0035** |
| | | | **7** | | **2.9632** | **94.8075** | **94.1365** |
| | | | **8** | | **3.0813** | **94.9242** | **94.1074** |
| | | | **9** | | **3.1953** | **95.0117** | **95.9032** |
| | | | **10** | | **3.4162** | **95.1867** | **95.249** |

**Table 10: benchmark data set along with results**

| Dataset | No. of class | Cluster | Training data | Testing data | NB (%) | ECNBDMM-I (%) | ECNBDMM-II (%) |
|---|---|---|---|---|---|---|---|
| **Thyroid** | 3 | 10 | 3772 | 3428 | 94.1365 | 95.59 | 96.0035 |
| **Iris** | 3 | 3 | 150 | 68 | 96.0 | 98.53 | 100 |
| **Adult** | 2 | 5 | 32561 | 16281 | 83.13 | 87.38 | 89.97 |
| **Car** | 2 | 7 | 3000 | 2000 | 87.9 | 89.90 | 90.65 |

## 5. SUMMARY AND CONCLUSION

Classification is one of the significant functions of data mining which accurately predicts the target class for each case in the data. In our proposed model we have focused on the improvement of the classification accuracy rates for naïve Bayesian classifier. We tested our proposed model on a number of benchmark data and achieved higher classification accuracy rates than conventional naïve Bayesian classifier. This is because to predict the class for an unknown data we use probability of the appropriate cluster only whereas in conventional naïve Bayesian classifier probabilities are calculated from the whole data. This work is also a good example which advocates achieving better results for a data mining mixing up supervised and unsupervised learning. In our future works, we plan to apply this concept in other classification methods like decision tree, neural networks etc. In addition, we plan to apply this approach in real world problem domains.

## 6. REFERENCES

[1] Agarwal, R., Imielinski, T. and Swami, A. (1993) ‘Database Mining: A Performance Perspective’, IEEE: Special issue on Learning and Discovery in Knowledge-Based Databases, pp. 914-925.

[2] Agarawal, R. and R. Srikant, (1994) ‘Fast algorithms for mining association rules’, Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco, CA., USA., pp: 487-499.

[3] Maindonald, J. H. (1999) ‘New approaches to using scientific data statistics, data mining and related technologies in research and research training’ Occasional Paper 98/2, The Graduate School, Australian National University.

[4] Quinlan, J. (1986), “Induction of Decision Trees,” Machine Learning, vol. 1, pp.81-106.

[5] Berson, A., Smith, S. J. and Thearling, K. (1999) Building Data Mining Applications for CRM McGraw-Hill.

[6] Zarefsky, D., (2002), Argumentation: The Study of Effective Reasoning Parts I and II, The Teaching Company 2002.

[7] Hastie T., Tibshrirani R., and Friedman J.,(2001), The Elements of Statistical Learning, Springer, New York, 2001.

[8] R. O. Duda et al, Pattern Classification, 2nd ed. Chichester, U.K.: Wiley-Interscience, 2000

[9] XHEMALI, Daniela and J. HINDE, Christopher and G. STONE ,Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages, IJCSI International Journal of Computer Science Issues, Vol. 4, No. 1, 2009

[10] A. Sopharak, K. Thet Nwe, Y. Aye Moe, Matthew N. Dailey and B. Uyyanonvara, “Automatic Exudates Detection with a Naive Bayes Classifier”, Proceedings of the International Conference on Embedded Systems and Intelligent Technology, pp. 139-142, February 27-29, 2008.

[11] Domingos, P., and Pazzani, M.,(1996), ‘Beyond independence: conditions for the optimality of the simple Bayesian classifier’, Proceedings of ICML 1996.

[12] Leung, K., (2007), ‘Naive Bayesian Classifier’, Technical Report, Department of Computer Science / Finance and Risk Engineering, Polytechnic University, Brooklyn, New York, USA.

[13] Networks in the Classification of Training Web Pages, IJCSI International Journal of Computer Science Issues, Vol. 4, No. 1, 2009

[14] Matousek, ‘On Approximate Geometric k-clustering’, Discrete and Computational Geometry, vol. 24, pp. 61-84, 2000

[15] Jain, A and R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, N.J.: Prentice Hall, 1988.

[16] Gersho, A. and R.M. Gray, Vector Quantization and Signal Compression. Boston: Kluwer Academic, 1992.

[17] Dasgupta, ‘Learning Mixtures of Gaussians’, Proc. 40th IEEE Symp. Foundations of Computer Science, pp. 634-644, Oct. 1999.

[18] Irina, R., (2001). ‘An empirical study of the naive Bayes classifier’. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence.