International Journal of Engineering Science Invention (IJESI)
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 7 Issue 1|| January 2018 || PP. 62-69

Comparison of Different Classifiers Using Medical Domain Datasets

Susmita Nandi[1], Kasturi Ghosh[2]

[1]*(Computer Science and Engineering, University Institute of Technology/ Burdwan University, India)*
[2]*(Information Technology, University Institute of Technology / Burdwan University, India)*
Corresponding Author: Susmita Nandi[1]

Abstract : *Now a day's proper medical diagnostic is very much important. Manual diagnosis results in some incorrect outputs. As a result wrong treatment is applied on the patients. So automation of the diagnosis process is important. Application of different Data Mining algorithms proved to be efficient for automatic diagnostic system. For doing this job application of different Data Mining classifiers are important. Hence comparison among several classifiers is required to find out the best one for better performance of the system.*

Keywords : *Classification Methods, Decision tree, Discretization, J48, k-NN, Naive Bayes, Pre-processing*

Date of Submission: 13-01-2017 Date of acceptance: 27-01-2018

I. Introduction

Classification is a data mining function which assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each instance in the dataset. The simplest type of classification problem is binary classification. In this type of classification, target attribute has two possible values. Multiclass targets have more than two values: for example, low, medium, high. Classification models are tested by applying the technique to test data with known target values and comparing the predicted values with the known values. The test data must be compatible with the data used to build the model and must be prepared in the same way that the build data (training set) was prepared.

1.1. Classification methods

Classification is the separation or ordering of objects into classes[1]. There are two phases in classification algorithm: first, the algorithm tries to find a model for the class attribute as a function of other variables of the datasets. In the second phase, it applies previously designed models on the new datasets for determining the related class of each instance[2]. Classification has been applied in many fields such as medical, astronomy, commerce, biology, media, etc. There are several techniques in classification method like: Decision Tree, Naïve Bayes, k-Nearest Neighbor, Support Vector Machine etc. In this paper three techniques are used for comparison Decision Tree, Naïve Bayes, and k-Nearest Neighbor.

1.2. Pre-processing

Data pre-processing is an important step in the data mining applications. Methods of gathering data from multiple sources are often loosely controlled, which leads to out-of-range or impossible data combinations, missing values etc. Analyzing data that has not been carefully pre-processed may produce incorrect results[3]. Thus, maintaining quality of data is very important before running an analysis. If there is inconsistent and redundant information present in the dataset then knowledge discovery during the training stage is more difficult. Data pre-processing includes cleaning, Instance selection, normalization etc. The outcome of data pre-processing is the final training set.

1.3. Decision Tree

In a decision tree each internal node denotes a test on an attribute of the dataset, each branch of the tree represents an outcome of that particular test, and leaf nodes denote classes or class distributions[4]. Decision trees are used to model problems in which a series of decisions leads to a solution. The possible solutions correspond to the paths starting from the root to the leaves of the tree. As the name implies, this technique recursively separates observations in branches to construct a tree. Most decision tree classifiers perform classification in two steps: tree-growing and tree-pruning. First one is done in top-down manner. During this step the tree is recursively partitioned. This process continues till all the data items belong to some class label. In the second step the full grown tree is cut back to prevent over fitting. It also improves the accuracy. So, basically the prediction and classification accuracy of the algorithm is improved by minimizing the over-fitting. Compared to other techniques, this method is broadly applied in various areas as it is robust to data scales and distributions. The popular Decision Tree algorithms are ID3, C4.5. The ID3 algorithm is considered as a very simple decision tree algorithm. It uses information gain as splitting criteria. C4.5 is an evolution of ID3. It uses gain ratio as splitting criteria[5].

1.4. Discretization

The datasets used here consist of continuous data. So, to evaluate the performance of Naïve Bayes algorithm they are discretized. Discretization is one of the commonly used data pre-processing technique to improve the efficiency of the knowledge extraction process on clinical data. Generally, clinical data contains numeric attributes with continuous values. Data discretization simplifies the original data by transforming continuous data attribute values into a finite set of intervals. Although discretization is capable of handling continuous attributes on clinical data, there are cases where discretization is not an appropriate technique for handling continuous attributes. Discretization techniques can be classified into two categories: unsupervised and supervised[6]. Unsupervised methods simply apply a prescribed scheme to discretize the continuous value without making use of the attribute-class information, whereas supervised methods take into consideration the attribute-class information. A typical problem of unsupervised methods is that it is difficult to determine how many intervals are the best for a given attribute. Theoretically, directed by class information, supervised discretization methods can automatically determine the best number of intervals for each continuous attribute for classification. In this case to discretize the datasets Equal Interval Binning Method is used which is an unsupervised method.

1.5. Naive Bayes algorithm

Naive Bayes classifier is a probabilistic classifier and is based on the Bayes theorem. Naïve Bayesian classifiers assume that there are no dependencies amongst attributes of a dataset. This assumption is called class conditional independence. Rather than predictions, the Naïve Bayes classifier produces probability estimates. For each class value they estimate the probability that a given instance belongs to that class. Small amount of training data is required to estimate the parameters necessary for classification. This is an advantage of the Naive Bayes classifier[7].

Product rule: $P(A \wedge B) = P(A|B)\,P(B) = P(B|A)P(A)$ (1)

Sum rule: $P(A \vee B) = P(A)+P(B)-P(A \wedge B)$ (2)

Bayes theorem: $P(h|D) = \frac{P(D|h)P(h)}{P(D)}$ (3)

Theorem of total probability, if event Ai is mutually exclusive and probability sum to 1.

$P(B) = \sum_{i=1}^{n} P(B|Ai)P(Ai)$ (4)

Given a hypothesis h and data D which bears on the hypothesis:

P(h): independent probability of h: prior probability

P(D): independent probability of D

P(D|h): conditional probability of D given h: likelihood

P(h|D): conditional probability of h given D: posterior probability

1.6. K Nearest Neighbour classification

k-nearest neighbor algorithm (k-NN) is a method to classify an object based on the majority class amongst its k-nearest neighbors. It is a type of Lazy learning where the function is only approximated locally and all computation is deferred until classification. An instance is classified by majority of its neighbors. K is always a positive integer. Neighbors are selected from a set of instances for which the correct class value is known. In WEKA this classifier is known as IBK. The k-NN algorithm for continuous-valued target functions calculates the mean values of the k nearest neighbors. K-NN algorithm usually uses the Euclidean or the Manhattan distance. However, any other distance calculating methods can also be used[8]. In this experiment, Euclidean distance is used. Suppose the instance has coordinates (a1, b1) and the coordinate of training sample is (c1, d1) then square Euclidean distance:

$x^2 = (c1 - a1)^2 + (d1 - b1)^2$ (5)

1.7. C4.5 or j48 algorithm

This algorithm was proposed in 1993, again by Ross Quinlan, to overcome the limitations of ID3 algorithm discussed earlier. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy [9]. One limitation of ID3 is that it is overly sensitive to features with large numbers of values. To overcome this problem, C4.5 uses "Information gain ratio". Gain ratio, is defined as follows:

GainRatio (p,T) = $\frac{\mathrm{Gain(p,T)}}{\mathrm{SplitInfo(p,T)}}$(6)

SplitInfo(p,test) = $-\sum_{u=1}^{n} p'\left(\frac{u}{p}\right) \times \log(\ p'\left(\frac{u}{p}\right)\ )$ (7)

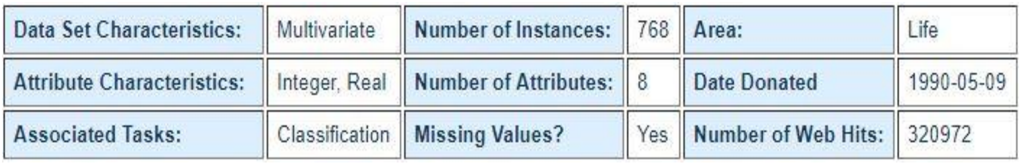
P'(u/p) - proportion of elements present at the position p, taking the value of u-th test. At each node, C4.5 chooses the attribute that most effectively splits the samples into subsets enriched in one class or the other. The splitting criterion is the information gain ratio. The attribute with the highest information gain ratio is chosen to make the decision. Then the algorithm recurs on the smaller sub lists.

1.8. Dataset description

Here three datasets are used to evaluate the performance of three classification algorithms.

Pima Indians Diabetes Data Set (PIDD) - Attribute Information: [10]

- Number of times pregnant
- Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- Diastolic blood pressure (mm Hg)
- Triceps skin fold thickness (mm)
- 2-Hour serum insulin (mu U/ml)
- Body mass index (weight in kg/(height in m)^2)
- Diabetes pedigree function
- Age (years)
- Class variable (0 or 1)

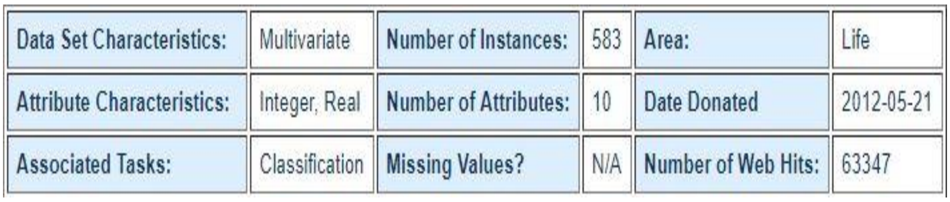


Data Set Characteristics:	Multivariate	Number of Instances:	768	Area:	Life
Attribute Characteristics:	Integer, Real	Number of Attributes:	8	Date Donated	1990-05-09
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	320972

Fig. 1. Pima Indians Diabetes Data Set Details

Indian Liver Patient Dataset (ILPD) – AttributeInformation: [11]

- Age of the patient
- Gender of the patient
- TB Total Bilirubin
- DB Direct Bilirubin
- Alkphos Alkaline Phosphatase
- Sgpt Alamine Aminotransferase
- Sgot Aspartate Aminotransferase
- TP Total Protiens
- ALB Albumin
- A/G Ratio Albumin and Globulin Ratio
- Selector field used to split the data into two sets (labeled by the experts)



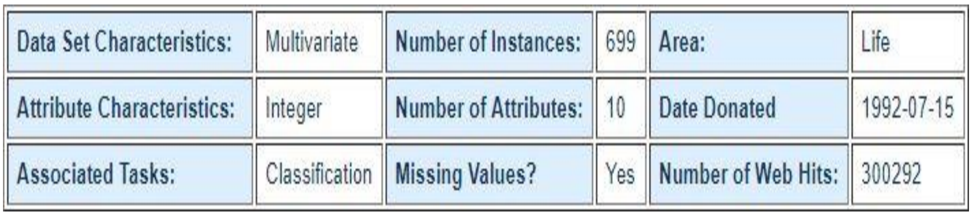
Data Set Characteristics:	Multivariate	Number of Instances:	583	Area:	Life
Attribute Characteristics:	Integer, Real	Number of Attributes:	10	Date Donated	2012-05-21
Associated Tasks:	Classification	Missing Values?	N/A	Number of Web Hits:	63347

Fig. 2. Indian Liver Patient Dataset Details

Breast Cancer Wisconsin (Original) Data Set- Attribute Information: [12]

- Sample code number: id number
- Clump Thickness: 1 - 10
- Uniformity of Cell Size: 1 - 10
- Uniformity of Cell Shape: 1 - 10
- Marginal Adhesion: 1 - 10
- Single Epithelial Cell Size: 1 - 10

- Bare Nuclei: 1 - 10
- Bland Chromatin: 1 - 10
- Normal Nucleoli: 1 - 10
- Mitoses: 1 - 10
- Class: (2 for benign, 4 for malignant)



Data Set Characteristics:	Multivariate	Number of Instances:	699	Area:	Life
Attribute Characteristics:	Integer	Number of Attributes:	10	Date Donated	1992-07-15
Associated Tasks:	Classification	Missing Values?	Yes	Number of Web Hits:	300292

Fig. 3. Breast Cancer Wisconsin (Original) Data Set Details

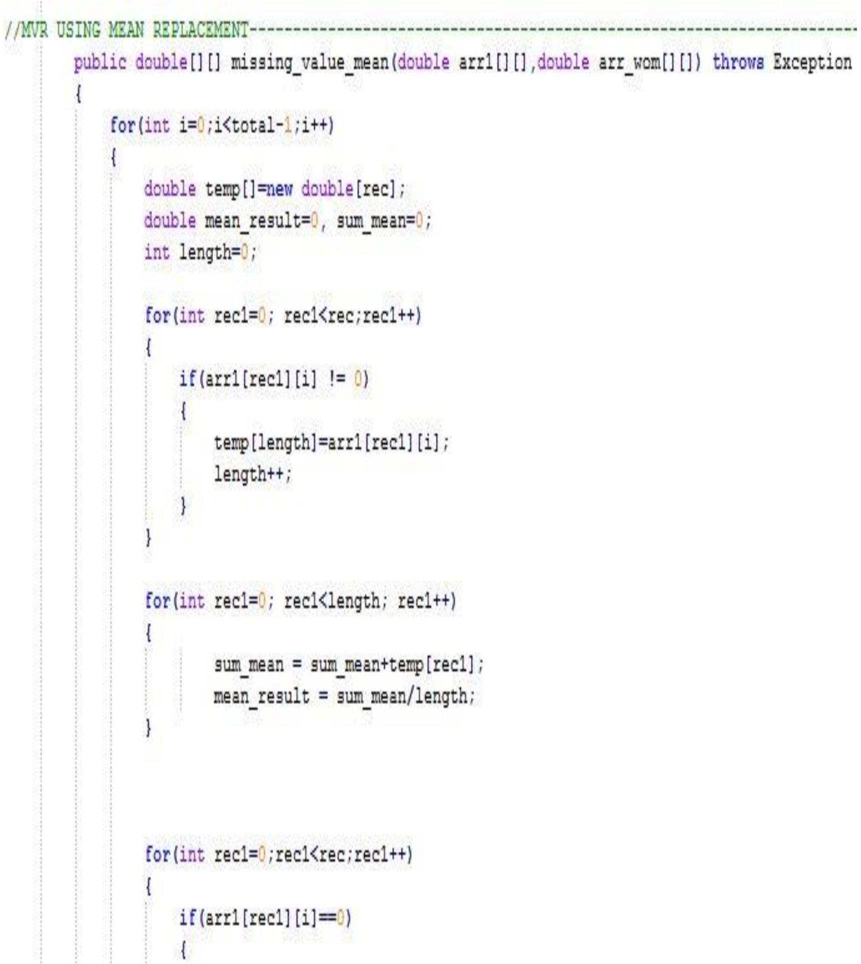
2. Work done

2.1. Comparison of Different Classifiers (CDC)

Input. PIDD, ILPD, BCWD Datasets

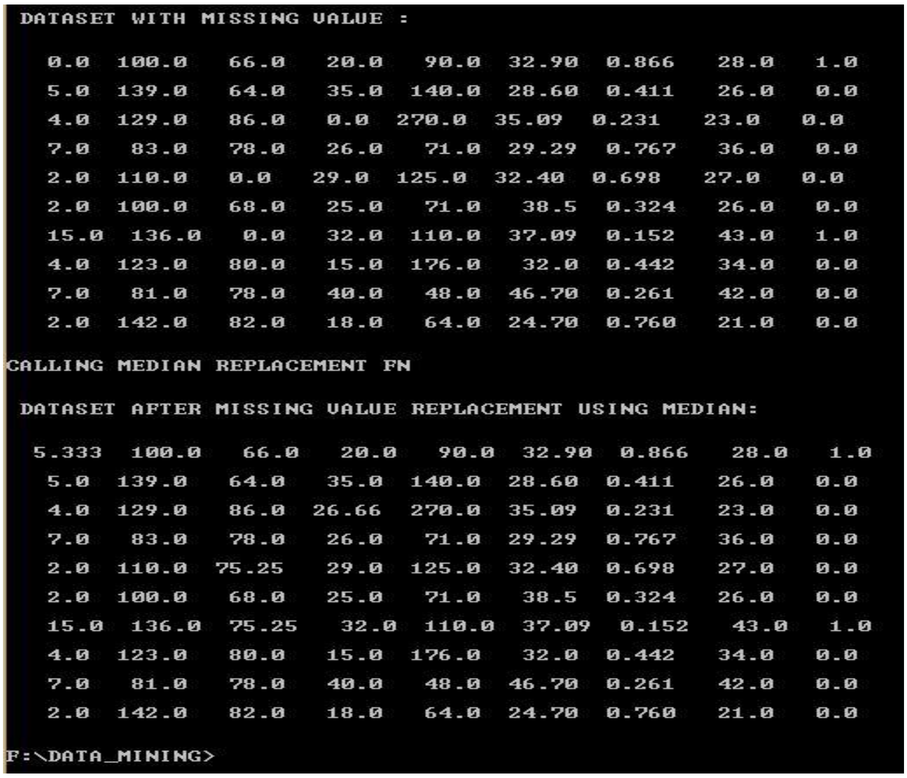
Output. Error Rate in classification

Step1. Pre-processing is an important step of Datamining projects. One important task in pre-processing stage is replacement of missing values. Missing values may lead to wrong classification hence before applying the classification algorithms all three datasets are pre-processed. In this step missing values are replaced using mean valuereplacement approach. In this approach missing values of an attribute are replaced by mean of the domain of that attribute.



```
//MVR USING MEAN REPLACEMENT----------------------------------------------------------------------
    public double[][] missing_value_mean(double arr1[][],double arr_wom[][]) throws Exception
    {
        for(int i=0;i<total-1;i++)
        {
            double temp[]=new double[rec];
            double mean_result=0, sum_mean=0;
            int length=0;

            for(int rec1=0; rec1<rec;rec1++)
            {
                if(arr1[rec1][i] != 0)
                {
                    temp[length]=arr1[rec1][i];
                    length++;
                }
            }

            for(int rec1=0; rec1<length; rec1++)
            {
                    sum_mean = sum_mean+temp[rec1];
                    mean_result = sum_mean/length;
            }



            for(int rec1=0;rec1<rec;rec1++)
            {
                if(arr1[rec1][i]==0)
                {
```

Fig. 4. Pre-processing using mean replacement approach

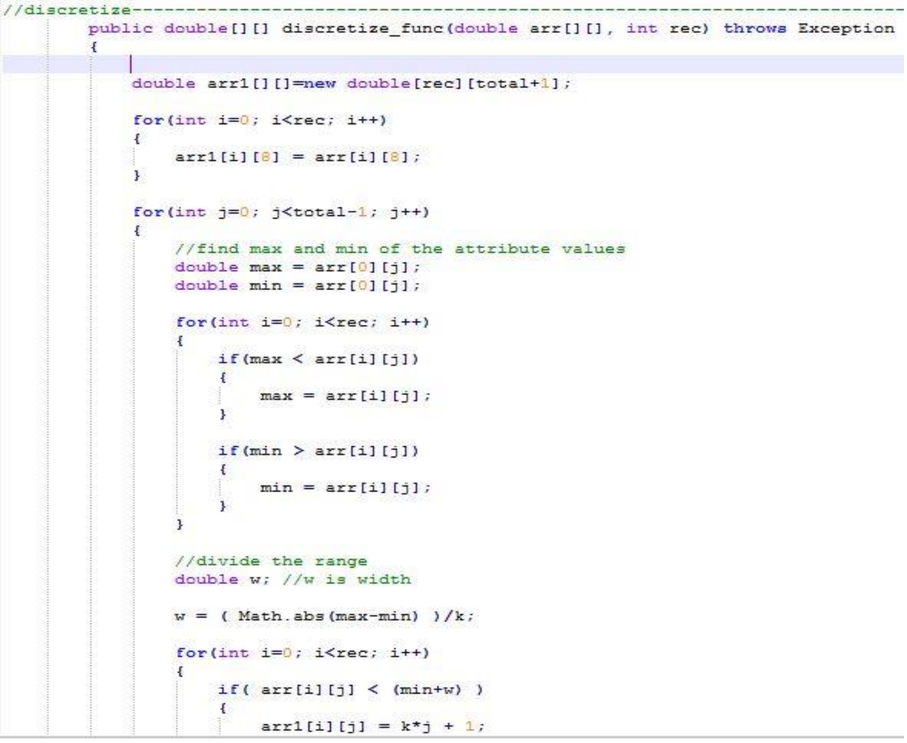


```
 DATASET WITH MISSING VALUE :

   0.0  100.0   66.0   20.0    90.0  32.90  0.866   28.0   1.0
   5.0  139.0   64.0   35.0  140.0  28.60  0.411   26.0   0.0
   4.0  129.0   86.0   0.0  270.0  35.09  0.231   23.0   0.0
   7.0   83.0   78.0   26.0    71.0  29.29  0.767   36.0   0.0
   2.0  110.0   0.0   29.0  125.0  32.40  0.698   27.0   0.0
   2.0  100.0   68.0   25.0    71.0   38.5  0.324   26.0   0.0
   15.0  136.0   0.0   32.0  110.0  37.09  0.152   43.0   1.0
   4.0  123.0   80.0   15.0  176.0   32.0  0.442   34.0   0.0
   7.0   81.0   78.0   40.0    48.0  46.70  0.261   42.0   0.0
   2.0  142.0   82.0   18.0    64.0  24.70  0.760   21.0   0.0

CALLING MEDIAN REPLACEMENT FN

 DATASET AFTER MISSING VALUE REPLACEMENT USING MEDIAN:

  5.333  100.0   66.0   20.0    90.0  32.90  0.866   28.0   1.0
   5.0  139.0   64.0   35.0  140.0  28.60  0.411   26.0   0.0
   4.0  129.0   86.0  26.66  270.0  35.09  0.231   23.0   0.0
   7.0   83.0   78.0   26.0    71.0  29.29  0.767   36.0   0.0
   2.0  110.0  75.25   29.0  125.0  32.40  0.698   27.0   0.0
   2.0  100.0   68.0   25.0    71.0   38.5  0.324   26.0   0.0
   15.0  136.0  75.25   32.0  110.0  37.09  0.152   43.0   1.0
   4.0  123.0   80.0   15.0  176.0   32.0  0.442   34.0   0.0
   7.0   81.0   78.0   40.0    48.0  46.70  0.261   42.0   0.0
   2.0  142.0   82.0   18.0    64.0  24.70  0.760   21.0   0.0

F:\DATA_MINING>
```

Fig. 5. View of PIDD dataset before and after missing value replacement

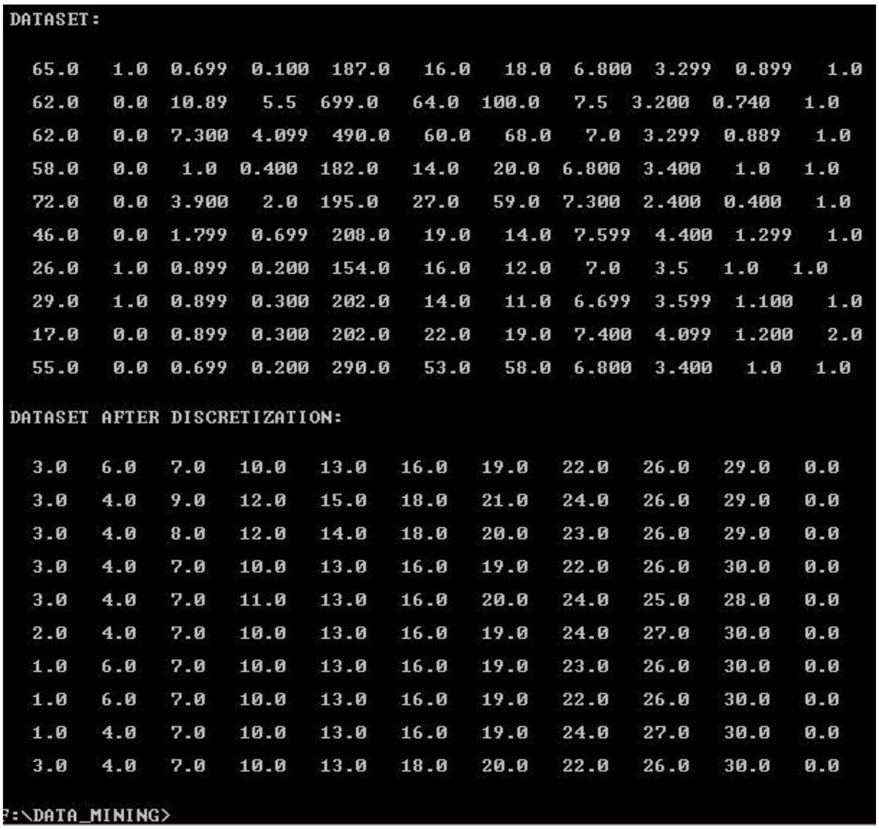
Step2. Datasets are divided into two parts after they are pre-processed.2/3rd part of the dataset is used for training purpose and 1/3rd part is used in testing.

Step3. All datasets are discretized since Naïve Bayes algorithm cannot handle continuous data. Equal Interval Binning method is used for discretization. For each attribute maximum and minimum value within that range is searched. For k intervals width of an interval is calculated. Using boundary values min+w, min+2w, min+(k-1)w… continuous values are grouped.



```
//discretize---------------------------------------------------------------------------
        public double[][] discretize_func(double arr[][], int rec) throws Exception
        {
            
            double arr1[][]=new double[rec][total+1];

            for(int i=0; i<rec; i++)
            {
                arr1[i][8] = arr[i][8];
            }

            for(int j=0; j<total-1; j++)
            {
                //find max and min of the attribute values
                double max = arr[0][j];
                double min = arr[0][j];

                for(int i=0; i<rec; i++)
                {
                    if(max < arr[i][j])
                    {
                        max = arr[i][j];
                    }

                    if(min > arr[i][j])
                    {
                        min = arr[i][j];
                    }
                }

                //divide the range
                double w; //w is width

                w = ( Math.abs(max-min) )/k;

                for(int i=0; i<rec; i++)
                {
                    if( arr[i][j] < (min+w) )
                    {
                        arr1[i][j] = k*j + 1;
```

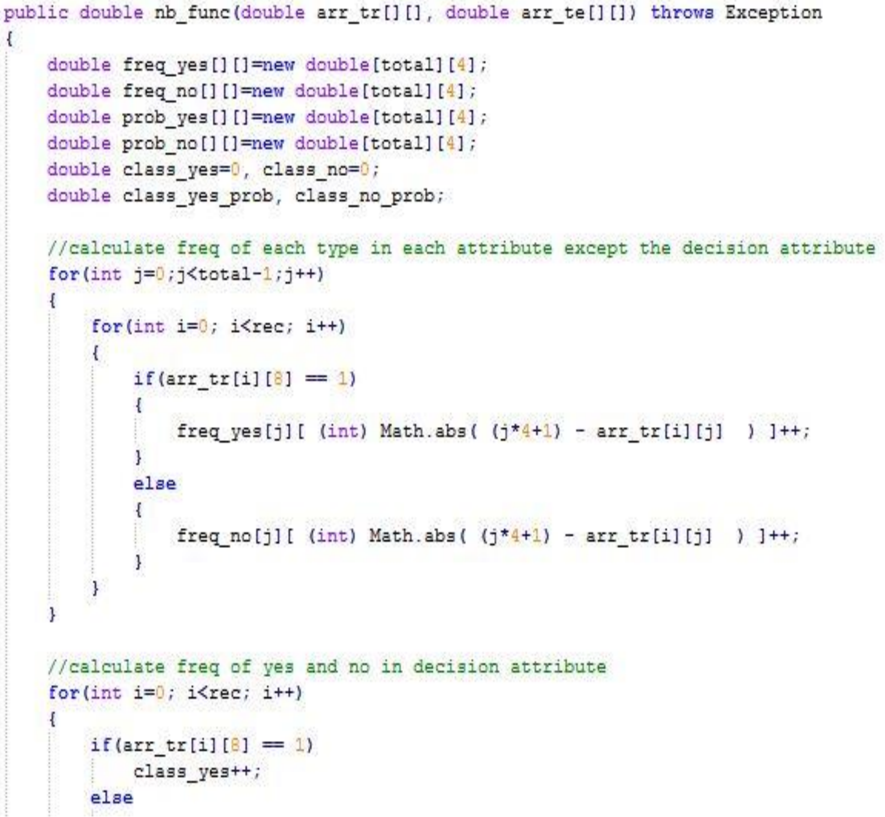
Fig. 6. Equal Interval Binning Implementation



```
DATASET:

  65.0   1.0  0.699  0.100  187.0   16.0   18.0  6.800  3.299  0.899   1.0
  62.0   0.0  10.89   5.5  699.0   64.0  100.0   7.5  3.200  0.740   1.0
  62.0   0.0  7.300  4.099  490.0   60.0   68.0   7.0  3.299  0.889   1.0
  58.0   0.0   1.0  0.400  182.0   14.0   20.0  6.800  3.400   1.0   1.0
  72.0   0.0  3.900   2.0  195.0   27.0   59.0  7.300  2.400  0.400   1.0
  46.0   0.0  1.799  0.699  208.0   19.0   14.0  7.599  4.400  1.299   1.0
  26.0   1.0  0.899  0.200  154.0   16.0   12.0   7.0   3.5   1.0   1.0
  29.0   1.0  0.899  0.300  202.0   14.0   11.0  6.699  3.599  1.100   1.0
  17.0   0.0  0.899  0.300  202.0   22.0   19.0  7.400  4.099  1.200   2.0
  55.0   0.0  0.699  0.200  290.0   53.0   58.0  6.800  3.400   1.0   1.0

DATASET AFTER DISCRETIZATION:

  3.0   6.0   7.0   10.0   13.0   16.0   19.0   22.0   26.0   29.0   0.0
  3.0   4.0   9.0   12.0   15.0   18.0   21.0   24.0   26.0   29.0   0.0
  3.0   4.0   8.0   12.0   14.0   18.0   20.0   23.0   26.0   29.0   0.0
  3.0   4.0   7.0   10.0   13.0   16.0   19.0   22.0   26.0   30.0   0.0
  3.0   4.0   7.0   11.0   13.0   16.0   20.0   24.0   25.0   28.0   0.0
  2.0   4.0   7.0   10.0   13.0   16.0   19.0   24.0   27.0   30.0   0.0
  1.0   6.0   7.0   10.0   13.0   16.0   19.0   23.0   26.0   30.0   0.0
  1.0   6.0   7.0   10.0   13.0   16.0   19.0   22.0   26.0   30.0   0.0
  1.0   4.0   7.0   10.0   13.0   16.0   19.0   24.0   27.0   30.0   0.0
  3.0   4.0   7.0   10.0   13.0   18.0   20.0   22.0   26.0   30.0   0.0

F:\DATA_MINING>
```

Fig. 7. View of ILPD dataset before and after discretization

Step4. Input datasets are fed into the classification algorithms and results are recorded for evaluation.



```java
public double nb_func(double arr_tr[][], double arr_te[][]) throws Exception
{
    double freq_yes[][]=new double[total][4];
    double freq_no[][]=new double[total][4];
    double prob_yes[][]=new double[total][4];
    double prob_no[][]=new double[total][4];
    double class_yes=0, class_no=0;
    double class_yes_prob, class_no_prob;

    //calculate freq of each type in each attribute except the decision attribute
    for(int j=0;j<total-1;j++)
    {
        for(int i=0; i<rec; i++)
        {
            if(arr_tr[i][8] == 1)
            {
                freq_yes[j][ (int) Math.abs( (j*4+1) - arr_tr[i][j]  ) ]++;
            }
            else
            {
                freq_no[j][ (int) Math.abs( (j*4+1) - arr_tr[i][j]  ) ]++;
            }
        }
    }

    //calculate freq of yes and no in decision attribute
    for(int i=0; i<rec; i++)
    {
        if(arr_tr[i][8] == 1)
            class_yes++;
        else
```

Fig. 8. Naïve Bayes Classification algorithm implementation

2.2. Workflow

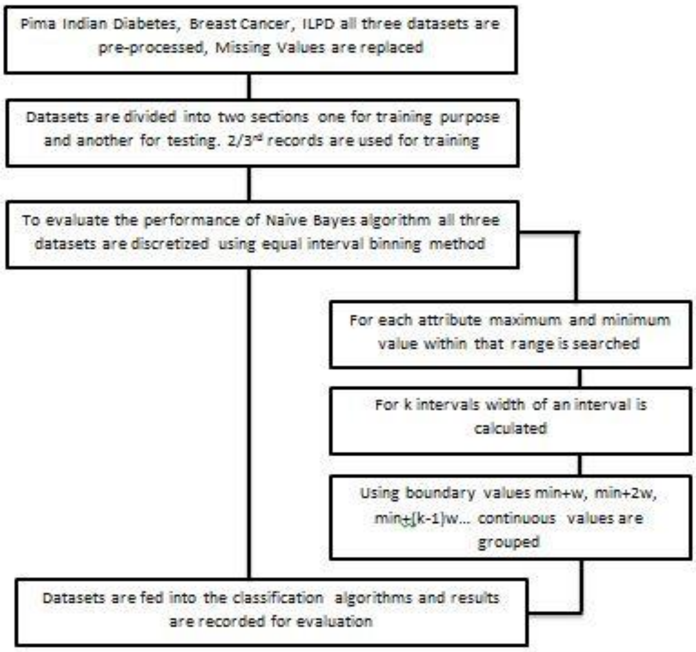

Pima Indian Diabetes, Breast Cancer, ILPD all three datasets are pre-processed, Missing Values are replaced
Datasets are divided into two sections one for training purpose and another for testing. 2/3rd records are used for training
To evaluate the performance of Naive Bayes algorithm all three datasets are discretized using equal interval binning method
For each attribute maximum and minimum value within that range is searched
For k intervals width of an interval is calculated
Using boundary values min+w, min+2w, min+(k-1)w… continuous values are grouped
Datasets are fed into the classification algorithms and results are recorded for evaluation

Fig. 9. Pictorial depiction of the workflow

III. Experimental Results

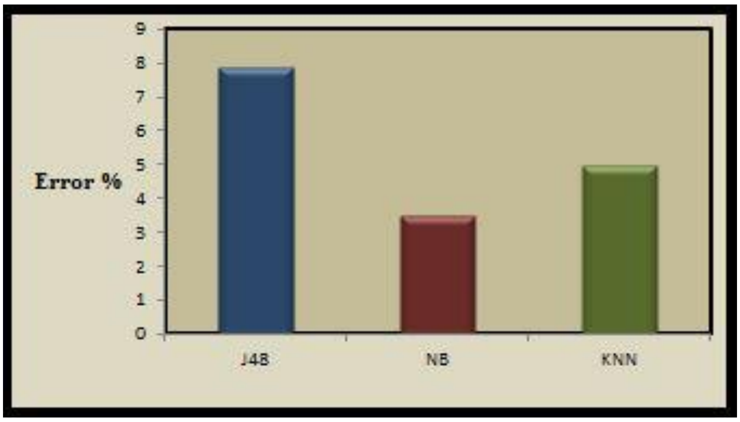


Fig. 10. Error rate of C4.5 (J48) Naïve Bayes and k-NN on Pima Indian diabetes dataset

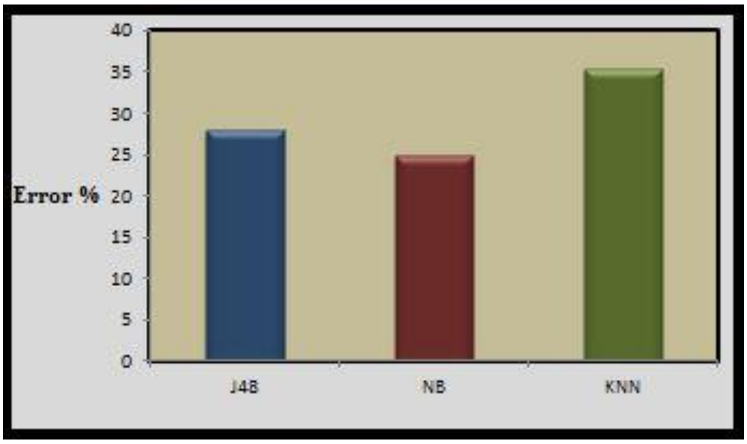


Fig. 11. Error rate of C4.5 (J48) Naïve Bayes and k-NN on Breast Cancer Dataset

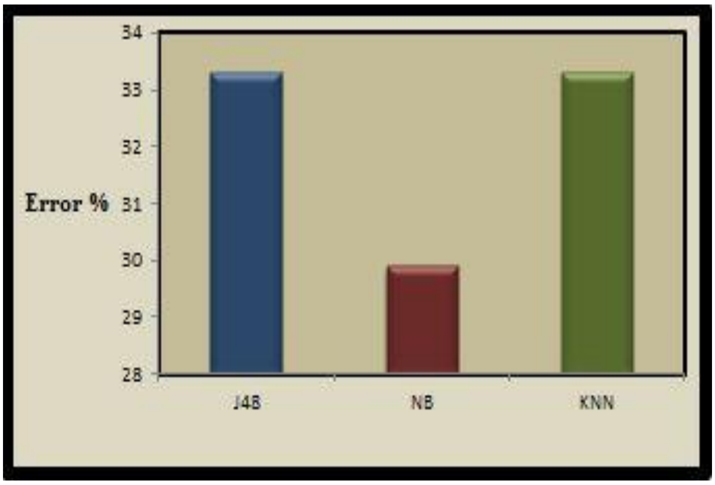


Fig. 12. Error rate of C4.5 (J48) Naïve Bayes and k-NN on ILPD Dataset

IV. Conclusion

After evaluating error rate of three classifiers for all three data sets it is clear that Naive Bayes algorithm performs well on medical domain datasets. Error rate of K-NN and j48 is almost same on PIDD and ILPD datasets, but K-NN performs well on Breast Cancer Wisconsin (Original) in comparison to J48.

References

[1]. G. K. Gupta, Introduction to Data Mining with Case Studies, Prentice Hall of India, New Delhi (2006)
[2]. P-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining. Addison Wesley Publishing (2006)
[3]. P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining. Pearson Publications (2009)
[4]. O.Maimon and L.Rokach, Data Mining and Knowledge Discovery, Springer Science and Business Media (2005)
[5]. X. Niuniu and L. Yuxun, "Review of Decision Trees," IEEE, 2010.
[6]. Dougherty, J., R. Kohavi and M. Sahami, Supervised and unsupervised discretization of continuous features, Proceeding of the 12th International Conference on Machine Learning, 12: 194-202 (1995)
[7]. A. Ashari I. Paryudi and A Min Tjoa, Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool in International Journal of Advanced Computer Science and Applications, Vol. 4, No. 11 (2013)
[8]. V. Garcia, C. Debreuve, "Fast k Nearest Neighbor Search using GPU," IEEE, 2008.
[9]. Payam Emami Khoonsari and AhmadReza Motie, A Comparison of Efficiency and Robustness of ID3 and C4.5 Algorithms Using Dynamic Test and Training Data Sets International Journal of Machine Learning and Computing, Vol. 2, No. 5, October (2012)
[10]. PIDD Dataset, https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes
[11]. https://archive.ics.uci.edu/ml/datasets/ILPD+(Indian+Liver+Patient+Dataset)
[12]. https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)

International Journal of Engineering Science Invention (IJESI) is UGC approved Journal with Sl. No. 3822, Journal no. 43302.

Susmita Nandi. "Comparison of Different Classifiers Using Medical Domain Datasets." International Journal of Engineering Science Invention(IJESI), vol. 07, no. 01, 2018, pp. 62–69.