Analysis of supervised learning Classification Algorithms By using Weka 3.8

Name:Debobrata Chakraborty

ID:18-36103-1

Course: Data warehousing and Data Mining

Section: B

Dept: Computer Engineering(CoE)

Email:

debobratachakraborty80@gmail.com

***Introduction*—** Heart disease is one of the most deadly and chronic illnesses, increasing the risk of heart failure and heart attack. The aim of this paper is to use supervised classification algorithms to classify heart disease. To detect heart disease, this experiment employs three machine learning classification algorithms: Naive Bayes, KNN, and Decision Tree.

Experiments are performed on the UCI machine learning repository's Heart Disease UCI dataset. Precision, Accuracy, F-Measure, and Recall are all used to test the performance of the three algorithms. The accuracy of a classification system is determined by the number of instances that are correctly classified and those that are incorrectly classified. According to the results, KNN outperforms with the highest precision of 70 percent. Those algorithms are also performed in a test dataset.

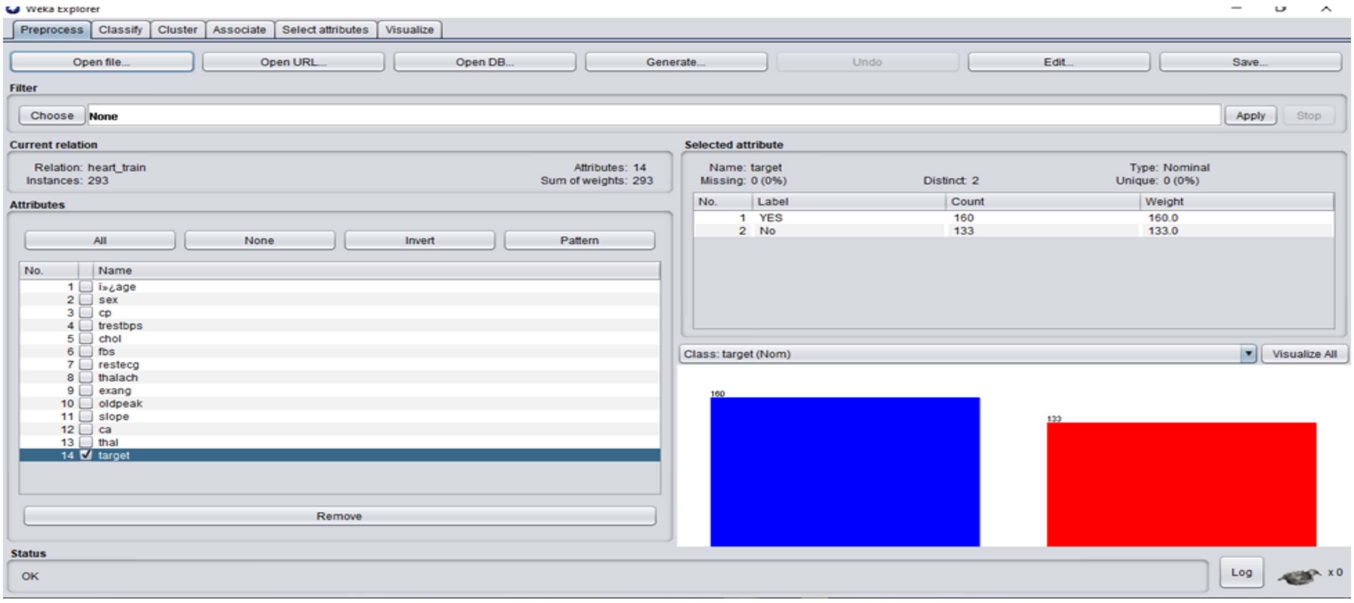
Naive Bayes: The Bayes Theorem is the basis for the Nave Bayes algorithm, which is used in a wide range of classification tasks.

KNN: Nearest Neighbour classification calculates the classification of an unseen instance using the classification of the instances "closest" to it, and Nave Bayes uses probability theory to find the most probable of the possible classifications. Since it does not learn from the training set right away, the K-Nearest Neighbor lazy learner algorithm stores the dataset and performs an operation on it at the time of classification.

Decision Tree: The Decision Tree is a supervised machine learning algorithm that solves classification problems using an intermediate tree-like structure. A decision tree is a diagram that depicts the various outcomes of a set of similar choices. It enables a person or organization to compare and contrast various options based on their costs, probabilities, and benefits. They can be used to ignite informal discussion or to create an algorithm that mathematically predicts the best option.

I. Result

At first, the heart disease dataset was loaded in Weka.



Filter: None

Current relation

Relation: heart_train
Instances: 293
Attributes: 14
Sum of weights: 293

Attributes

No.		Name
1	□	ï»¿age
2	□	sex
3	□	cp
4	□	trestbps
5	□	chol
6	□	fbs
7	□	restecg
8	□	thalach
9	□	exang
10	□	oldpeak
11	□	slope
12	□	ca
13	□	thal
14	☑	target

Selected attribute

Name: target
Missing: 0 (0%)
Distinct: 2
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	YES	160	160.0
2	No	133	133.0

Status

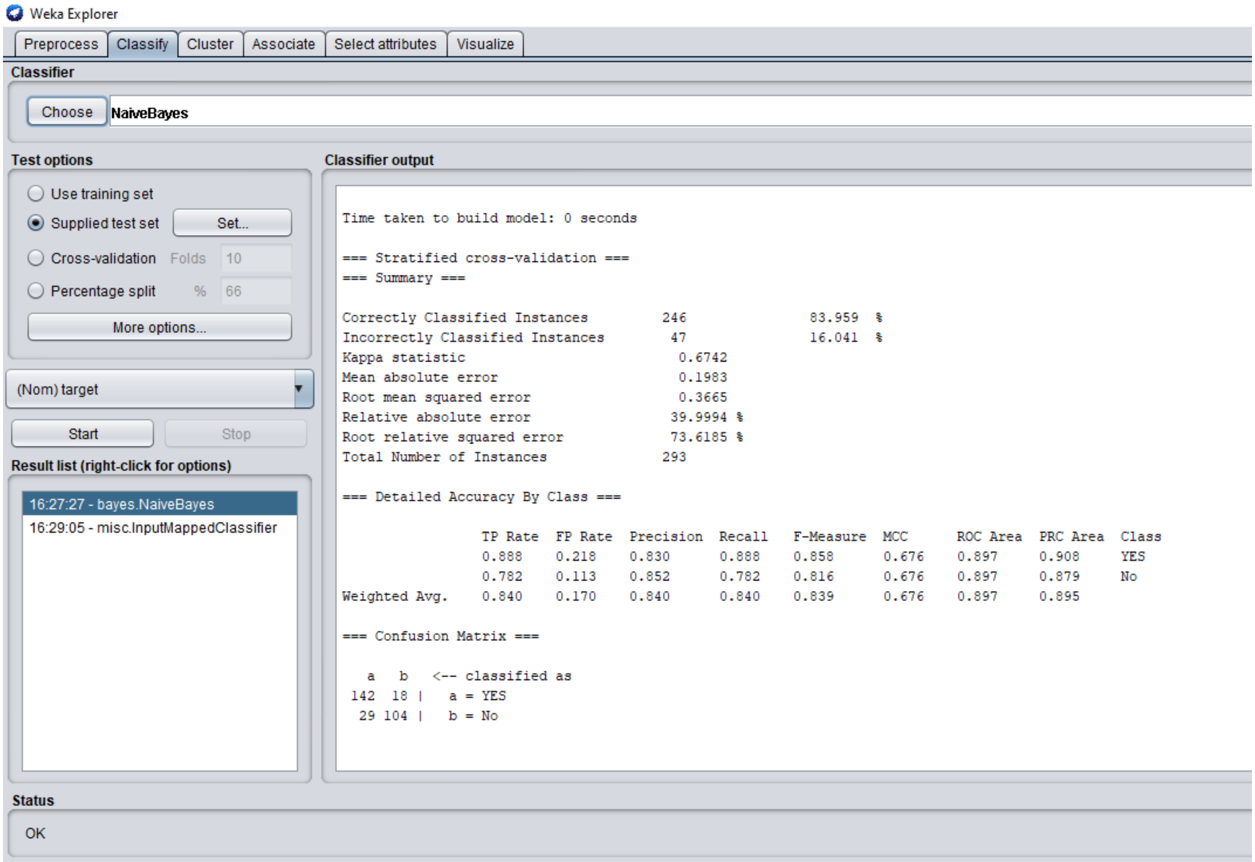
OK

Figure 1: Preprocess the data

There are 293 instances in the dataset. Target indicates 2 classes. Those who have heart disease are labeled as 'Yes,' and those who do not have heart disease are labeled as 'No.' There are 14 attributes in a dataset. The classify algorithm was introduced after the data was pre-processed. Test dataset contain 10 instance.

For each classification algorithm, a 10-fold cross validation set was used.

Naive Bayes: After applying naïve bayes algorithm on the training dataset it classified 246 correctly and 47 incorrectly.



Classifier: NaiveBayes

● Supplied test set

(Nom) target

16:27:27 - bayes.NaiveBayes
16:29:05 - misc.InputMappedClassifier

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         246               83.959  %
Incorrectly Classified Instances        47               16.041  %
Kappa statistic                          0.6742
Mean absolute error                      0.1983
Root mean squared error                  0.3665
Relative absolute error                 39.9994 %
Root relative squared error             73.6185 %
Total Number of Instances              293

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.888    0.218    0.830      0.888    0.858      0.676    0.897     0.908     YES
                 0.782    0.113    0.852      0.782    0.816      0.676    0.897     0.879     No
Weighted Avg.    0.840    0.170    0.840      0.840    0.839      0.676    0.897     0.895

=== Confusion Matrix ===

   a   b   <-- classified as
 142  18 |   a = YES
  29 104 |   b = No
```

Figure 2: Naïve Bayes classification (training set)

Confusion matrix: A confusion matrix is a method of summarizing a classification algorithm's results.

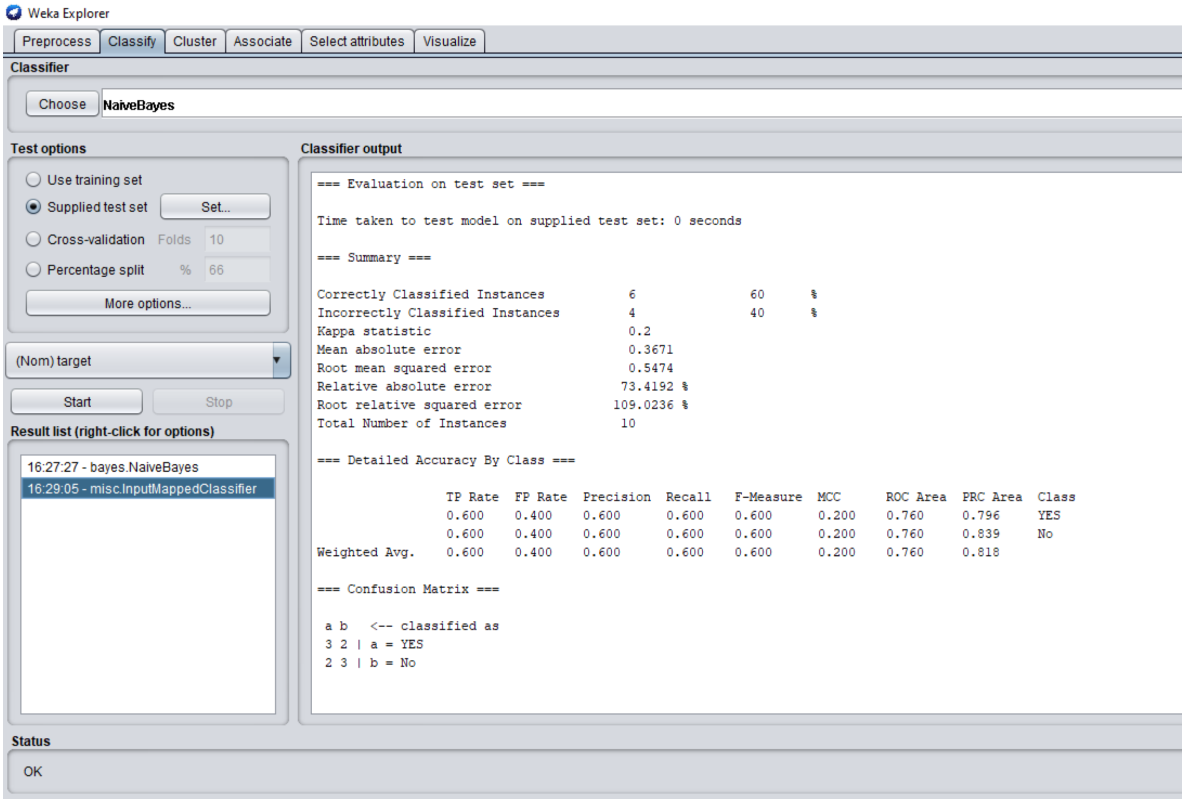
Table 1:Confusion matrix of Naïve Bayes(training set):

a	b	< classified as
142	18	a=Yes
29	104	b=No

Table 2:Counfusion matrix of Naïve Bayes (test set)

a	b	< classified as
3	2	a=Yes
2	3	b=No

Naïve bayes correctly classified 6 instances from the test set.



Classifier: NaiveBayes

Test options: ○ Use training set; ● Supplied test set (Set...); ○ Cross-validation Folds 10; ○ Percentage split % 66

(Nom) target

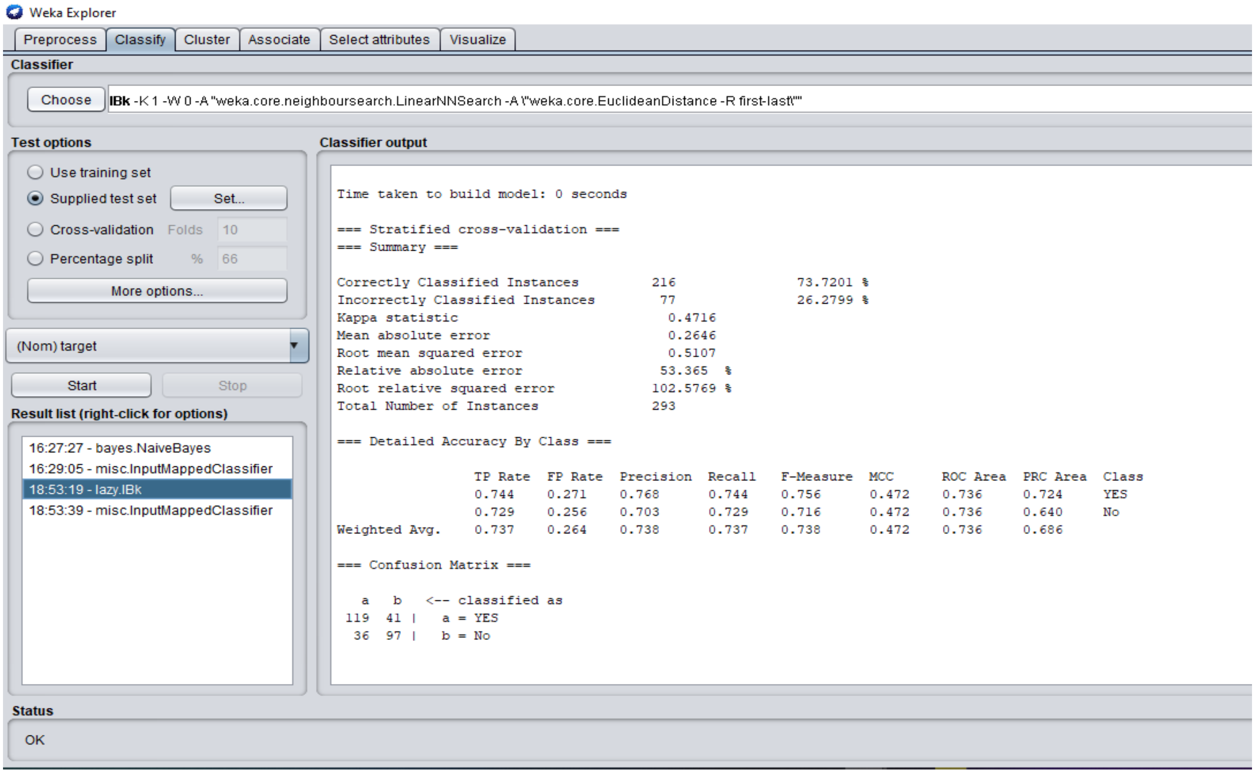
Result list (right-click for options):
16:27:27 - bayes.NaiveBayes
16:29:05 - misc.InputMappedClassifier

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances           6               60      %
Incorrectly Classified Instances         4               40      %
Kappa statistic                          0.2
Mean absolute error                      0.3671
Root mean squared error                  0.5474
Relative absolute error                 73.4192 %
Root relative squared error            109.0236 %
Total Number of Instances               10

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.600    0.400    0.600      0.600    0.600      0.200    0.760     0.796     YES
                 0.600    0.400    0.600      0.600    0.600      0.200    0.760     0.839     No
Weighted Avg.    0.600    0.400    0.600      0.600    0.600      0.200    0.760     0.818

=== Confusion Matrix ===

 a b   <-- classified as
 3 2 | a = YES
 2 3 | b = No
```

Status: OK

Figure 3: Naïve Bayes classification (test set)

KNN Algorithm: To apply the knn algorithm, Lazy IBK was chosen from the classify option in weka.



Classifier: IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

Test options: ○ Use training set; ● Supplied test set (Set...); ○ Cross-validation Folds 10; ○ Percentage split % 66

(Nom) target

Result list (right-click for options):
16:27:27 - bayes.NaiveBayes
16:29:05 - misc.InputMappedClassifier
18:53:19 - lazy.IBk
18:53:39 - misc.InputMappedClassifier

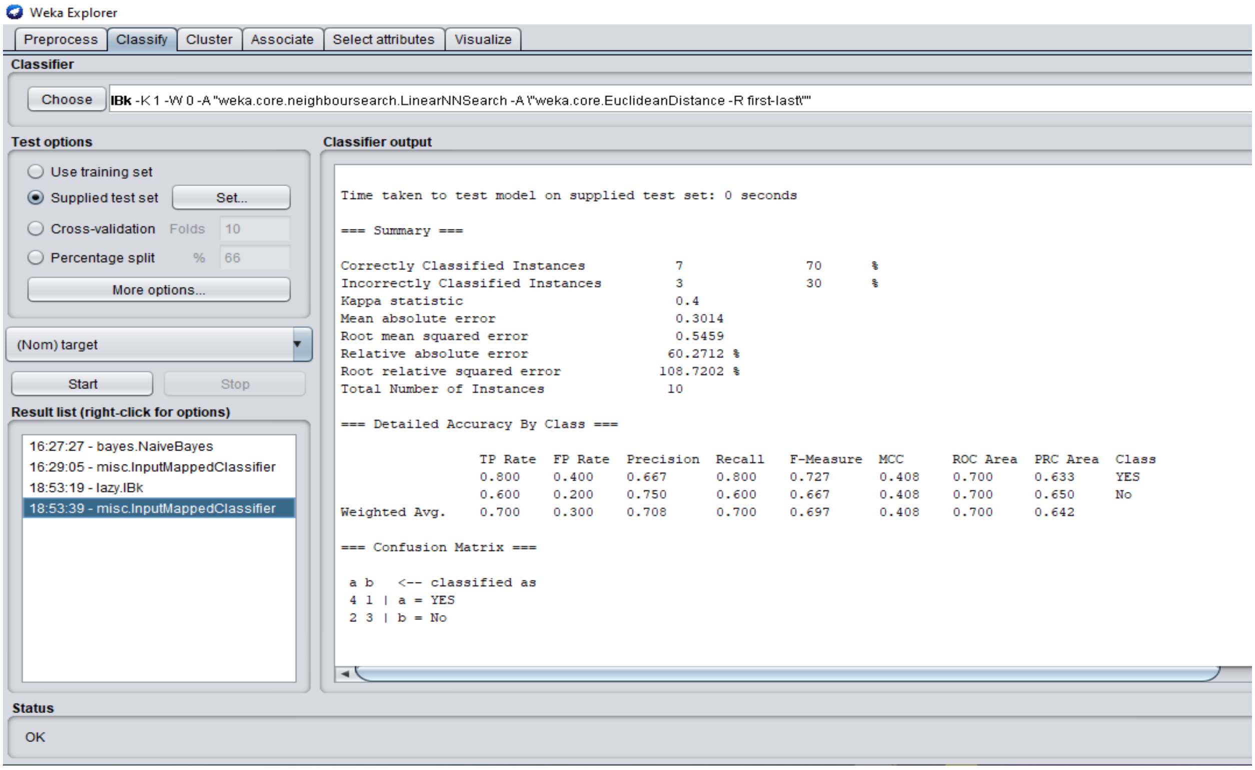
```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         216               73.7201 %
Incorrectly Classified Instances        77               26.2799 %
Kappa statistic                          0.4716
Mean absolute error                      0.2646
Root mean squared error                  0.5107
Relative absolute error                 53.365  %
Root relative squared error            102.5769 %
Total Number of Instances              293

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.744    0.271    0.768      0.744    0.756      0.472    0.736     0.724     YES
                 0.729    0.256    0.703      0.729    0.716      0.472    0.736     0.640     No
Weighted Avg.    0.737    0.264    0.738      0.737    0.738      0.472    0.736     0.686

=== Confusion Matrix ===

   a   b   <-- classified as
 119  41 |   a = YES
  36  97 |   b = No
```

Status: OK

Figure 4: KNN classification (training set)

Table 3:Confusion matrix of KNN(training set):

a	b	< classified as
119	41	a=Yes
36	97	b=No



IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

○ Use training set
● Supplied test set Set...
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) target

16:27:27 - bayes.NaiveBayes
16:29:05 - misc.InputMappedClassifier
18:53:19 - lazy.IBk
18:53:39 - misc.InputMappedClassifier

```
Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances           7               70      %
Incorrectly Classified Instances         3               30      %
Kappa statistic                          0.4
Mean absolute error                      0.3014
Root mean squared error                  0.5459
Relative absolute error                 60.2712 %
Root relative squared error            108.7202 %
Total Number of Instances               10

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.800    0.400    0.667      0.800    0.727      0.408    0.700     0.633     YES
                 0.600    0.200    0.750      0.600    0.667      0.408    0.700     0.650     No
Weighted Avg.    0.700    0.300    0.708      0.700    0.697      0.408    0.700     0.642

=== Confusion Matrix ===

 a b   <-- classified as
 4 1 | a = YES
 2 3 | b = No
```

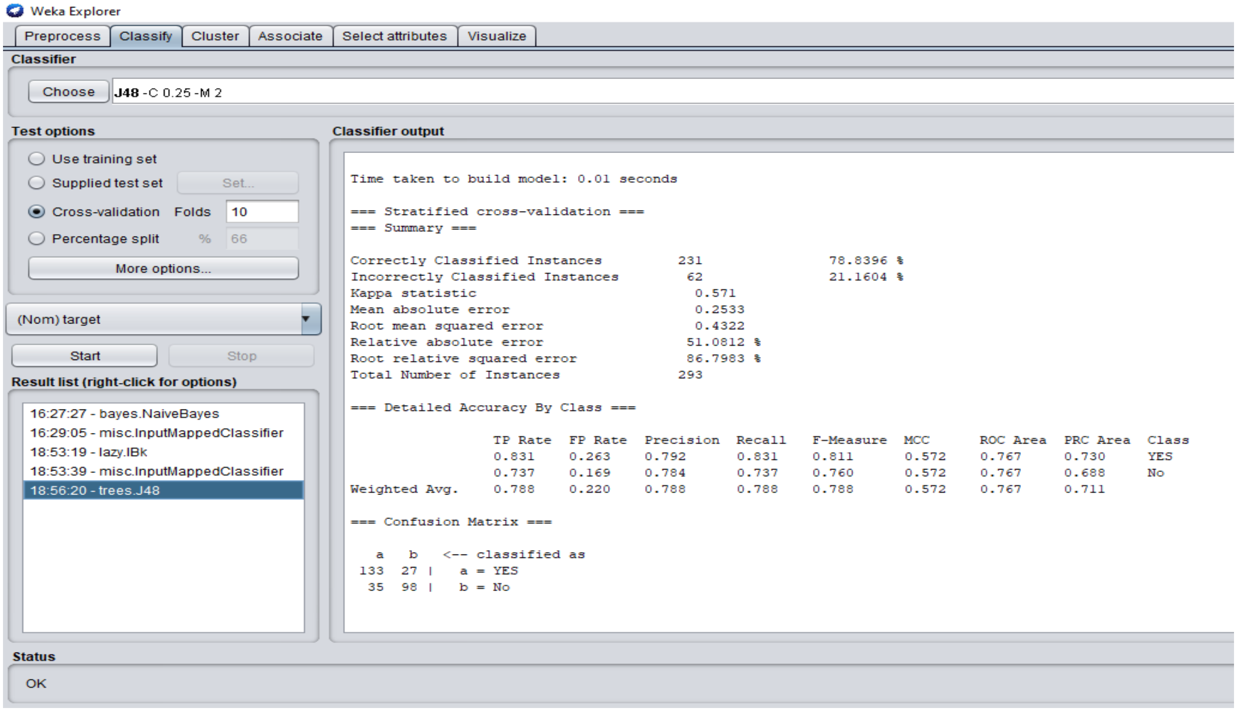
Figure 4: KNN classification (test set)

Table 4:Confusion matrix of KNN(Test set):

a	b	< classified as
4	1	a=Yes
2	3	b=No

KNN correctly classified 7 instances from the test set.

Decision Tree: Tree-J48 algorithm was implemented on the dataset.



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set Set...
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) target

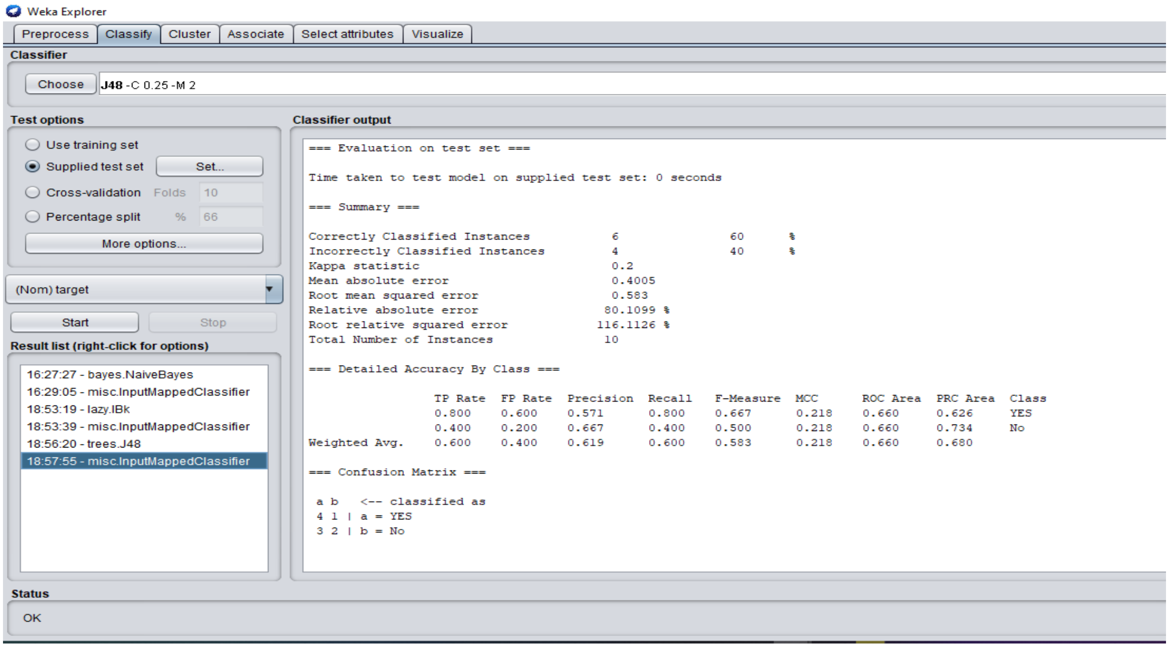
16:27:27 - bayes.NaiveBayes
16:29:05 - misc.InputMappedClassifier
18:53:19 - lazy.IBk
18:53:39 - misc.InputMappedClassifier
18:56:20 - trees.J48

```
Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         231               78.8396 %
Incorrectly Classified Instances        62               21.1604 %
Kappa statistic                          0.571
Mean absolute error                      0.2533
Root mean squared error                  0.4322
Relative absolute error                 51.0812 %
Root relative squared error             86.7983 %
Total Number of Instances              293

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.831    0.263    0.792      0.831    0.811      0.572    0.767     0.730     YES
                 0.737    0.169    0.784      0.737    0.760      0.572    0.767     0.688     No
Weighted Avg.    0.788    0.220    0.788      0.788    0.788      0.572    0.767     0.711

=== Confusion Matrix ===

   a   b   <-- classified as
 133  27 |   a = YES
  35  98 |   b = No
```

Figure 5: Decision Tree classification (training set)

Table 5:Confusion matrix of Decision Tree(training set):

a	b	< classified as
133	27	a=Yes
35	98	b=No



Classifier: J48 -C 0.25 -M 2

Test options

○ Use training set
● Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

(Nom) target

Result list (right-click for options)

16:27:27 - bayes.NaiveBayes
16:29:05 - misc.InputMappedClassifier
18:53:19 - lazy.IBk
18:53:39 - misc.InputMappedClassifier
18:56:20 - trees.J48
18:57:55 - misc.InputMappedClassifier

Classifier output

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances           6               60      %
Incorrectly Classified Instances         4               40      %
Kappa statistic                          0.2
Mean absolute error                      0.4005
Root mean squared error                  0.583
Relative absolute error                 80.1099 %
Root relative squared error            116.1126 %
Total Number of Instances               10

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.800    0.600    0.571      0.800    0.667      0.218    0.660     0.626     YES
                 0.400    0.200    0.667      0.400    0.500      0.218    0.660     0.734     No
Weighted Avg.    0.600    0.400    0.619      0.600    0.583      0.218    0.660     0.680

=== Confusion Matrix ===

 a b   <-- classified as
 4 1 | a = YES
 3 2 | b = No
```

Status

OK

Figure 6: Decision Tree classification (test set)

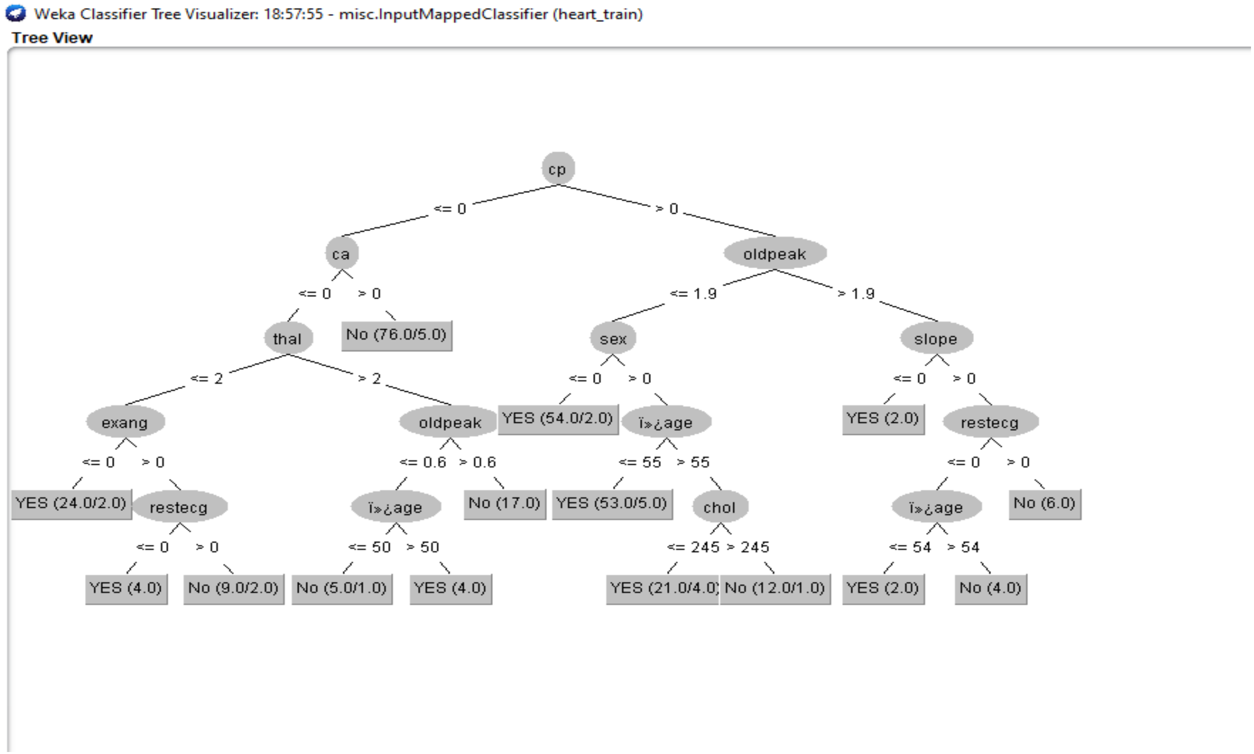


Figure 7: Tree

A decision tree usually begins with a single node and branches out into different outcomes. Each of those outcomes leads to new nodes, each of which leads to new possibilities. It takes on a tree-like form as a result of this.

Table 6:Confusion matrix of Decision Tree(training set):

a	b	< classified as
4	1	a=Yes
3	2	b=No

Table 7:Accuracy Measure

Measures	**Definitions**	**Formula**
1. Accuracy (A)	Accuracy determines the accuracy of the algorithm in predicting instances.	A=(TP+TN) / (Total no of samples)
2. Precision (P)	Classifiers correctness/accuracy is measured by Precision.	P = TP / (TP+ FP)
3. Recall (R)	To measure the classifiers completeness or sensitivity, Recall is used.	R =TP / (TP+FN)
4. F-Measure	F-Measure is the weighted average of precision and recall.	F=2*(P*R) / (P+R)
5. ROC	ROC(Receiver Operating Curve) curves are used to compare the usefulness of tests.	

Test dataset was considered to find out the high accuracy classification.

Table 8: Comparative Performance of Classification Algorithms on Various Measures.

Algorithm	Precision	Recall	F-Measure	ROC	Accuracy(%)
Naïve Bayes	0.600	0.600	0.600	0.760	60
KNN	0.708	0.700	0.697	0.700	70
Decision Tree	0.619	0.600	0.583	0.660	60

Table-7 shows the performance of corresponding classifiers in terms of Accuracy, Precision, F-measure, Recall, and ROC values, while Table-8 shows the performance of classifiers based on classified instances. TP stands for True Positive, TN for True Negative, FP for False Positive, and FN for False Negative. The output of the corresponding classifiers is measured in terms of accuracy, precision, and recall.

After applying the model on test dataset the below data was found.

Table 9. Classifier's Performance on The Basis of Classified Instances

Total Instance	Algorithm	Correctly Classified Instance	Incorrectly Classified Instances
10	Naïve Bayes	6	4
	KNN	7	3
	Decision Tree	6	4

As a result, the KNN algorithm is the best supervised machine learning approach in this experiment because it has a higher accuracy than other classification algorithms, with a 70.00 percent accuracy.

II. Discussion

Three supervised machine learning algorithms were used in this experiment. After calculating all of the values, the KNN value has a maximum accuracy of 70%. On the evaluation dataset, it performs admirably. This machine learning approach was applied to a dataset of heart diseases. This algorithm can also be used on other datasets to build a classifier model that can predict various diseases.

III. Reference

[1] https://www.kaggle.com/ronitf/heart-disease-uci

Analysis of unsupervised learning Clustering Algorithm By using Weka 3.8

Name:Debobrata Chakraborty

ID:18-36103-1

Course: Data warehousing and Data Mining

Section: B

Dept: Computer Engineering(CoE)

Email:

debobratachakraborty80@gmail.com

Introduction- Diabetes is one of the most deadly and chronic diseases that causes blood sugar levels to rise. The aim of this paper is to use unsupervised learning algorithms to clustering diabetes disease. To detect diabetes, this experiment employs hierarchical cluster.

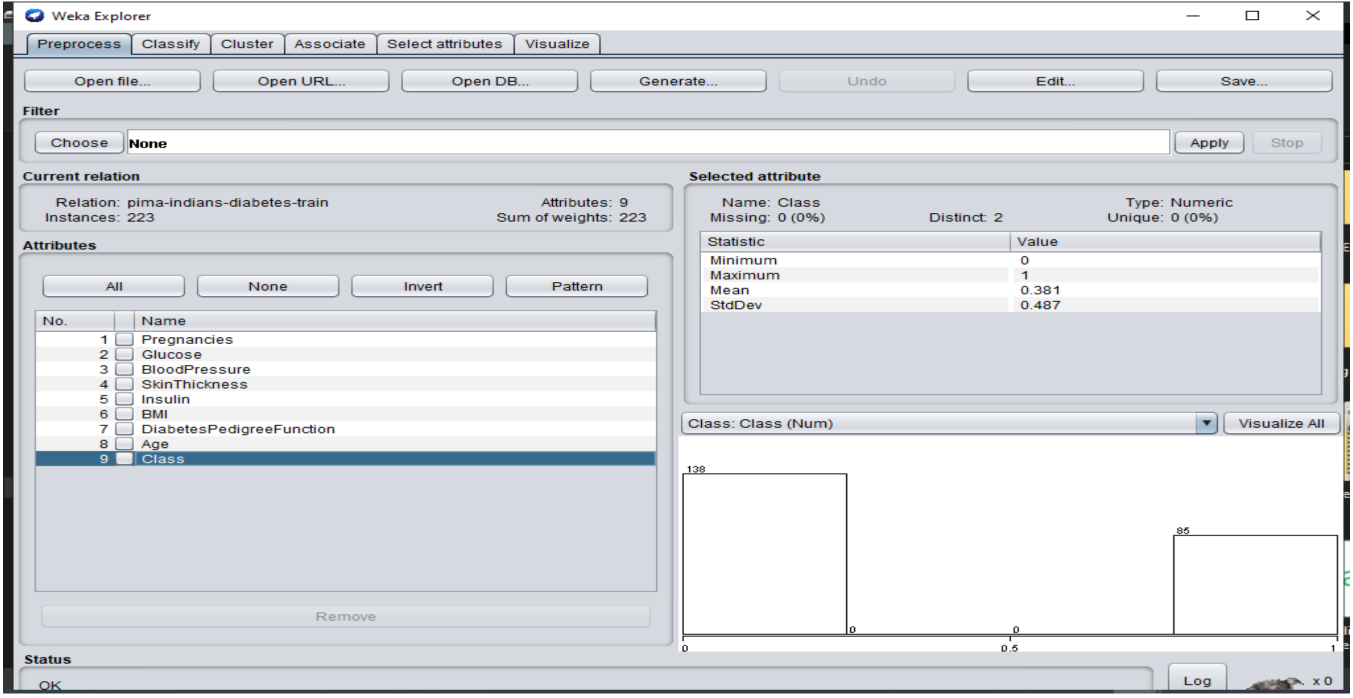
Experiments are performed on the Prima Indian Diabetes dataset. Clustering is a descriptive model that divides a set of objects into groups based on their relationships.

Clustering is a method that is used in a variety of areas.Image analysis, pattern recognition, statistical data analysis, and so on are only a few examples. Clustering is the division of data into groups of items that are identical. Clustering is the division of data into groups of items that are identical. Each cluster is made up of a variety of objects that are identical to one another but not to objects from other classes. To shape clusters, various clustering algorithms are available. To compare various clustering algorithms, the WEKA tool is used.

In this paper, hierarchical cluster is used, an algorithm that groups related objects into clusters, also known as hierarchical cluster analysis. The endpoint is a set of clusters, each of which is distinct from the others while the artifacts within each cluster are broadly identical.

I. Result

At first, the diabetes test dataset was loaded in Weka. There are 223 instances and 9 attribute on the diabetes dataset.



Filter

None

Current relation

Relation: pima-indians-diabetes-train
Attributes: 9
Instances: 223
Sum of weights: 223

Attributes

No.	Name
1	Pregnancies
2	Glucose
3	BloodPressure
4	SkinThickness
5	Insulin
6	BMI
7	DiabetesPedigreeFunction
8	Age
9	Class

Selected attribute

Name: Class
Type: Numeric
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)

Statistic	Value
Minimum	0
Maximum	1
Mean	0.381
StdDev	0.487

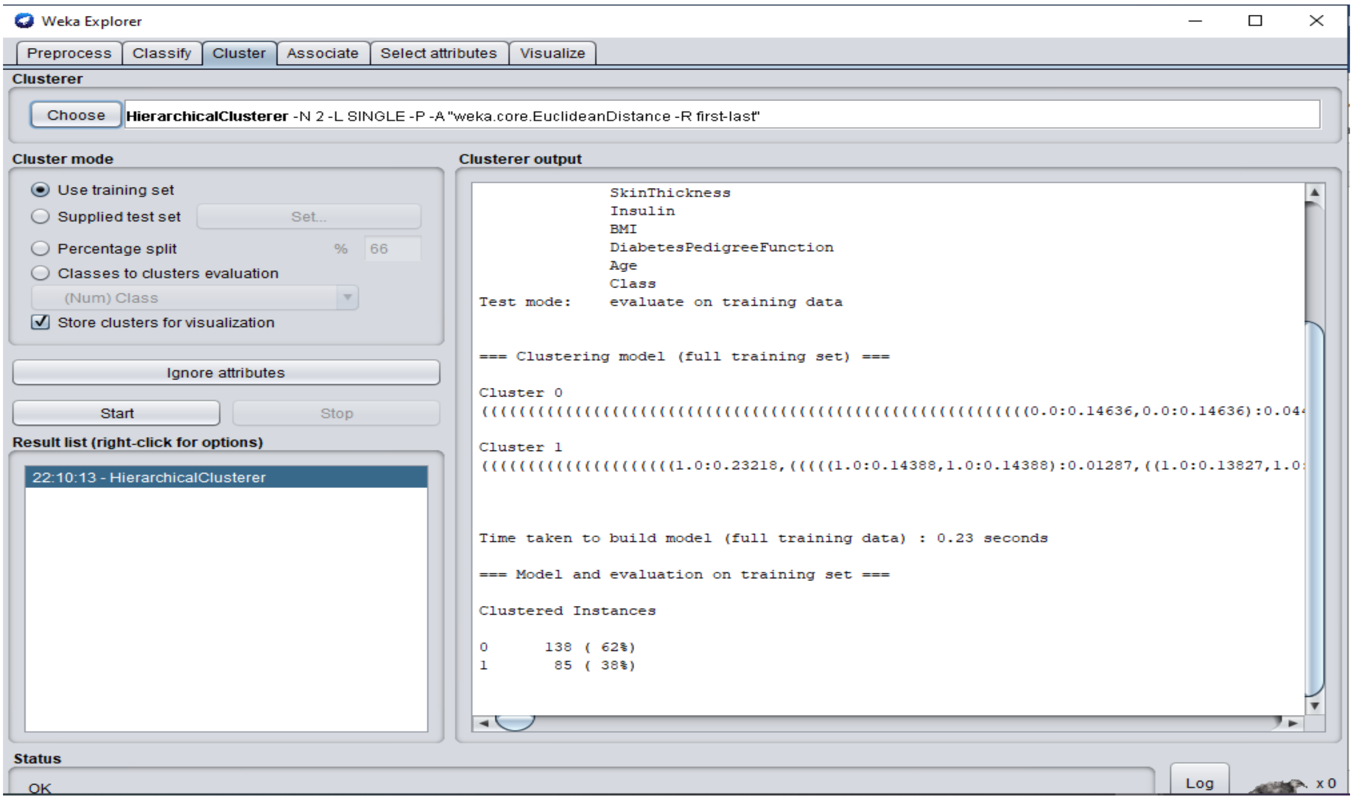
Class: Class (Num)

Status

OK

Figure 1: Preprocess the data

After that, the cluster option was selected and heirarchical clustering algorithm was implemented.



HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Num) Class
☑ Store clusters for visualization

22:10:13 - HierarchicalClusterer

```
                 SkinThickness
                 Insulin
                 BMI
                 DiabetesPedigreeFunction
                 Age
                 Class
Test mode:    evaluate on training data


=== Clustering model (full training set) ===

Cluster 0
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((0.0:0.14636,0.0:0.14636):0.04

Cluster 1
((((((((((((((((((((1.0:0.23218,(((((1.0:0.14388,1.0:0.14388):0.01287,((1.0:0.13827,1.0:



Time taken to build model (full training data) : 0.23 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      138 ( 62%)
1       85 ( 38%)
```

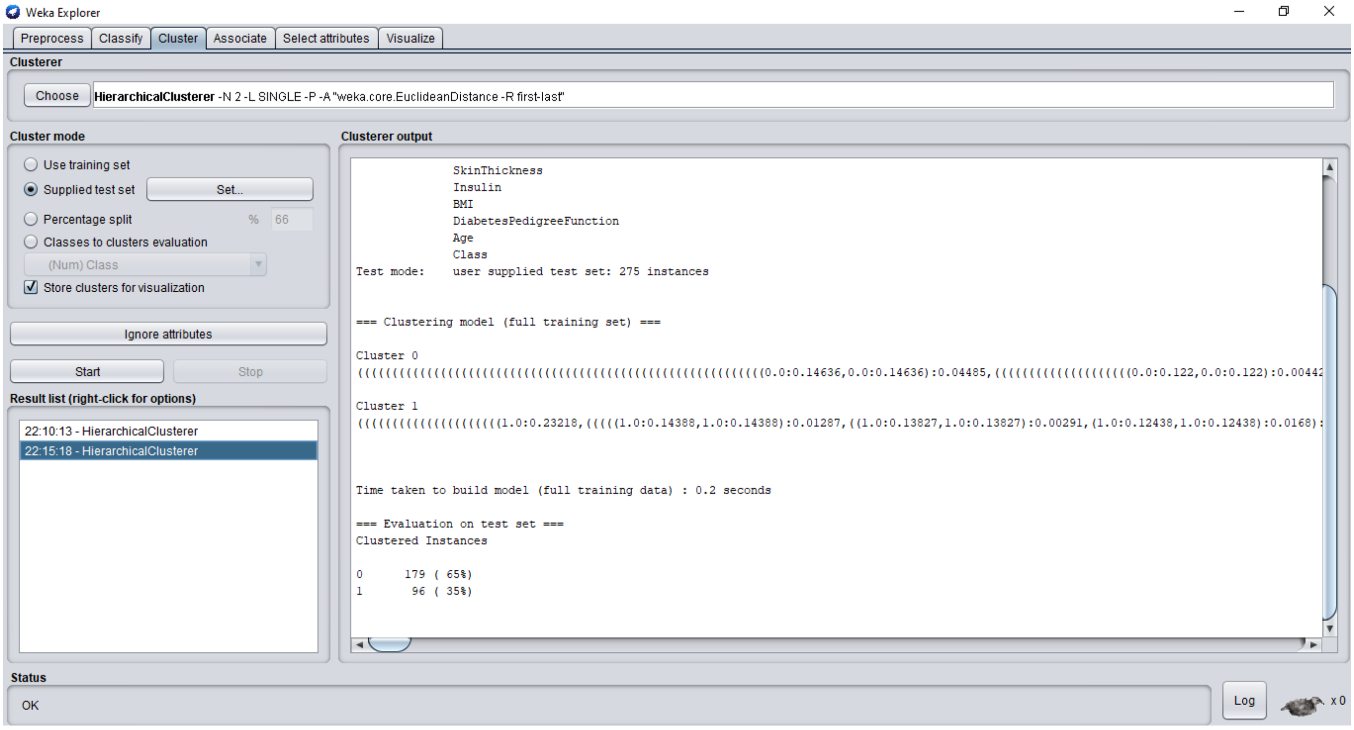
OK

Figure 2: Clustering(training dataset)

After the clustering, the training set clustering into two part.

0	138 instance	62%
1	85 instance	38%

138 instance clustered into 0 category and 85 instances clustered into 1 category.



HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

○ Use training set
● Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Num) Class
☑ Store clusters for visualization

22:10:13 - HierarchicalClusterer
22:15:18 - HierarchicalClusterer

```
              SkinThickness
              Insulin
              BMI
              DiabetesPedigreeFunction
              Age
              Class
Test mode:    user supplied test set: 275 instances


=== Clustering model (full training set) ===

Cluster 0
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((0.0:0.14636,0.0:0.14636):0.04485,(((((((((((((((((((0.0:0.122,0.0:0.122):0.00442

Cluster 1
((((((((((((((((((((1.0:0.23218,(((((1.0:0.14388,1.0:0.14388):0.01287,((1.0:0.13827,1.0:0.13827):0.00291,(1.0:0.12438,1.0:0.12438):0.0168):



Time taken to build model (full training data) : 0.2 seconds

=== Evaluation on test set ===
Clustered Instances

0      179 ( 65%)
1       96 ( 35%)
```

OK

Figure 3: Clustering (test set)

By supplying the test set the model clustered the data. 179 instances(65%) are clustered into category 0 and 96 instances(35%) are clustered into category 1.

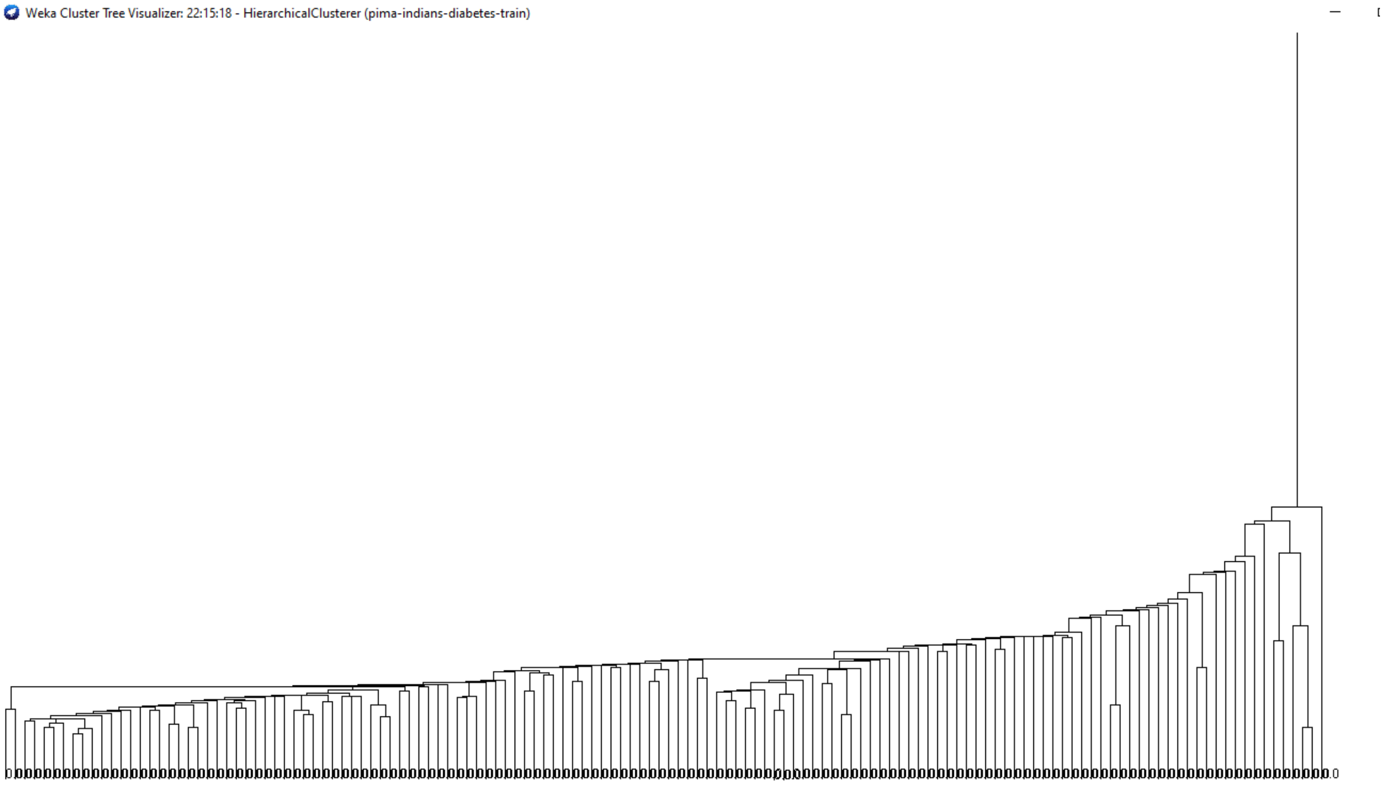

Weka Cluster Tree Visualizer: 22:15:18 - HierarchicalClusterer (pima-indians-diabetes-train)

Figure 4: Dendrogram

This is the dendrogram of the hierarchical cluster. A dendrogram is a type of tree diagram that is commonly used to show how hierarchical clustering produces clusters. In computational biology, dendrograms are often used to depict the clustering of genes or samples.

The following table represents the analysis process of the clustering on test set.

Algorithm	No. of cluster	Cluster instance	Time taken to build
hierarchical cluster	2	. 179 (65%) 96(35%)	0.2s

II. Discussion

Data mining is a computer science and information technology discipline. There is a vast amount of data scattered across the globe, much of which is in the form of raw data, which often includes relevant information. A mining method is used to obtain relevant data. A cluster is formed from a large amount of data by objects of a similar nature. Hierarchical cluster was performed in this paper. The algorithm generates clusters based on related objects and the amount of time it takes to construct them

III. Reference

[1]https://www.kaggle.com/kumargh/pimaindiansdiabetescsv