DATA WAREHOSING & DATA MINING

SEMESTER: Spring 2021-2022

FINAL-TERM ASSIGNMENT

DATA MINING WITH WEKA

SUBMITTED BY:

STUDENT NAME	STUDENT ID
KHANAM, FAHMIDA	19-40351-1

SUBMITTED TO:

COURSE TEACHER: TOHEDUL ISLAM

INTRODUCTION:

Data mining is the process of uncovering patterns and other valuable information from large data sets. It is also known as knowledge discovery in data (KDD). Data mining is used in many areas of research and business, including healthcare, education, sales and marketing, product development, etc. It is a computer science and statistics multidisciplinary topic with the general purpose of extracting information from data collection and transforming the information into an accessible structure for subsequent use. KNN, Naive Bayes, and Decision Tree are some of the classification algorithms used in data mining. I have chosen the "Heart-Diseases dataset" to classify the type by using two different classifiers and find the best-suited classifier for the dataset. I have chosen Naive Bayes and Decision Tree to classify the dataset.

Information about the dataset: In this report, the used "Heart-failure dataset", a CSV dataset file [Later converted into .arff], collected from Kaggle.com was used to predict the outcome of the heart failure that might be accurate for the patient according to their health condition.

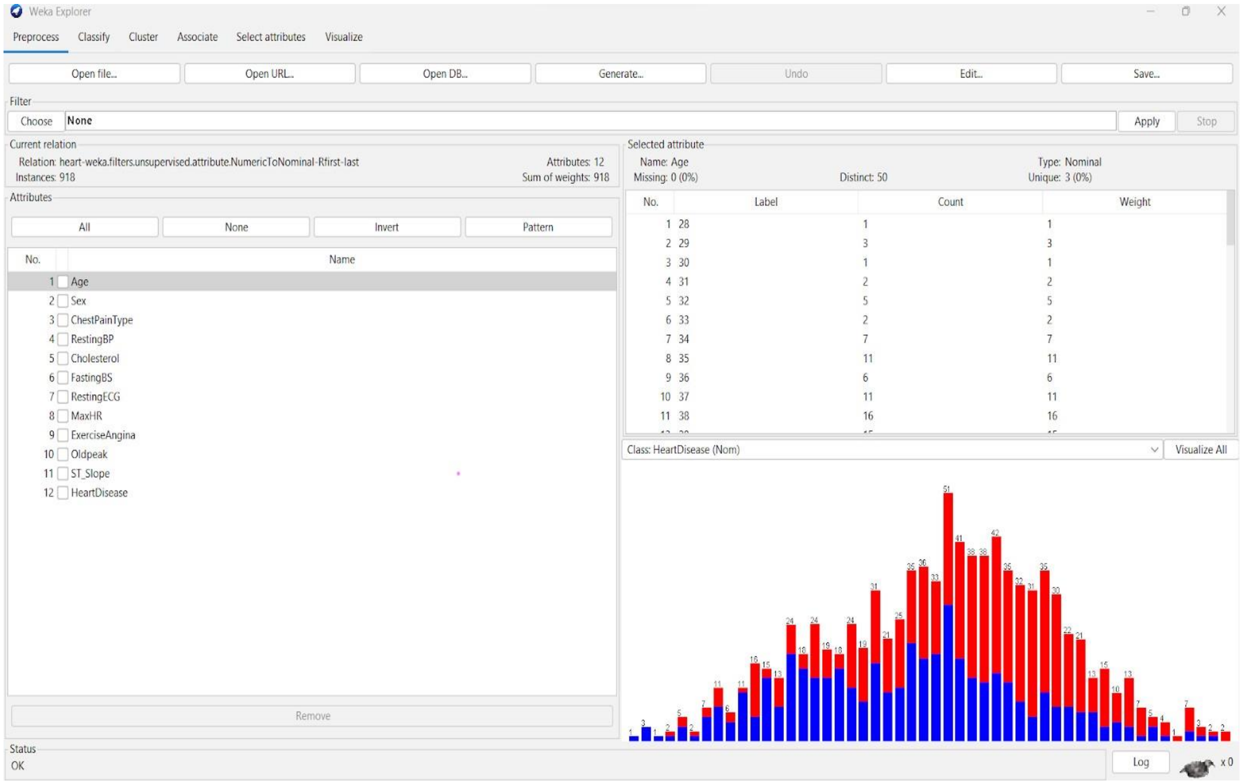
The targeted feature is:

- HeartDisease

The other feature sets are:

- Age
- Sex
- ChestPainType
- RestingBP
- Cholesterol
- FastingBS
- RestingECG
- MaxHR
- ExerciseAngina
- Oldpeak
- ST_Slope

There is a total of 918 instances of these 12 attributes and all these instances were used for classification. Here are the graphical details of the attributes:



Filter: None

Current relation
Relation: heart-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Attributes: 12
Instances: 918
Sum of weights: 918

Attributes

No.		Name
1	□	Age
2	□	Sex
3	□	ChestPainType
4	□	RestingBP
5	□	Cholesterol
6	□	FastingBS
7	□	RestingECG
8	□	MaxHR
9	□	ExerciseAngina
10	□	Oldpeak
11	□	ST_Slope
12	□	HeartDisease

Selected attribute
Name: Age
Type: Nominal
Missing: 0 (0%)
Distinct: 50
Unique: 3 (0%)

No.	Label	Count	Weight
1	28	1	1
2	29	3	3
3	30	1	1
4	31	2	2
5	32	5	5
6	33	2	2
7	34	7	7
8	35	11	11
9	36	6	6
10	37	11	11
11	38	16	16

Class: HeartDisease (Nom)

Status
OK

Figure 1: Selected dataset

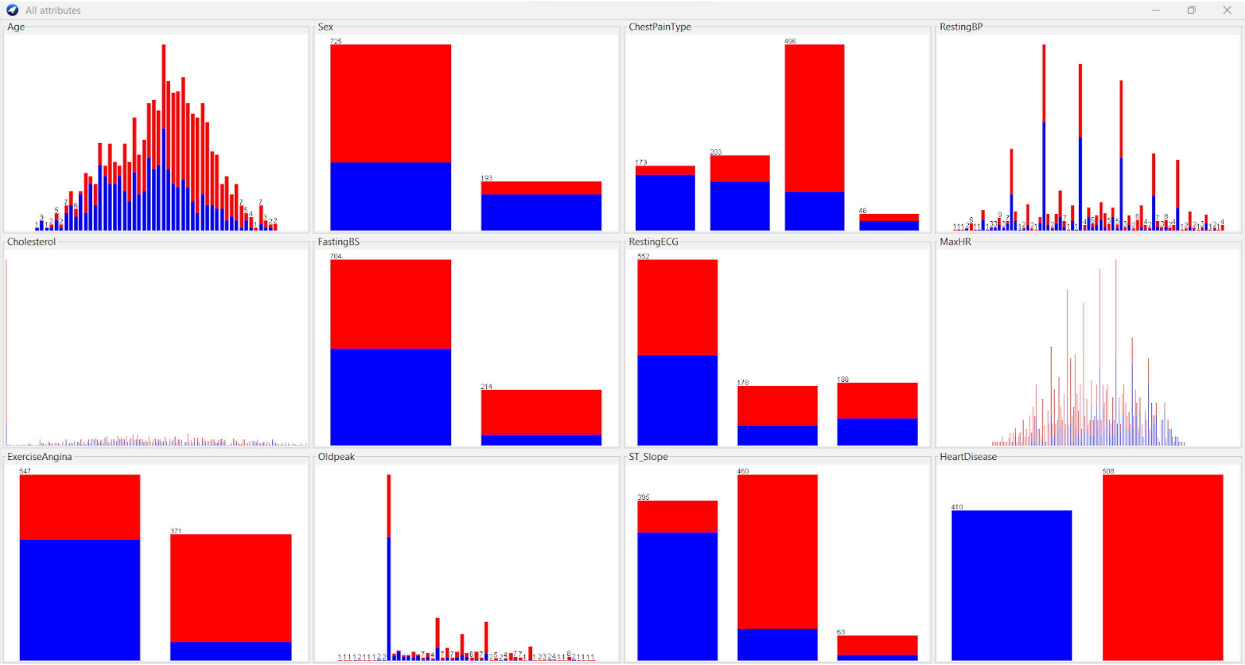


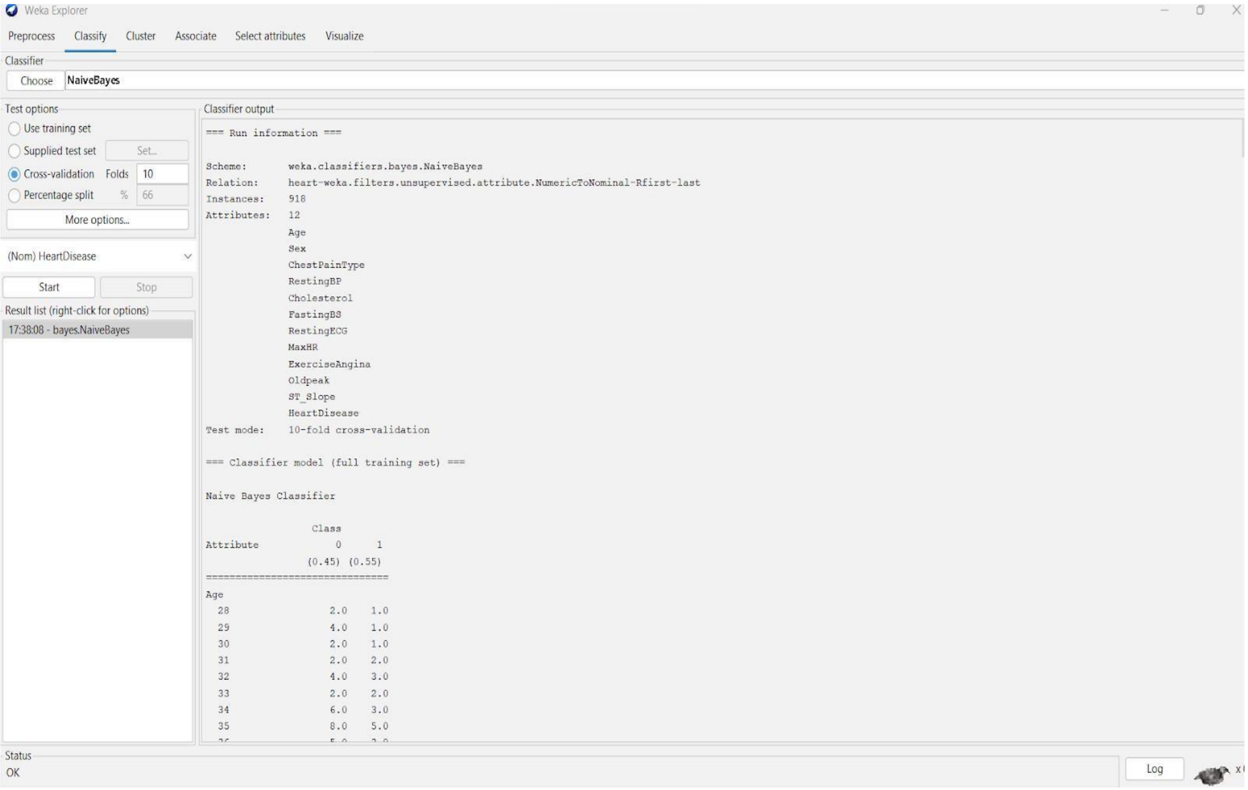
Figure 2: Details of all attribute

Classifier: A classifier is a machine learning model that is used to discriminate different objects based on certain features. Two kinds of classification have been used with the same data to compare the result. In this process, Naïve Bayes, and Decision Tree Classifiers were used.

RESULT OF THE CLASSIFIERS:

Weka 3.8.6 version software was used to construct the classifier. Weka is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Applying naïve Bayes classifier: Naïve Bayes algorithm is a supervised learning algorithm, which is based on the Bayes theorem and is used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. While classifying the selected dataset, the NaiveBayes format was selected from the Bayes folder.



Classifier: NaiveBayes

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

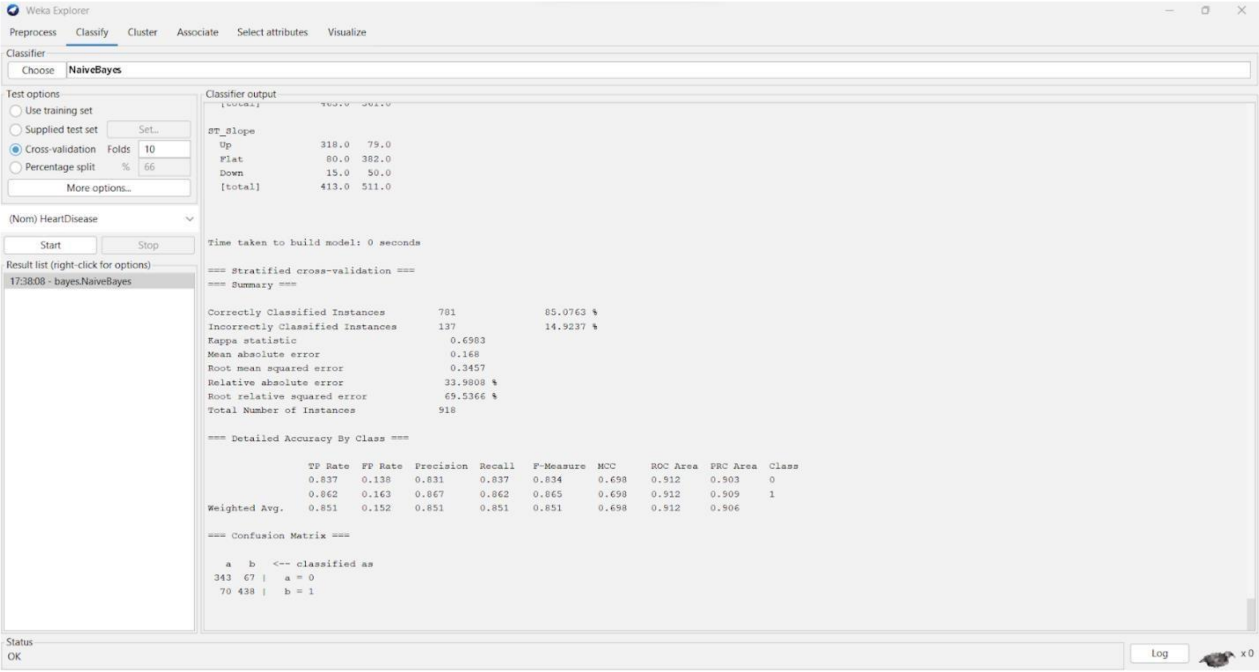
(Nom) HeartDisease

Result list (right-click for options): 17:38:08 - bayes.NaiveBayes

Status: OK

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     heart-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    918
Attributes:   12
              Age
              Sex
              ChestPainType
              RestingBP
              Cholesterol
              FastingBS
              RestingECG
              MaxHR
              ExerciseAngina
              Oldpeak
              ST_Slope
              HeartDisease
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                 Class
Attribute            0      1
                 (0.45) (0.55)
===============================
Age
  28               2.0    1.0
  29               4.0    1.0
  30               2.0    1.0
  31               2.0    2.0
  32               4.0    3.0
  33               2.0    2.0
  34               6.0    3.0
  35               8.0    5.0
```



Classifier: NaiveBayes

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

(Nom) HeartDisease

Result list (right-click for options): 17:38:08 - bayes.NaiveBayes

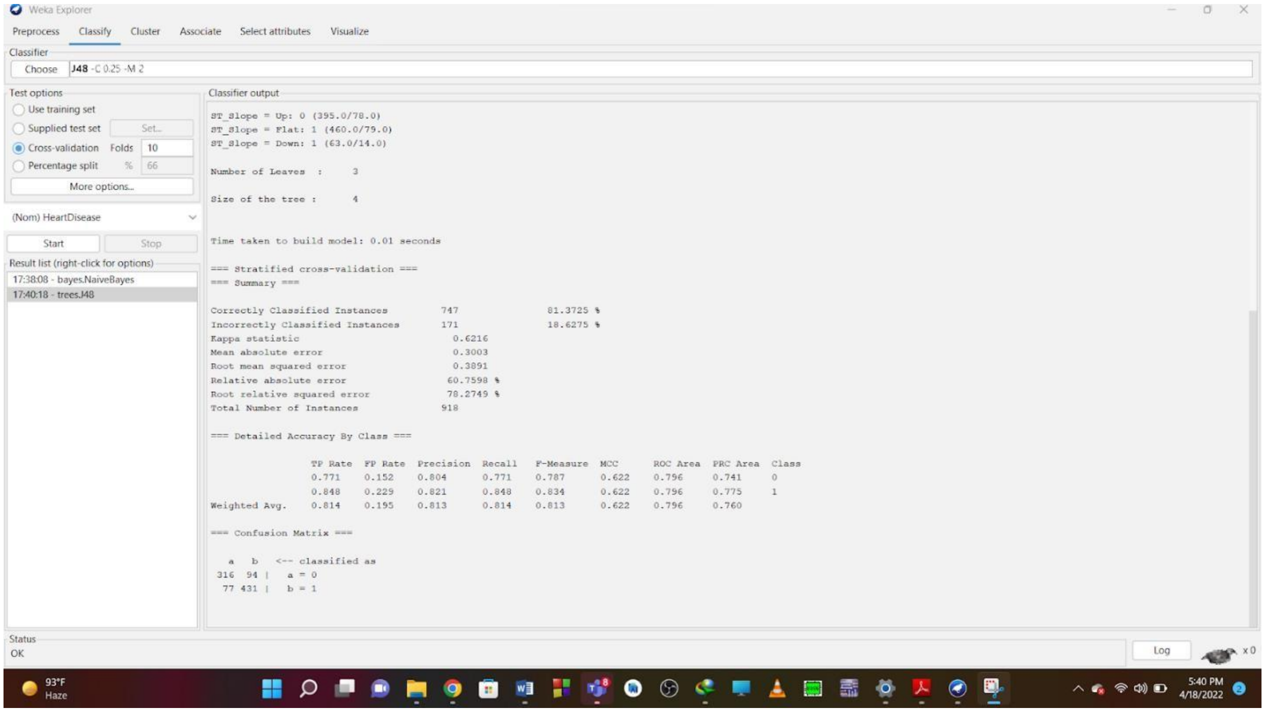
Status: OK

Classifier output

```
ST_Slope
  Up               318.0   79.0
  Flat              80.0  382.0
  Down              15.0   50.0
  [total]          413.0  511.0



Time taken to build model: 0 seconds

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

Correctly Classified Instances         781               85.0763 %
Incorrectly Classified Instances       137               14.9237 %
Kappa statistic                          0.6983
Mean absolute error                      0.168
Root mean squared error                  0.3457
Relative absolute error                 33.9808 %
Root relative squared error             69.5366 %
Total Number of Instances              918

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.837    0.138    0.831      0.837    0.834      0.698    0.912     0.903     0
                 0.862    0.163    0.867      0.862    0.865      0.698    0.912     0.909     1
Weighted Avg.    0.851    0.152    0.851      0.851    0.851      0.698    0.912     0.906

=== Confusion Matrix ===

   a   b   <-- classified as
 343  67 |   a = 0
  70 438 |   b = 1
```

Figure 3& 4: Naïve bayes classification

Applying decision tree classifier: A decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. While classifying the selected dataset, the J48 format was selected for Decision tree classification.



Classifier: J48 -C 0.25 -M 2

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

(Nom) HeartDisease

Result list (right-click for options)
17:38:08 - bayes.NaiveBayes
17:40:18 - trees.J48

Classifier output

```
ST_Slope = Up: 0 (395.0/78.0)
ST_Slope = Flat: 1 (460.0/79.0)
ST_Slope = Down: 1 (63.0/14.0)

Number of Leaves  :     3

Size of the tree :      4


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances         747               81.3725 %
Incorrectly Classified Instances       171               18.6275 %
Kappa statistic                          0.6216
Mean absolute error                      0.3003
Root mean squared error                  0.3891
Relative absolute error                 60.7598 %
Root relative squared error             78.2749 %
Total Number of Instances              918

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.771    0.152    0.804      0.771    0.787      0.622    0.796     0.741     0
                 0.848    0.229    0.821      0.848    0.834      0.622    0.796     0.775     1
Weighted Avg.    0.814    0.195    0.813      0.814    0.813      0.622    0.796     0.760

=== Confusion Matrix ===

   a   b   <-- classified as
 316  94 |   a = 0
  77 431 |   b = 1
```

Status
OK

93°F
Haze

5:40 PM
4/18/2022

Figure 5: Decision tree classification

After applying two types of classifiers, the highest percentage of correctly classified instances is for the naïve Bayes classifier with 85.0763%. After that comes the Decision tree classifier with 81.3725%. The naïve Bayes classifier is considered the best classifier for the dataset.

Reason to choose naïve Bayes classifier: From the obtained result of the two classifiers, it is clearly seen that naïve Bayes has the highest percentage of correctly classified instances which is 85.0763%. As it has the most accurate value so it would be a more suitable classifier for the dataset. the advantages of naïve Bayes is

- It is simple and easy to implement.
- It doesn't require as much training data.

- It handles both continuous and discrete data.
- It is highly scalable with a number of predictors and data points.
- It is fast and can be used to make real-time predictions.

Here is the summary of the naïve Bayes classifiers result:

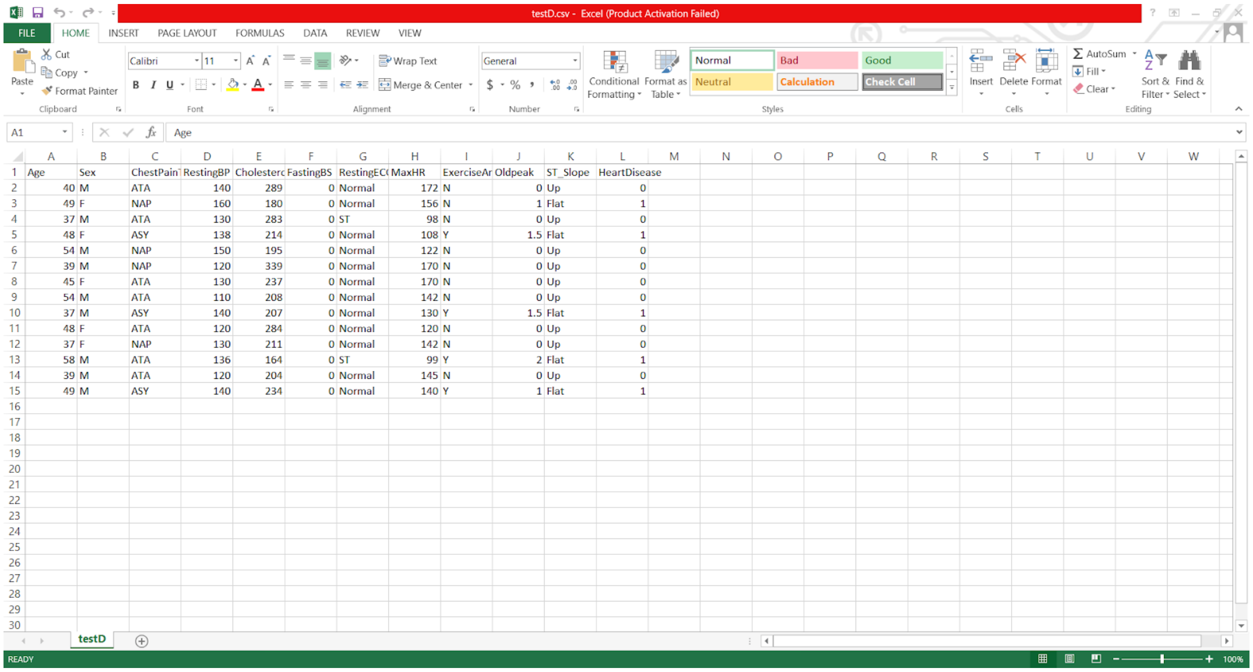
- Correctly Classified Instances : 781 85.0763 %
- Incorrectly Classified Instances : 137 14.9237 %
- Kappa statistic : 0.6983
- Mean absolute error : 0.168
- Root mean squared error : 0.3457
- Relative absolute error : 33.9808 %
- Root relative squared error : 69.5366 %
- Total Number of Instances : 918

DISCUSSION:

In this task 1 section at first, I have chosen a proper dataset that has proper context and content and is also applicable for two of my selected classifiers. After that I selected two classifiers from weka software that is applicable to my dataset with proper classification and accuracy. after applying these classifiers to my dataset, I got two different correctly classified instances, which are 85.0763%, 81.3725%. Also, with the value of some parameters like IP rate fp rate recall f measure, etc., and with the help of the confusion matrix I have selected my best classifier. confusion matrix helps us to identify his good is my algorithm doing. that’s how I have completed task 1 and found the best classifier which is the naïve Bayes classifier.

INTRODUCTION:

One of the most important mechanisms in machine learning is to train your algorithm on a training set that is separate and distinct from the test set for which will be gauged its accuracy. To detect a machine learning behavior, a training dataset has been set which was extracted subset from the referred dataset. Then this model had been tested with a test dataset which is a subset to test the trained model. While preparing the test dataset, things that were made sure are that the dataset was large enough to yield statistically meaningful results. Also, it was representative of the data set as a whole. In other words, the test sets with unusual characteristics than the training set were not chosen. The suitable classifier is then used to predict the classification for the instances in the test set. If the test set contains N instances of which C is correctly classified, C is correctly classified Predictive accuracy, P = C/N. There are 14 instances in this prepared test dataset.

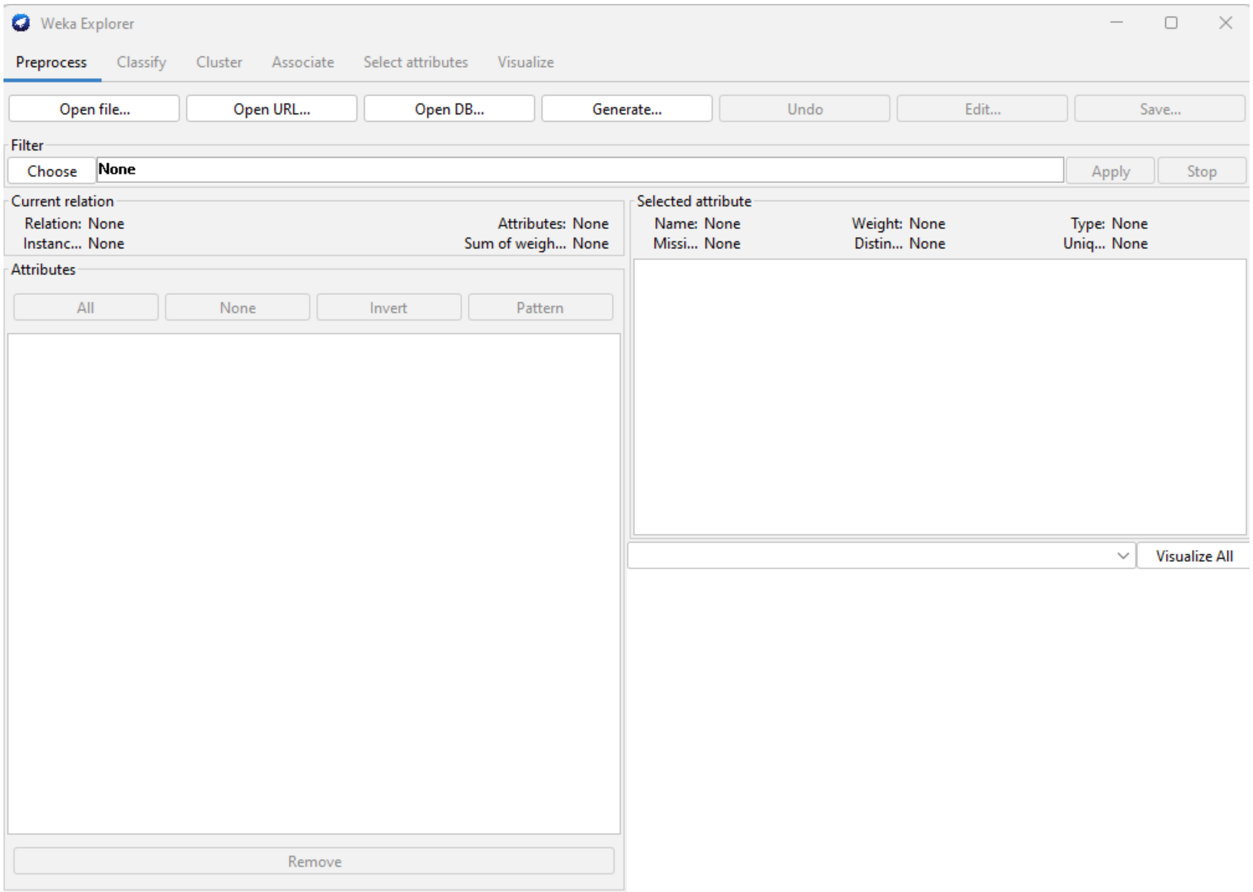


Age	Sex	ChestPain	RestingBP	Cholesterc	FastingBS	RestingEC(	MaxHR	ExerciseAr	Oldpeak	ST_Slope	HeartDisease
40	M	ATA	140	289	0	Normal	172	N	0	Up	0
49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
37	M	ATA	130	283	0	ST	98	N	0	Up	0
48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
54	M	NAP	150	195	0	Normal	122	N	0	Up	0
39	M	NAP	120	339	0	Normal	170	N	0	Up	0
45	F	ATA	130	237	0	Normal	170	N	0	Up	0
54	M	ATA	110	208	0	Normal	142	N	0	Up	0
37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
48	F	ATA	120	284	0	Normal	120	N	0	Up	0
37	F	NAP	130	211	0	Normal	142	N	0	Up	0
58	M	ATA	136	164	0	ST	99	Y	2	Flat	1
39	M	ATA	120	204	0	Normal	145	N	0	Up	0
49	M	ASY	140	234	0	Normal	140	Y	1	Flat	1

Figure 6: test dataset

PROCEDURE OF TESTING THE TEST DATASET:

1. First, Weka 3.8.6 was opened and the ‘Explorer’ option was chosen and such window named weka explorer was opened.



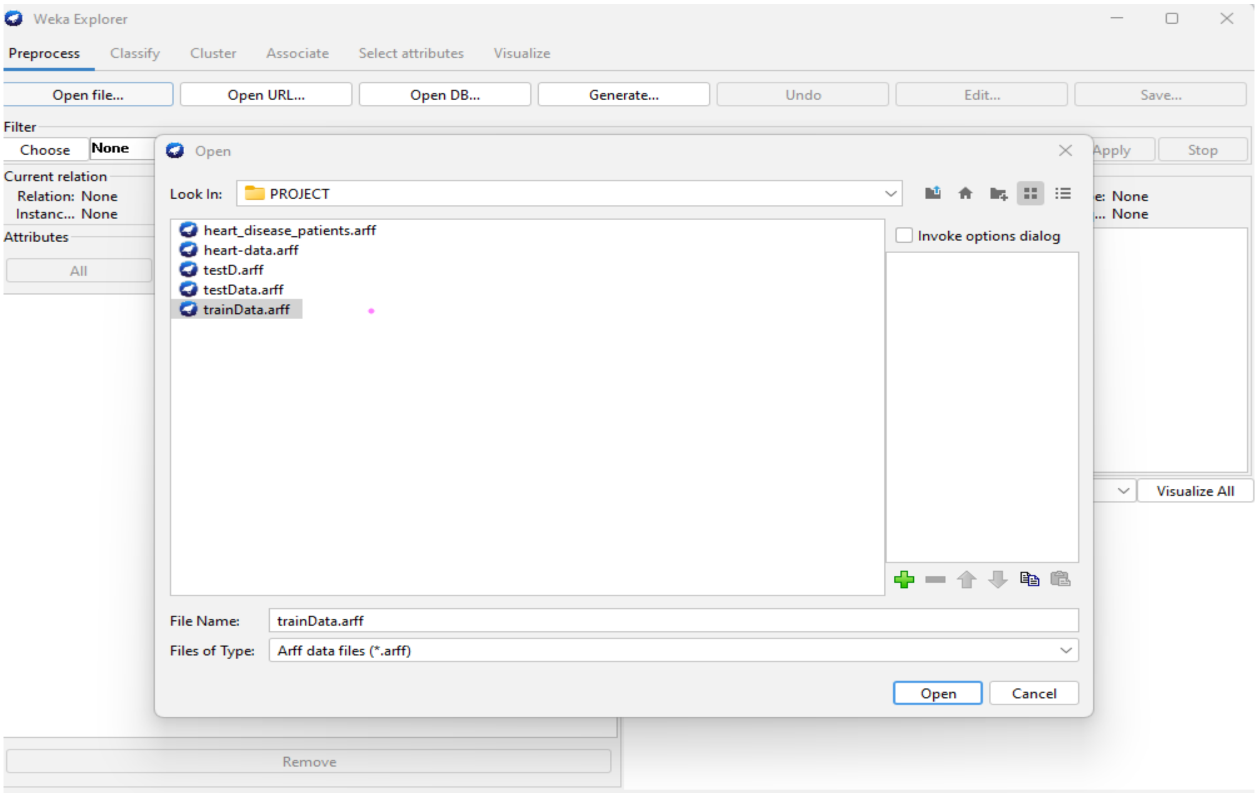
Filter: None

Current relation
Relation: None
Instanc... None
Attributes: None
Sum of weigh... None

Selected attribute
Name: None
Missi... None
Weight: None
Distin... None
Type: None
Uniq... None

Figure 7: Weka Explorer

2. Then, the open file option was selected and the extracted .arff file, training dataset was selected from the device.



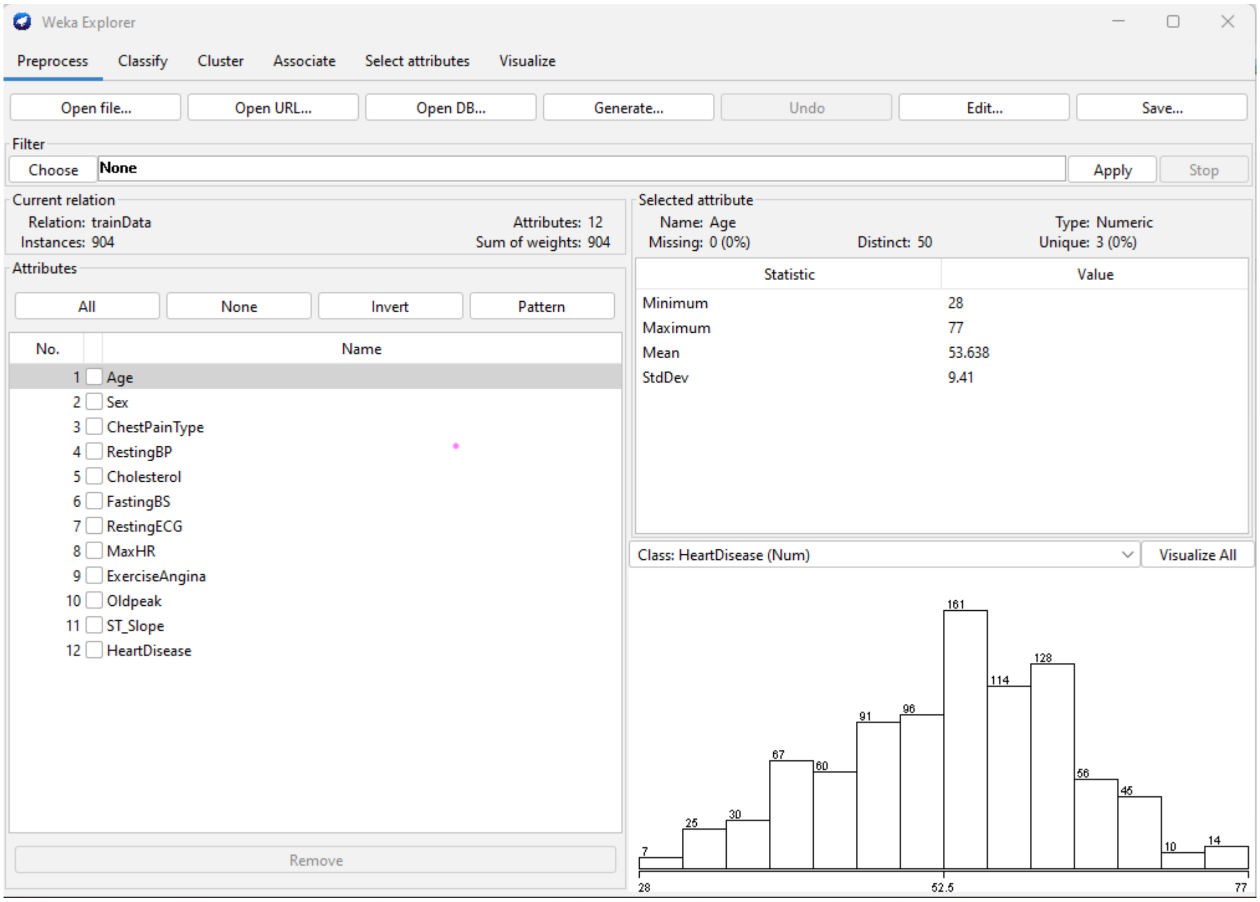
Look In: PROJECT

File Name: trainData.arff

Files of Type: Arff data files (*.arff)

Figure 8: Training Dataset

3. After the open option was clicked, the details of the dataset popped.



Filter: None

Current relation

Relation: trainData
Attributes: 12
Instances: 904
Sum of weights: 904

Attributes

No.	Name
1	□ Age
2	□ Sex
3	□ ChestPainType
4	□ RestingBP
5	□ Cholesterol
6	□ FastingBS
7	□ RestingECG
8	□ MaxHR
9	□ ExerciseAngina
10	□ Oldpeak
11	□ ST_Slope
12	□ HeartDisease

Selected attribute

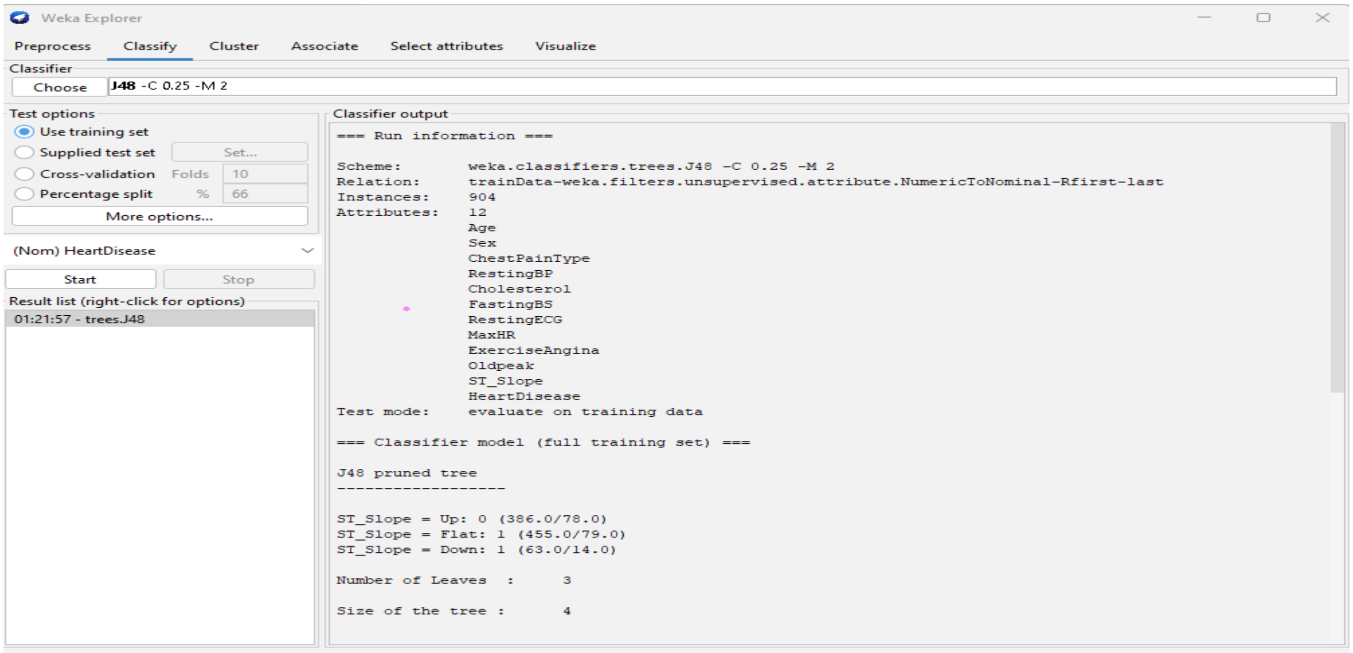
Name: Age
Type: Numeric
Missing: 0 (0%)
Distinct: 50
Unique: 3 (0%)

Statistic	Value
Minimum	28
Maximum	77
Mean	53.638
StdDev	9.41

Class: HeartDisease (Num)

Figure 9: training dataset

4. Then the preferred classifier (Decision tree classifier) for the training dataset was selected. Then from the test options, the use training set was chosen and the start option was selected.



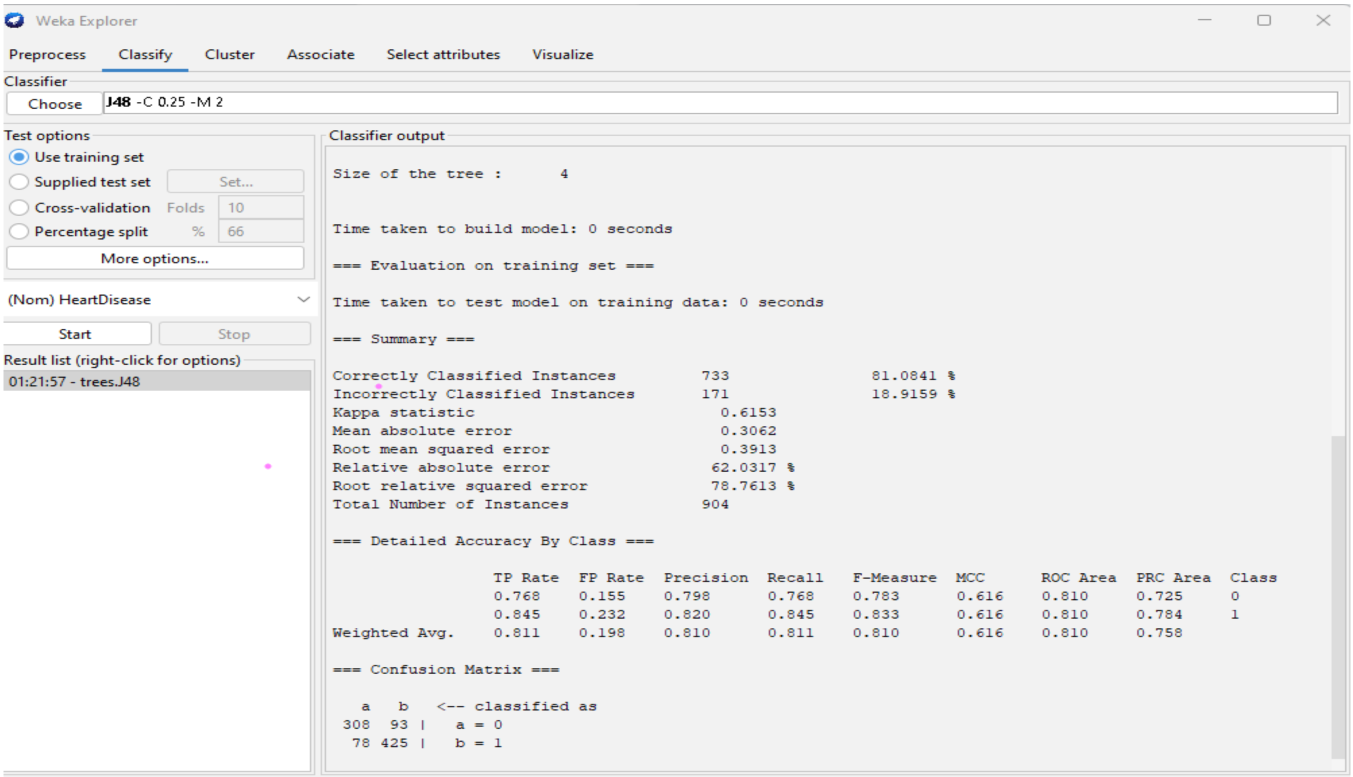
Classifier: J48 -C 0.25 -M 2

Test options: ● Use training set

(Nom) HeartDisease

01:21:57 - trees.J48

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     trainData-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    904
Attributes:   12
              Age
              Sex
              ChestPainType
              RestingBP
              Cholesterol
              FastingBS
              RestingECG
              MaxHR
              ExerciseAngina
              Oldpeak
              ST_Slope
              HeartDisease
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

J48 pruned tree
------------------

ST_Slope = Up: 0 (386.0/78.0)
ST_Slope = Flat: 1 (455.0/79.0)
ST_Slope = Down: 1 (63.0/14.0)

Number of Leaves  :     3

Size of the tree :      4
```



Classifier: J48 -C 0.25 -M 2

Test options: ● Use training set

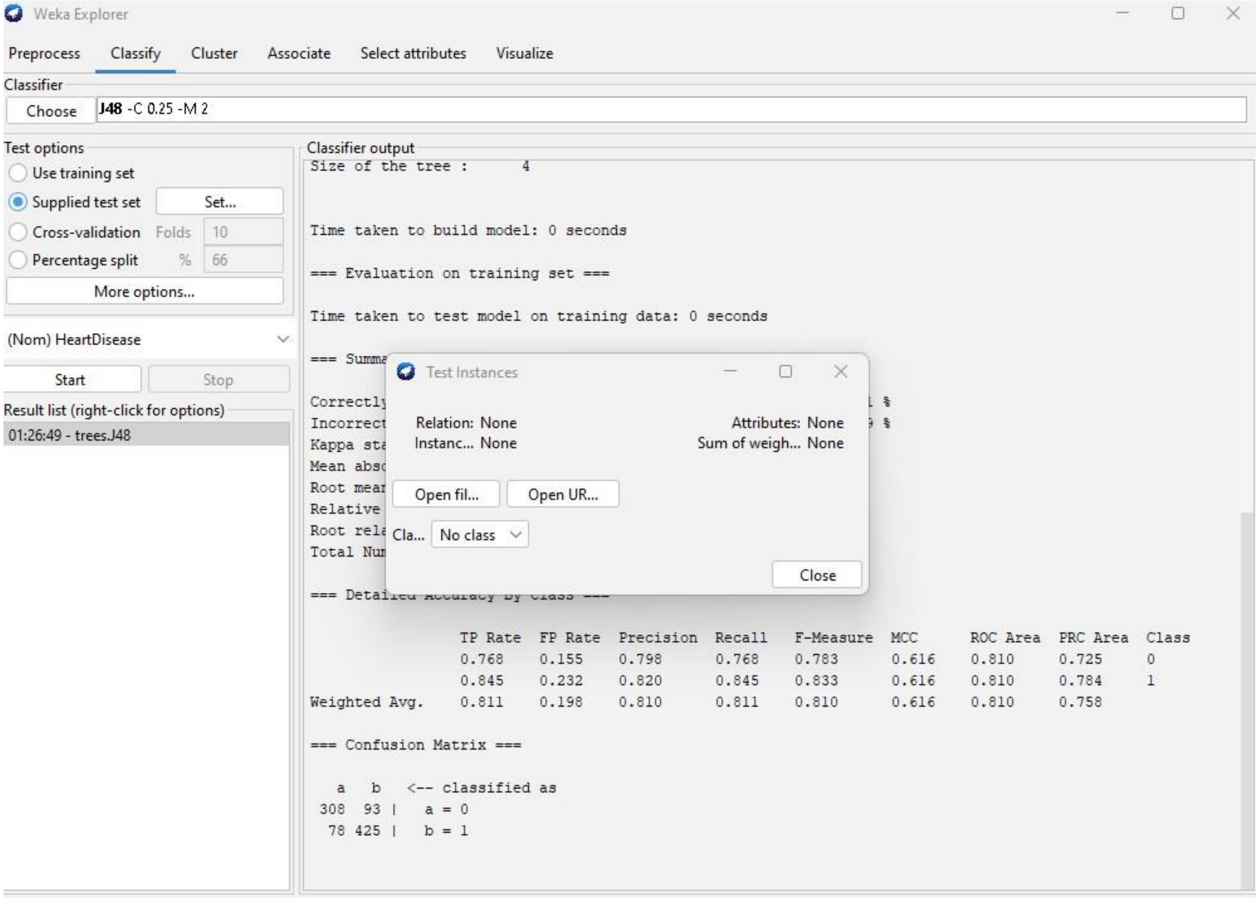
(Nom) HeartDisease

01:21:57 - trees.J48

```
Size of the tree :      4


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         733               81.0841 %
Incorrectly Classified Instances       171               18.9159 %
Kappa statistic                          0.6153
Mean absolute error                      0.3062
Root mean squared error                  0.3913
Relative absolute error                 62.0317 %
Root relative squared error             78.7613 %
Total Number of Instances              904

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.768    0.155    0.798      0.768    0.783      0.616    0.810     0.725     0
                 0.845    0.232    0.820      0.845    0.833      0.616    0.810     0.784     1
Weighted Avg.    0.811    0.198    0.810      0.811    0.810      0.616    0.810     0.758

=== Confusion Matrix ===

   a   b   <-- classified as
 308  93 |   a = 0
  78 425 |   b = 1
```

Figures 10 & 11: the result of the training set

5. To input the test set, from the test options, supplied test data was selected and then a window named test instances came.



Classifier: J48 -C 0.25 -M 2

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

(Nom) HeartDisease

Result list (right-click for options)
01:26:49 - trees.J48

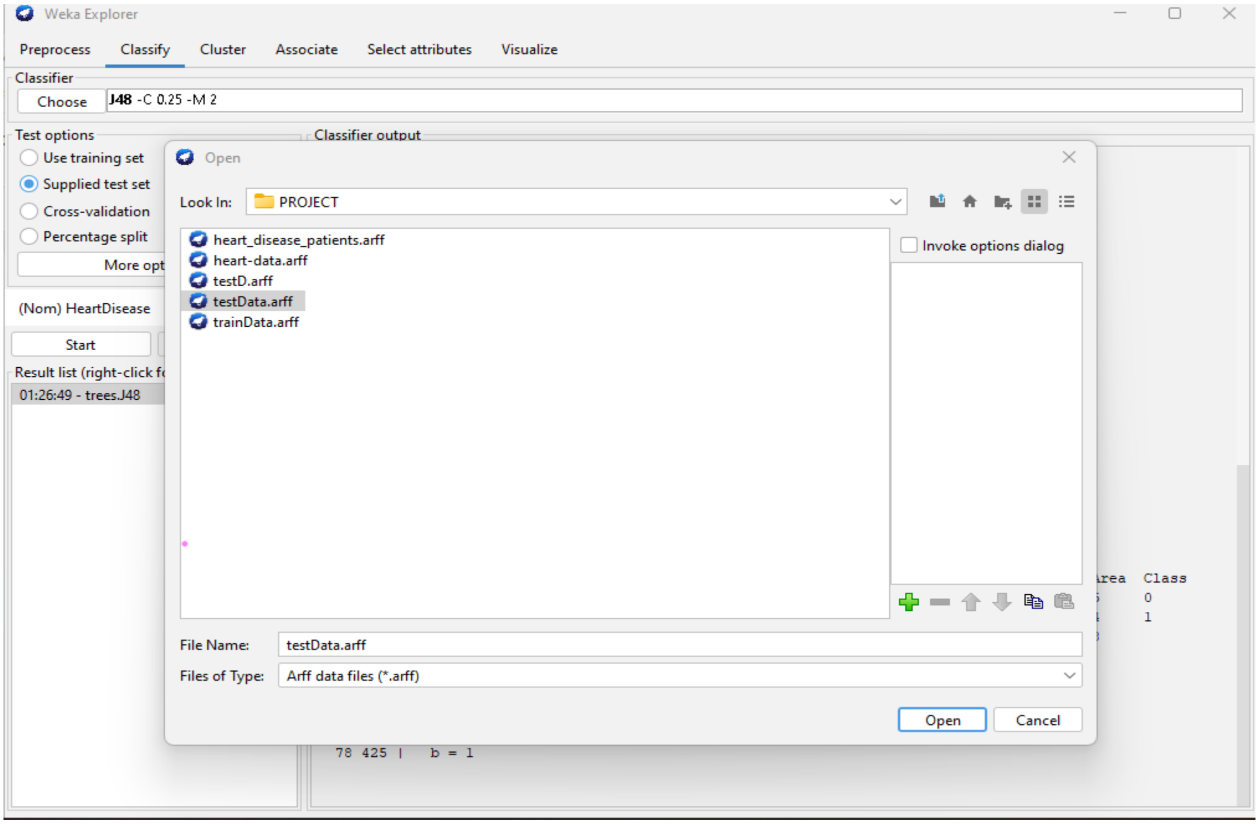
Test Instances
Relation: None
Instanc... None
Attributes: None
Sum of weigh... None
Cla... No class

Classifier output

```
Size of the tree :      4


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summa

Correctly                                                  1 %
Incorrect                                                  9 %
Kappa sta
Mean abso
Root mear
Relative
Root rela
Total Nu

=== Detai

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.768    0.155    0.798      0.768    0.783      0.616    0.810     0.725     0
                 0.845    0.232    0.820      0.845    0.833      0.616    0.810     0.784     1
Weighted Avg.    0.811    0.198    0.810      0.811    0.810      0.616    0.810     0.758

=== Confusion Matrix ===

   a   b   <-- classified as
 308  93 |   a = 0
  78 425 |   b = 1
```

Figure 12: supplied test set selected

6. Then the open file option was chosen to insert the test data and then the dataset was opened.



Classifier: J48 -C 0.25 -M 2

Test options: ○ Use training set ● Supplied test set ○ Cross-validation ○ Percentage split

(Nom) HeartDisease

01:26:49 - trees.J48

Look In: PROJECT

- heart_disease_patients.arff
- heart-data.arff
- testD.arff
- testData.arff
- trainData.arff

□ Invoke options dialog

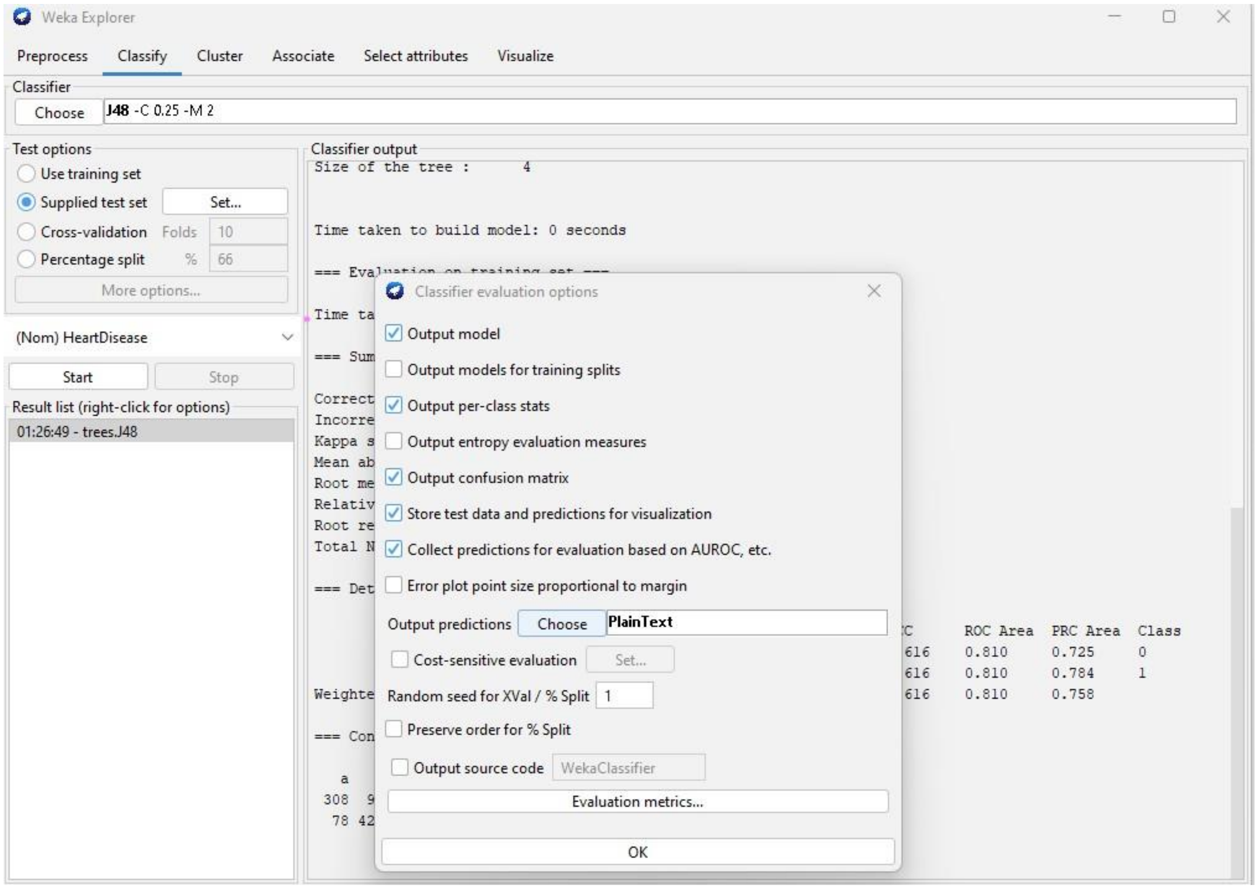
File Name: testData.arff

Files of Type: Arff data files (*.arff)

```
rea   Class
        0
        1
 78 425 |   b = 1
```

Figure 13: test set opened

7. To make sure the test set works properly, it was made sure that the output predictions was in PlainTest form



Classifier: J48 -C 0.25 -M 2

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

(Nom) HeartDisease

01:26:49 - trees.J48

```
Size of the tree :      4


Time taken to build model: 0 seconds

=== Evaluation on training set ===

Time ta

=== Sum

Correct
Incorre
Kappa s
Mean ab
Root me
Relativ
Root re
Total N

=== Det

                                                        C     ROC Area  PRC Area  Class
                                                      616     0.810     0.725     0
                                                      616     0.810     0.784     1
Weighte                                               616     0.810     0.758

=== Con

  a
 308  9
  78 42
```

Classifier evaluation options

☑ Output model
□ Output models for training splits
☑ Output per-class stats
□ Output entropy evaluation measures
☑ Output confusion matrix
☑ Store test data and predictions for visualization
☑ Collect predictions for evaluation based on AUROC, etc.
□ Error plot point size proportional to margin

Output predictions: Choose PlainText

□ Cost-sensitive evaluation Set...

Random seed for XVal / % Split: 1

□ Preserve order for % Split

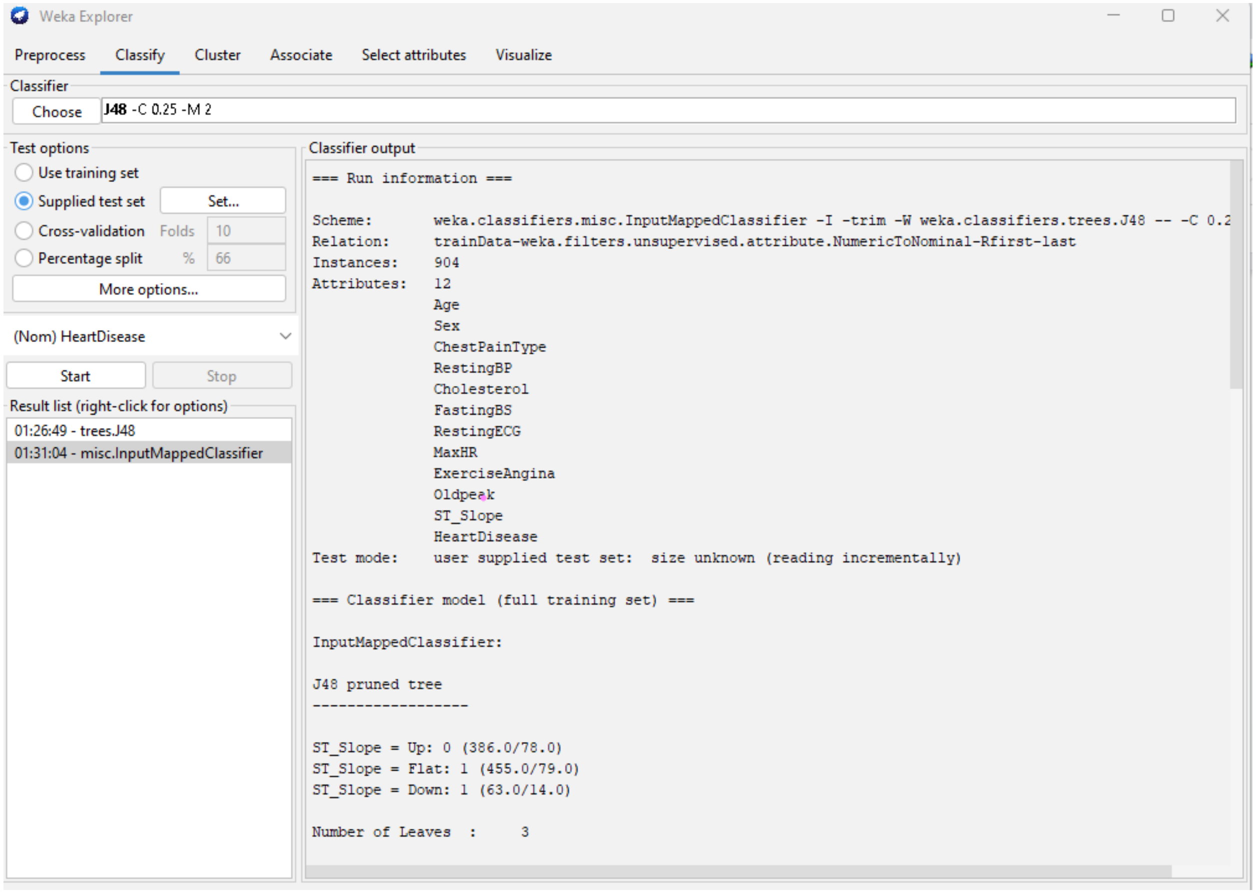
□ Output source code WekaClassifier

Evaluation metrics...

OK

Figure 14: PlainText format chosen for output prediction

8. Then the start button was clicked.



Classifier

J48 -C 0.25 -M 2

Test options

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

(Nom) HeartDisease

Result list (right-click for options)

01:26:49 - trees.J48
01:31:04 - misc.InputMappedClassifier

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.misc.InputMappedClassifier -I -trim -W weka.classifiers.trees.J48 -- -C 0.2
Relation:     trainData-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last
Instances:    904
Attributes:   12
              Age
              Sex
              ChestPainType
              RestingBP
              Cholesterol
              FastingBS
              RestingECG
              MaxHR
              ExerciseAngina
              Oldpeak
              ST_Slope
              HeartDisease
Test mode:    user supplied test set:  size unknown (reading incrementally)

=== Classifier model (full training set) ===

InputMappedClassifier:

J48 pruned tree
------------------

ST_Slope = Up: 0 (386.0/78.0)
ST_Slope = Flat: 1 (455.0/79.0)
ST_Slope = Down: 1 (63.0/14.0)

Number of Leaves  :     3
```

Figure 15: Test set run

RESULT OF TEST-DATASET MODEL:

Once the start button was clicked, the output for the test dataset came. In the result, the test mode was ‘user-supplied test set’ which means the classifier is evaluated on how well it predicts the class of a set of instances loaded from a file that was inputted by the user. The total time taken to build the model was 0.0 seconds and among the 14 instances, there were 0 instances where the error was found. In this model, the Correctly classified instances % Value describes the amount of accuracy of correctly classified instances provided by the algorithm. In this case, the percentage is 100%. Incorrectly classified instances % Value describes how much incorrect instances are given by the algorithm. In this case, the percentage is 0%.

Mean Absolute Error (MAE): It can define as a statistical measure of how far an estimate is from actual values i.e. the average of the absolute magnitude of the individual errors. It is usually similar in magnitude but slightly smaller than the root mean squared error. In this model the MAE is 0.1919

Root Mean-Squared Error (RMSE): The Root Mean Square Error (RMSE) calculates the differences between values predicted by a model / an estimator and the values observed from the thing being modeled/ estimated. RMSE is used to measure the accuracy. It is ideal if it is small. In this case, the RMSE is 0.1924 which is ideal.

DISCUSSION:

The idea behind training and test sets is to test the generalization error. Separating data into training and testing sets is an important part of evaluating data mining models. Since the test set data already contains known values for the property I want to predict, it is easy to determine if the model's assumptions are correct. So here I carefully separate the data into test/training and apply them in weka after that it gives the error table and accuracy rate that’s how I get my proper result and accuracy model.

TASK-3

INTRODUCTION:

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the mode**l**. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data. Unsupervised learning is helpful for finding useful insights from the data. Unsupervised learning is much similar to human learning to think by their own experiences, which makes it closer to the real AI. Unsupervised learning works on unlabeled and uncategorized data which makes unsupervised learning more important.

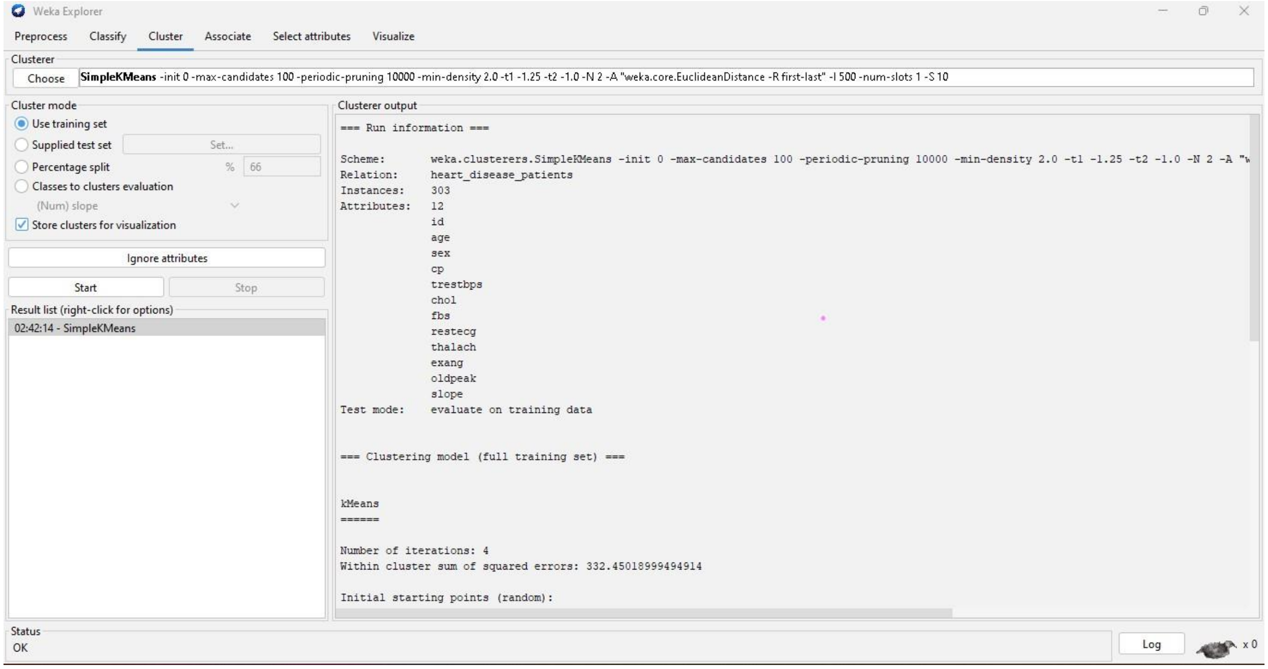
I have chosen “ Heart_Disease_Patients ” dataset. I will also use K means clustering Algorithm.

About the dataset:n this report, the used “ Heart_Disease_Patients ”, a CSV dataset file [Later converted into .arff], collected from Kaggle.com.

RESULT:

Applying K means clustering Algorithm.

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and is a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.



Clusterer

SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

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

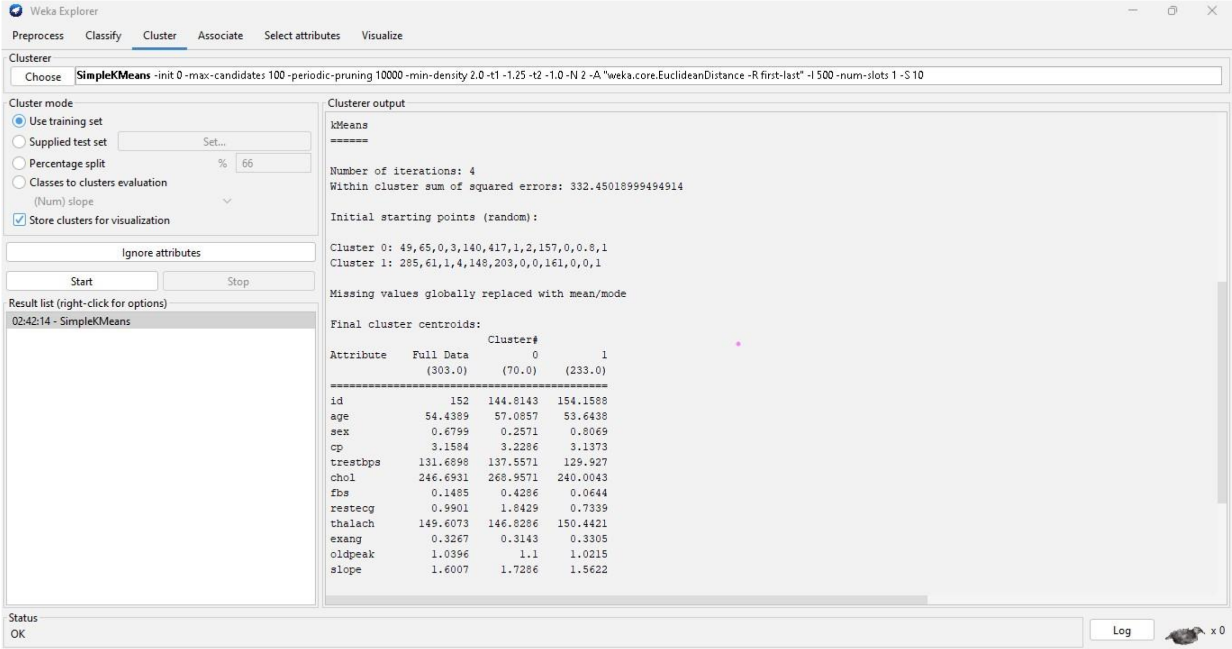
Result list (right-click for options)

02:42:14 - SimpleKMeans

Clusterer output

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "w
Relation:     heart_disease_patients
Instances:    303
Attributes:   12
              id
              age
              sex
              cp
              trestbps
              chol
              fbs
              restecg
              thalach
              exang
              oldpeak
              slope
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 332.45018999494914

Initial starting points (random):
```

Status
OK



Clusterer

SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

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

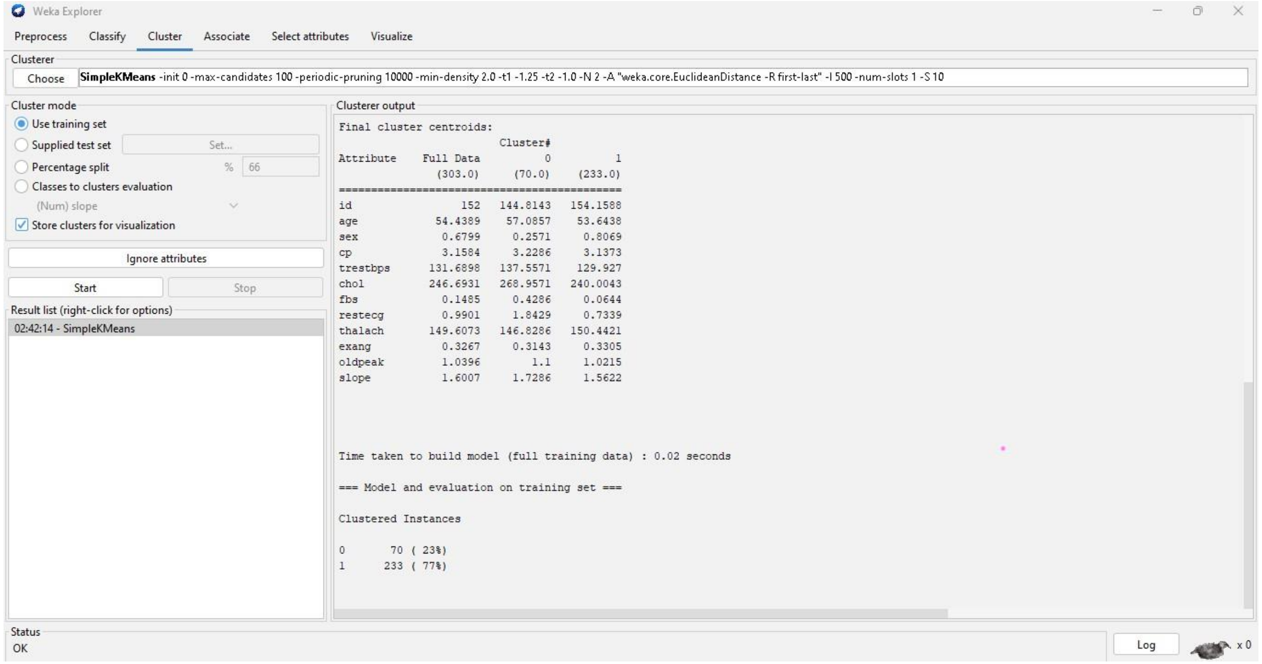
Result list (right-click for options)

02:42:14 - SimpleKMeans

Clusterer output

```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 332.45018999494914

Initial starting points (random):

Cluster 0: 49,65,0,3,140,417,1,2,157,0,0.8,1
Cluster 1: 285,61,1,4,148,203,0,0,161,0,0,1

Missing values globally replaced with mean/mode

Final cluster centroids:
                          Cluster#
Attribute    Full Data          0          1
               (303.0)     (70.0)    (233.0)
============================================
id                 152   144.8143   154.1588
age            54.4389    57.0857    53.6438
sex             0.6799     0.2571     0.8069
cp              3.1584     3.2286     3.1373
trestbps      131.6898   137.5571    129.927
chol          246.6931   268.9571   240.0043
fbs             0.1485     0.4286     0.0644
restecg         0.9901     1.8429     0.7339
thalach       149.6073   146.8286   150.4421
exang           0.3267     0.3143     0.3305
oldpeak         1.0396        1.1     1.0215
slope           1.6007     1.7286     1.5622
```

Status
OK



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

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

Result list (right-click for options)

02:42:14 - SimpleKMeans

Clusterer output

```
Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
               (303.0)     (70.0)    (233.0)
============================================
id                 152   144.8143   154.1588
age            54.4389    57.0857    53.6438
sex             0.6799     0.2571     0.8069
cp              3.1584     3.2286     3.1373
trestbps      131.6898   137.5571    129.927
chol          246.6931   268.9571   240.0043
fbs             0.1485     0.4286     0.0644
restecg         0.9901     1.8429     0.7339
thalach       149.6073   146.8286   150.4421
exang           0.3267     0.3143     0.3305
oldpeak         1.0396        1.1     1.0215
slope           1.6007     1.7286     1.5622



Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0       70 ( 23%)
1      233 ( 77%)
```

Status

OK

Figures 16, 17,18: K means clustering Algorithm

Here is the summary of the K means clustering Algorithm result:

Number of iterations: 4

Initial starting points (random):

Cluster 0: 49,65,0,3,140,417,1,2,157,0,0.8,1

Cluster 1: 285,61,1,4,148,203,0,0,161,0,0,1

Time taken to build model (full training data) : 0.02 seconds

Clustered Instances

0 70 (23%)

1 233 (77%)

DISCUSSION:

I have chosen a proper unsupervised dataset. I convert csv file to arff file and then apply K means Clustering Algorithm. By default, the value of K is 2 so there are two results.

For 0 instances it is 23% whereas for 1 is 77%. I also find the final cluster of centroids also. By that I have completed K means clustering for unsupervised data.

Reference:

1. https://www.kaggle.com/datasets/kingabzpro/heart-disease-patients
2. https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction
3. https://en.wikipedia.org/wiki/Data_mining
4. https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning