



**Faculty of Engineering & Technology**  
**Electrical & Computer Engineering Department**

**Artificial Intelligence - ENCS3340**

**Project #2**

**Machine Learning for Classification**

---

**Prepared by:**

Baraa Nasar                      1210880

Ahmad Abu Saleem              1201315

**Instructor:** Dr. Yazan Abu Farha, Dr. Ismail Khater

**Section:** 2 , 4

**Date:** 6/23/2024

## Table of Contents

Table of Figures .....	2
Introduction .....	3
Decision Tree.....	4
Discretization .....	4
5-Fold Cross Validation.....	6
Changing hyper-parameters .....	10
Naïve Bayes.....	12
Discretization .....	12
5-Fold Cross Validation.....	14
3. Changing hyper-parameters .....	15
Multi Layer Perceptron (MLP).....	16
Discretization .....	16
5-Fold Cross Validation.....	17
Changing hyper-parameters .....	18
Conclusion .....	20

## Table of Figures

Figure 1: Data before Discretization.....	4
Figure 2: Data after Discretization with bins=10.....	4
Figure 3: Data after Discretization with bins=4.....	5
Figure 4: Data after Discretization with bins=2.....	5
Figure 5: Classifier Output, bins=10.....	6
Figure 6: Tree, bins=10.....	6
Figure 7: Classifier Output, bins=10.....	7
Figure 8: Classifier Output, bins=2.....	7
Figure 9: Tree, bins=2.....	8
Figure 10: Classifier output, bins=2.....	8
Figure 11: Classifier Output, bins=4.....	9
Figure 12: Tree, bins=4.....	9
Figure 13: Classifier Output, bins=4.....	9
Figure 14: Changing Confidence Factor.....	10
Figure 15: Classifier Output.....	10
Figure 16: Tree.....	11
Figure 17: Classifier Output.....	11
Figure 18: Attributes before discretizing .....	12
Figure 19: Settings for Prepressing.....	12
Figure 20: Attributes after discretizing .....	13
Figure 21: The result of using Naive Bayes as Classifier .....	14
Figure 22: Naive Bayes results after changing Hyper Parameters.....	15
Figure 23: Attributes before discretization .....	16
Figure 24: Settings for Prepressing.....	16
Figure 25: Attributes after discretization .....	17
Figure 26: The result of using Multi Layer Perceptron (MLP) as Classifier.....	17
Figure 27: Changing hyper-parameter for MLP .....	18
Figure 28: Confusion Matrix after Changing hyper-parameter for MLP.....	18

## Introduction

In this project three machine learning algorithms were implemented using WEKA tool. The data set was Raisin data set, and it was chosen based on the last digit of the least student id in the team  $\% 3 \rightarrow 1201315 \rightarrow 5\%3 \rightarrow 2$ .

The image shows the WEKA software interface. At the top, the 'Current relation' section displays 'Relation: Raisin\_Grains\_Dataset' and 'Instances: 900'. To the right, a box shows 'Attributes: 8' and 'Sum of weights: 900'. Below this, the 'Attributes' section contains four buttons: 'All', 'None', 'Invert', and 'Pattern'. A table lists the attributes, with the first attribute, 'Area', selected and highlighted in blue. The table has two columns: 'No.' and 'Name'.

No.	Name
1	Area
2	MajorAxisLength
3	MinorAxisLength
4	Eccentricity
5	ConvexArea
6	Extent
7	Perimeter
8	Class

Raisin Data set has these attributes.

Number of attributes are 8, Number of instances is 900.

## Decision Tree Discretization

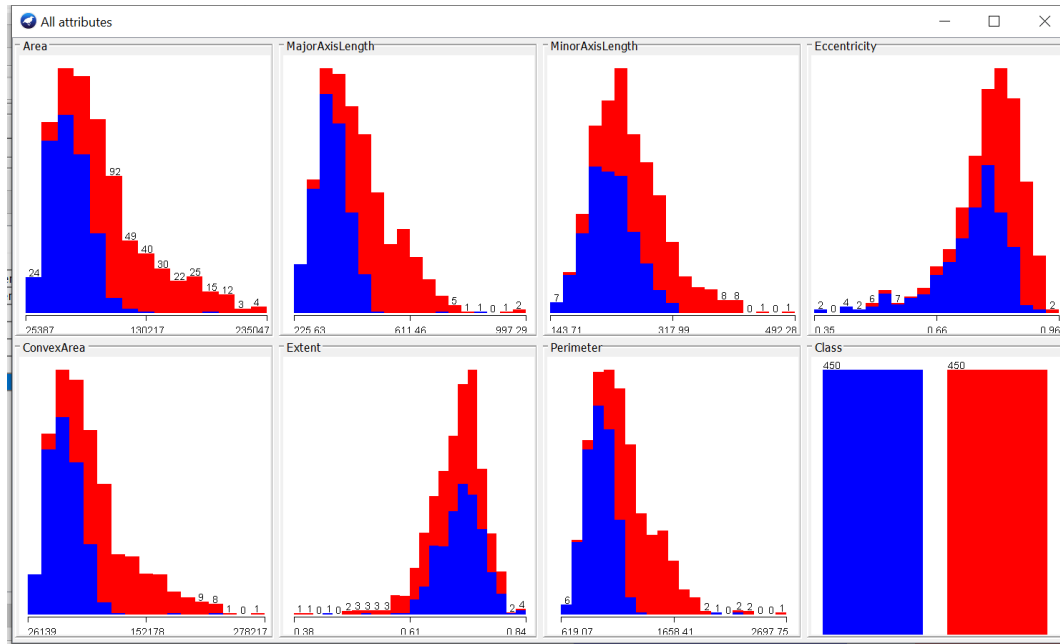


Figure 1: Data before Discretization

When number of bins were set to 10:

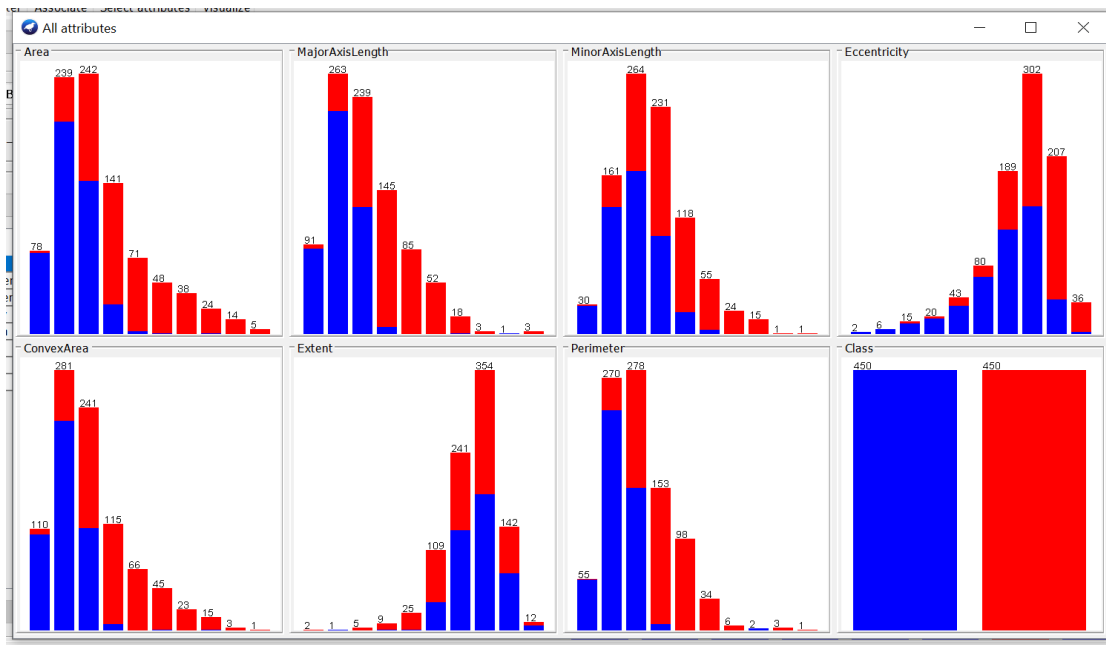


Figure 2: Data after Discretization with bins=10

**When number of bins were set to 4:**

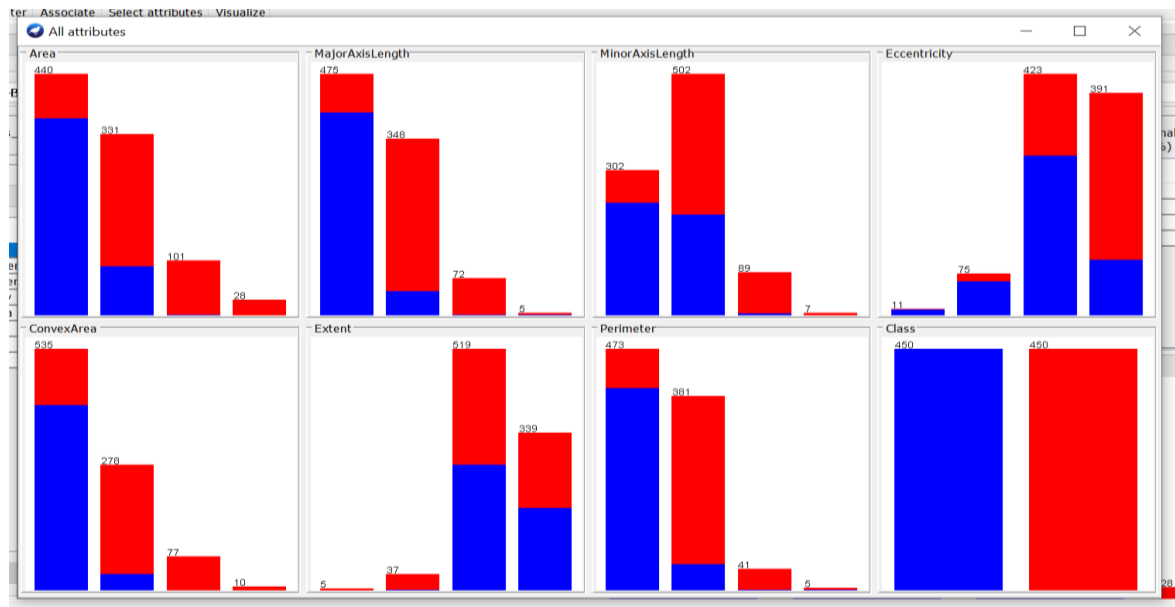


Figure 3: Data after Discretization with bins=4

**When number of bins were set to 2:**

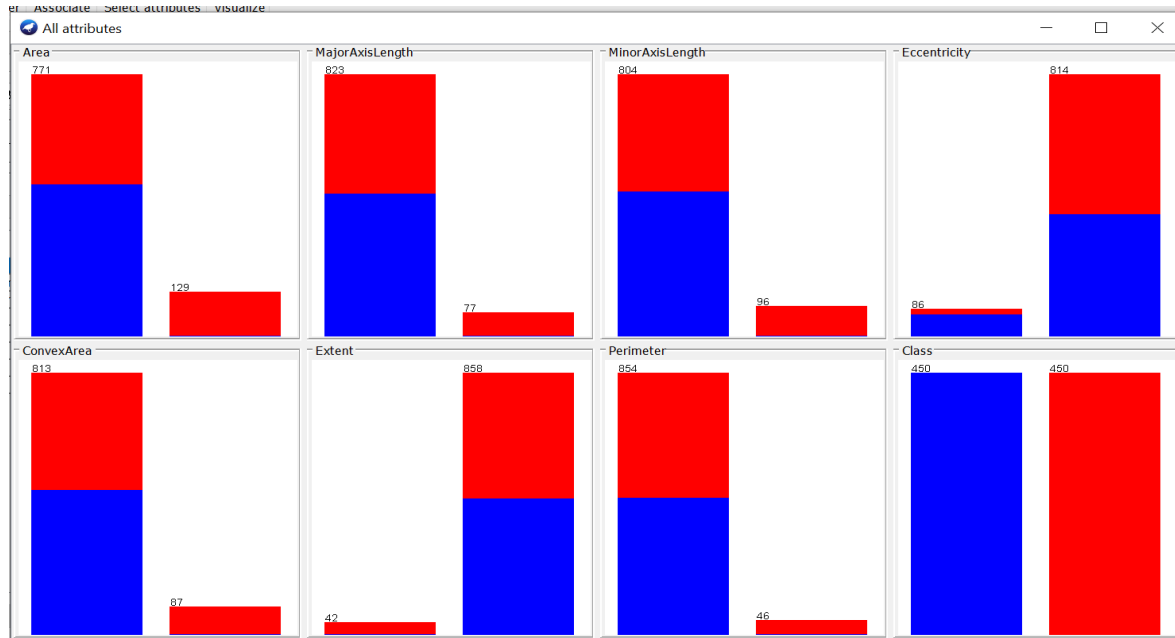


Figure 4: Data after Discretization with bins=2

## 5-Fold Cross Validation

Using the 5-fold cross validation to train the model, and applying Decision Tree algorithm as a classifier

**The case where there are 10 bins:**

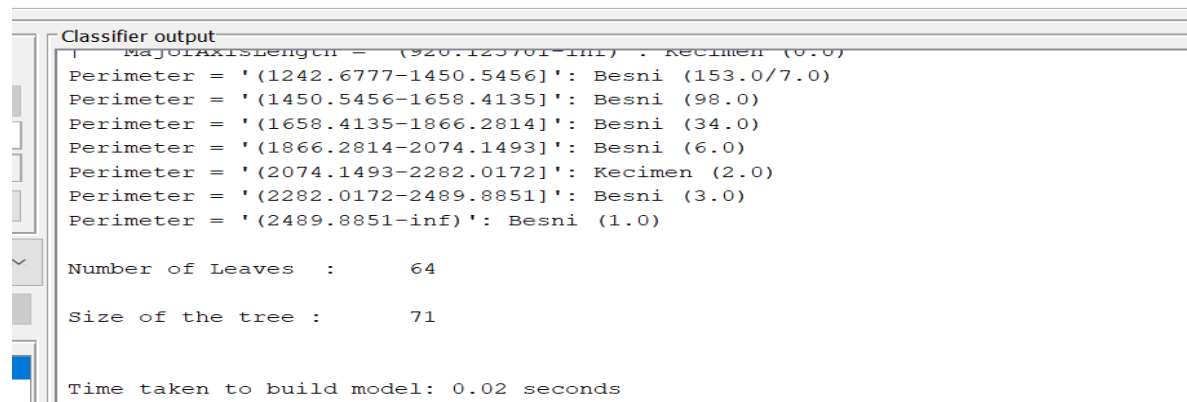


Figure 5: Classifier Output, bins=10

Here number of leaves as shown above is 64 and the tree size is 71.

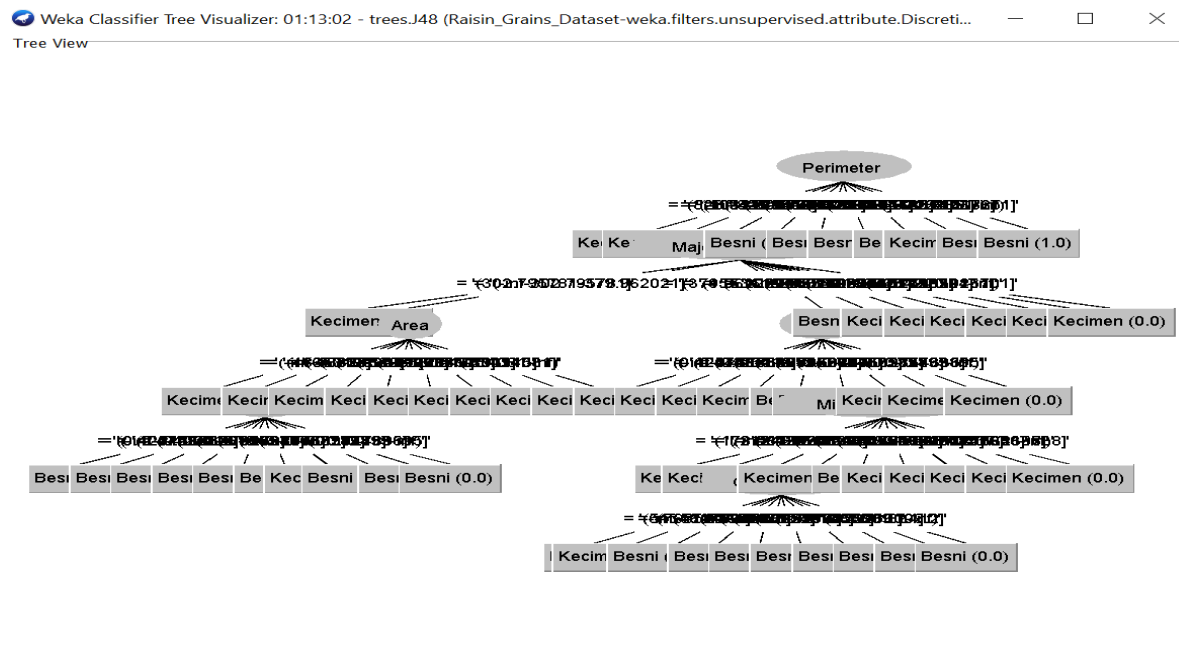


Figure 6: Tree, bins=10

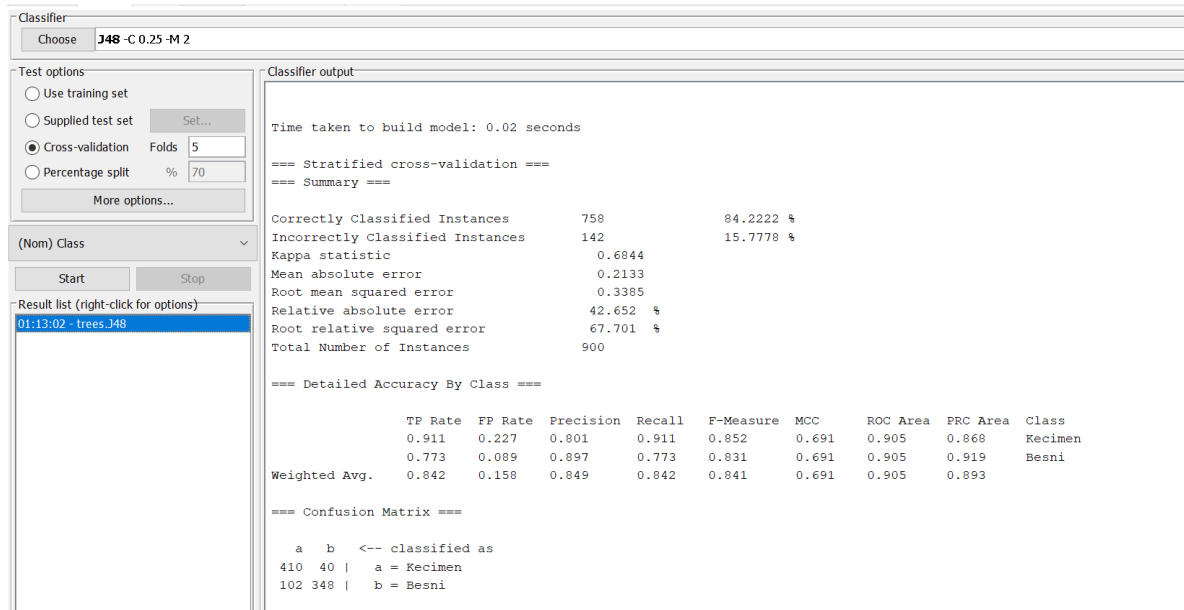


Figure 7: Classifier Output, bins=10

- From the figure, it is shown that the correctly classified instances are **84.22%**.
- From confusion matrix part, **TP =410, FN =40, FP = 102** and **TN = 348**. As the precision is **0.801**, recall **0.911**, F-Measure = **0.852**. These results could be acceptable.

### The case where there are 2 bins:

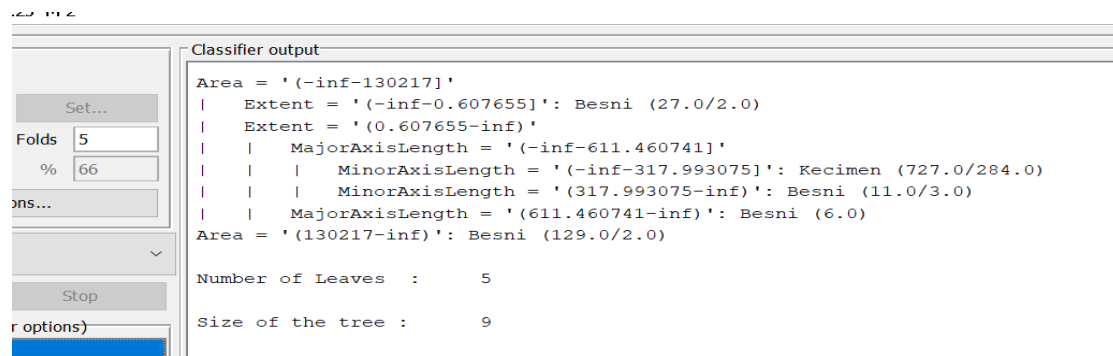


Figure 8: Classifier Output, bins=2



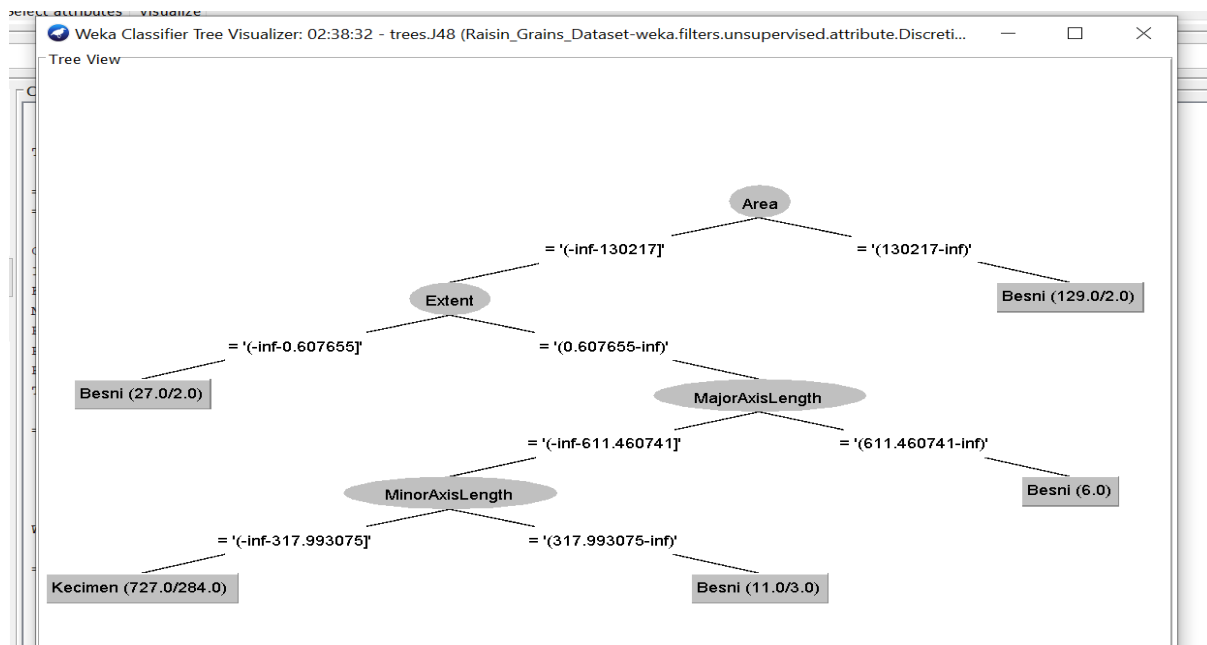


Figure 9: Tree, bins=2

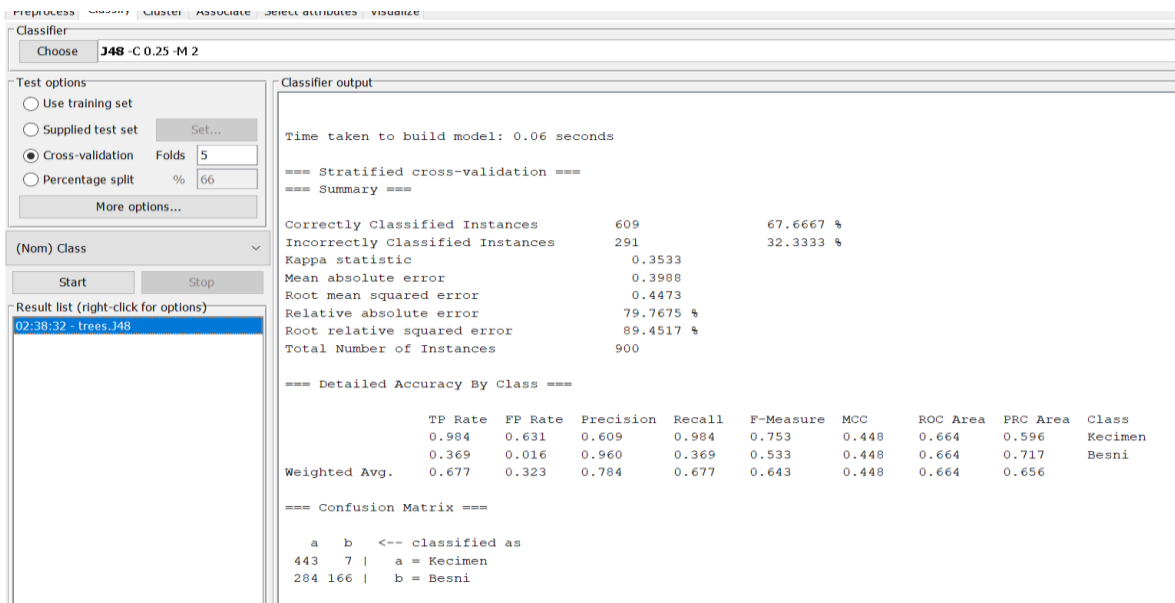


Figure 10: Classifier output, bins=2

- The correctly classified instances are **67.67%**.
- From confusion matrix part, **TP =443, FN =7, FP = 284** and **TN = 166**. As the precision is **0.609**, recall **0.984**, F-Measure = **0.753**. These results are worse than the previous one, it returns to many vales with a high recall, but not all of them is the required and that is shown in precision value.

## The case where there are 4 bins:

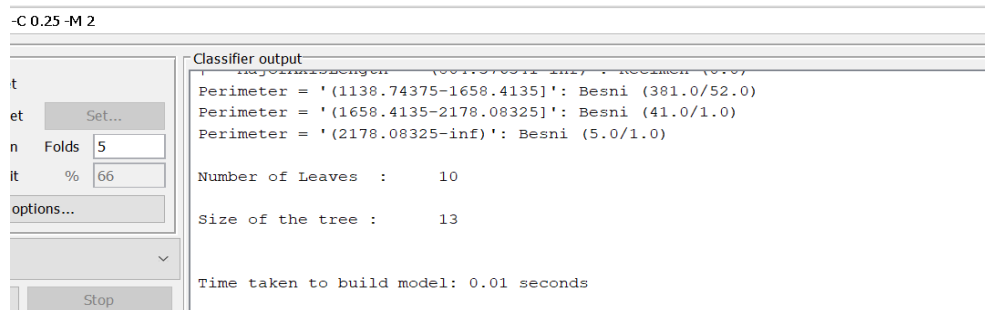


Figure 11: Classifier Output, bins=4

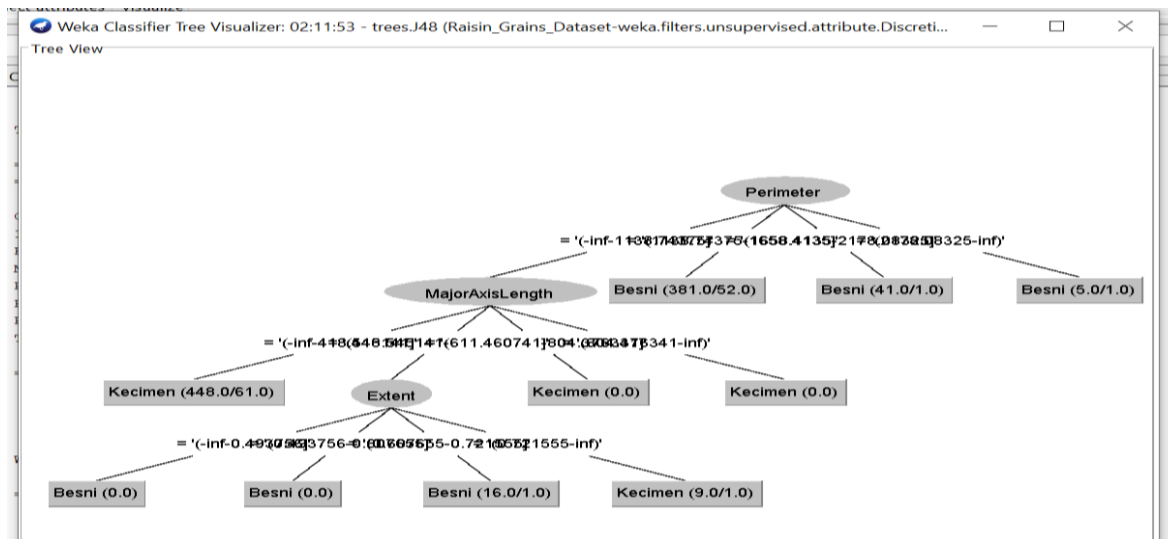


Figure 12: Tree, bins=4

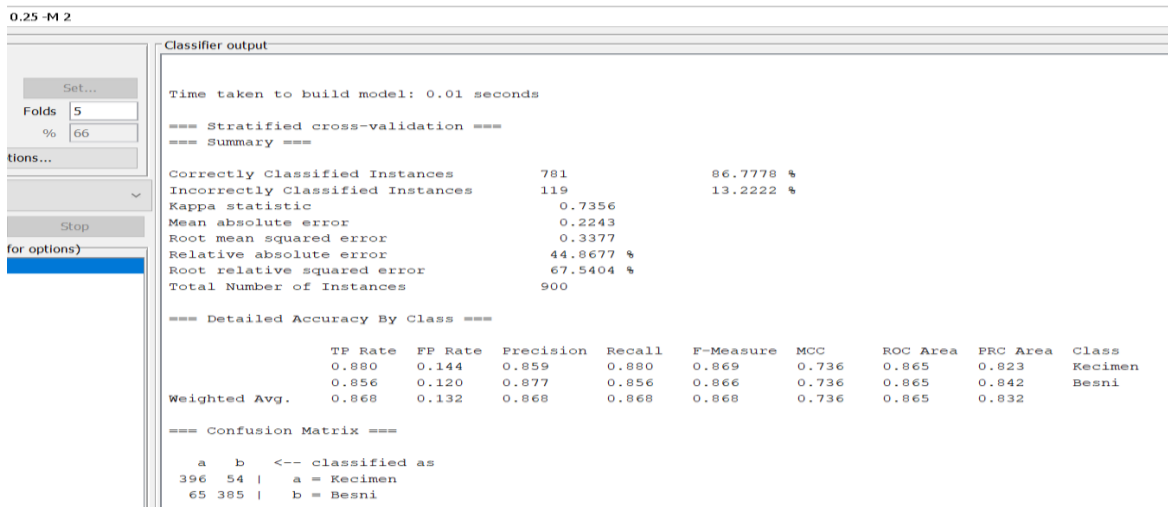


Figure 13: Classifier Output, bins=4

- The correctly classified instances are **86.778%**.
- From confusion matrix part, **TP = 396, FN = 54, FP = 65** and **TN = 385**. As the precision is **0.859**, recall **0.880**, F-Measure = **0.869**. These results are the best among the previous ones, both precision and recall are so close to each other, with a good result.

## Changing hyper-parameters

Changing was done on 4 bins data output, and the confidence factor was set to 0.75 from 0.25.

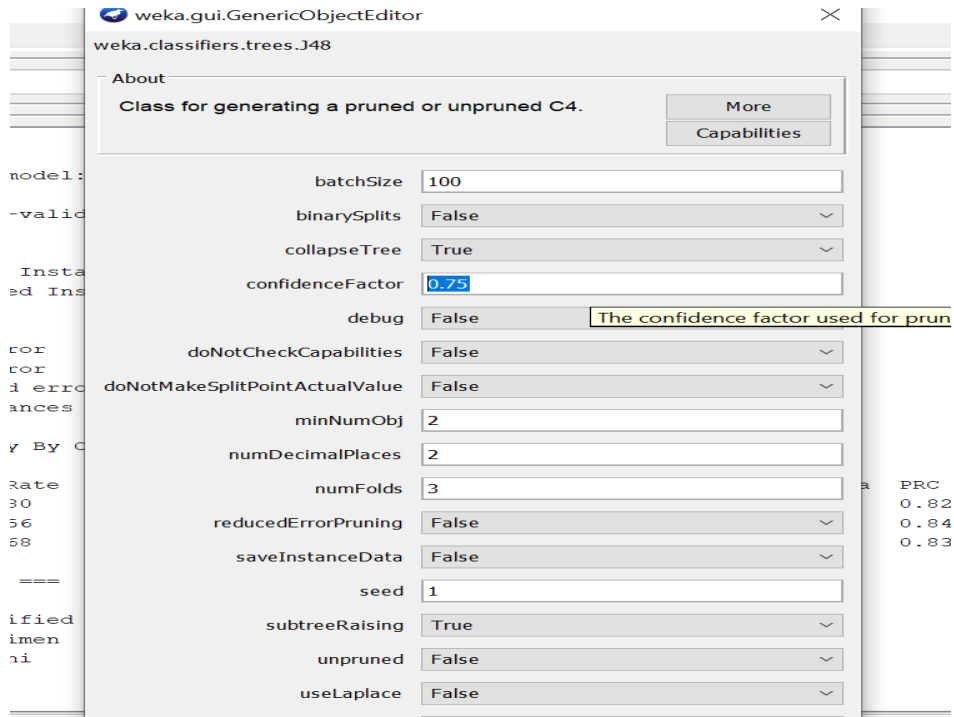


Figure 14: Changing Confidence Factor

```

Classifier output
Perimeter = '(1658.4135-2178.08325)': Besni (41.0/1.0)
Perimeter = '(2178.08325-inf)': Besni (5.0/1.0)

Number of Leaves :    28

Size of the tree :    37

Time taken to build model: 0.11 seconds

```

Figure 15: Classifier Output

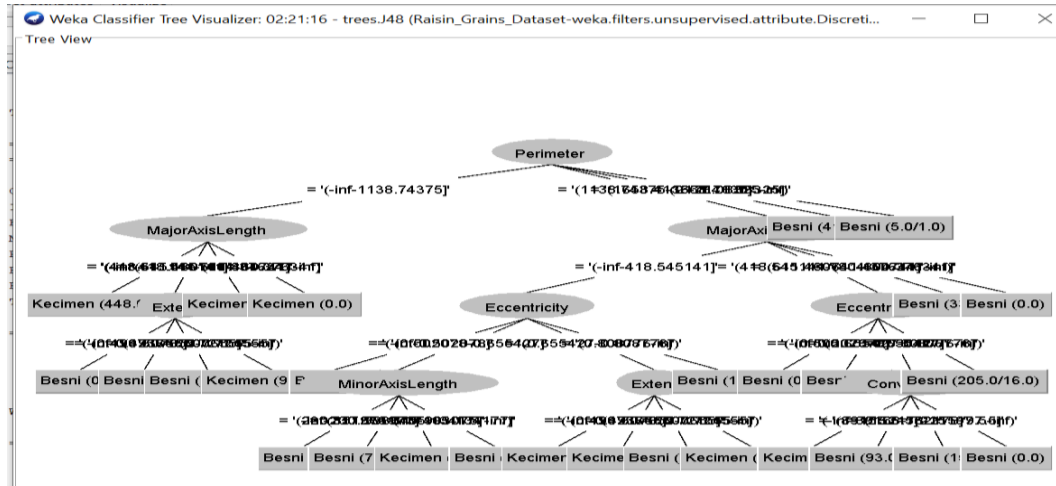


Figure 16: Tree

```

Classifier output

Time taken to build model: 0.11 seconds

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

Correctly Classified Instances      784              87.1111 %
Incorrectly Classified Instances    116              12.8889 %
Kappa statistic                    0.7422
Mean absolute error                 0.2081
Root mean squared error             0.3292
Relative absolute error              41.6221 %
Root relative squared error         65.8351 %
Total Number of Instances          900

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0.896    0.153    0.854      0.896    0.874      0.743    0.884     0.830    Kecimen
                0.847    0.104    0.890      0.847    0.868      0.743    0.884     0.884    Besni
Weighted Avg.   0.871    0.129    0.872      0.871    0.871      0.743    0.884     0.857

=== Confusion Matrix ===

  a  b  <-- classified as
403 47 |  a = Kecimen
 69 381 |  b = Besni

```

Figure 17: Classifier Output

- The correctly classified instances are **87.11%**.
- From confusion matrix part, **TP = 403, FN = 47, FP = 69** and **TN = 381**. As the precision is **0.854**, recall **0.896**, F-Measure = **0.874**. These results have the impact of improving the behavior a little. Increasing in confidence level gave a better result.

Classification	F-Measure
Discretization with bins=10	0.852
Discretization with bins=2	0.753
with bins=4, confidence factor= 0.25	0.869
with bins=4, confidence factor= 0.75	0.874

The best result obtained in the last scenario.

## Naïve Bayes Discretization

The data was filtered ‘proposed’, by discretize all continuous attributes to discrete values with two different ranges.

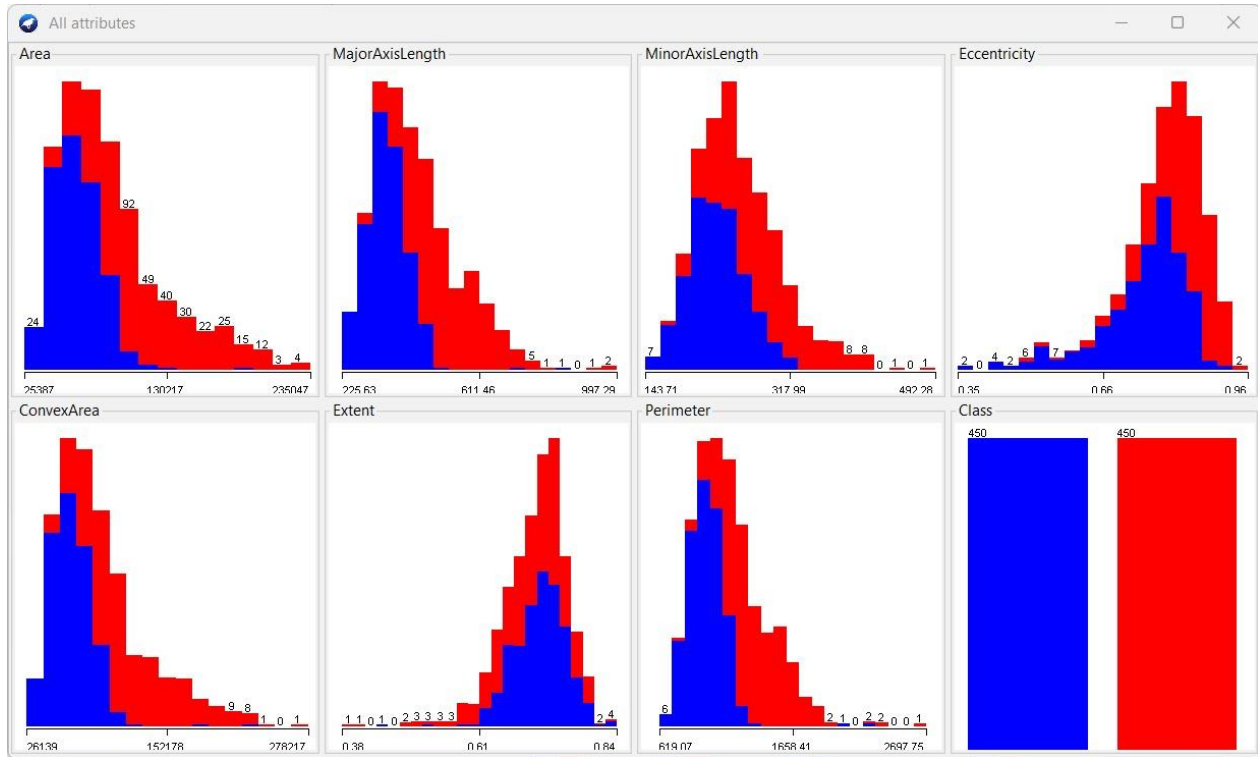


Figure 18: Attributes before discretizing

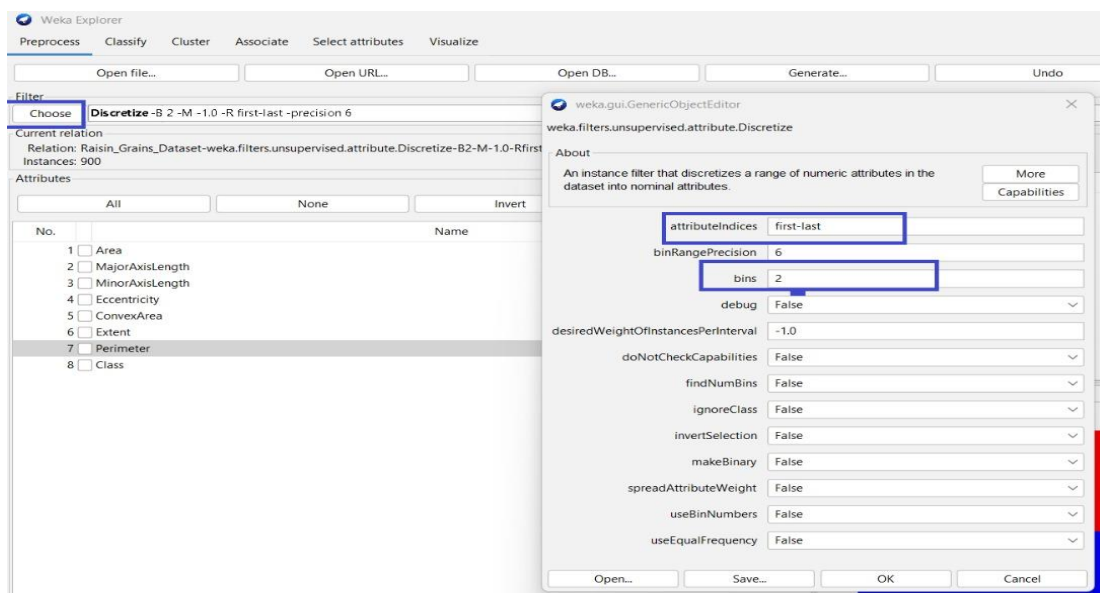


Figure 19: Settings for Prepressing

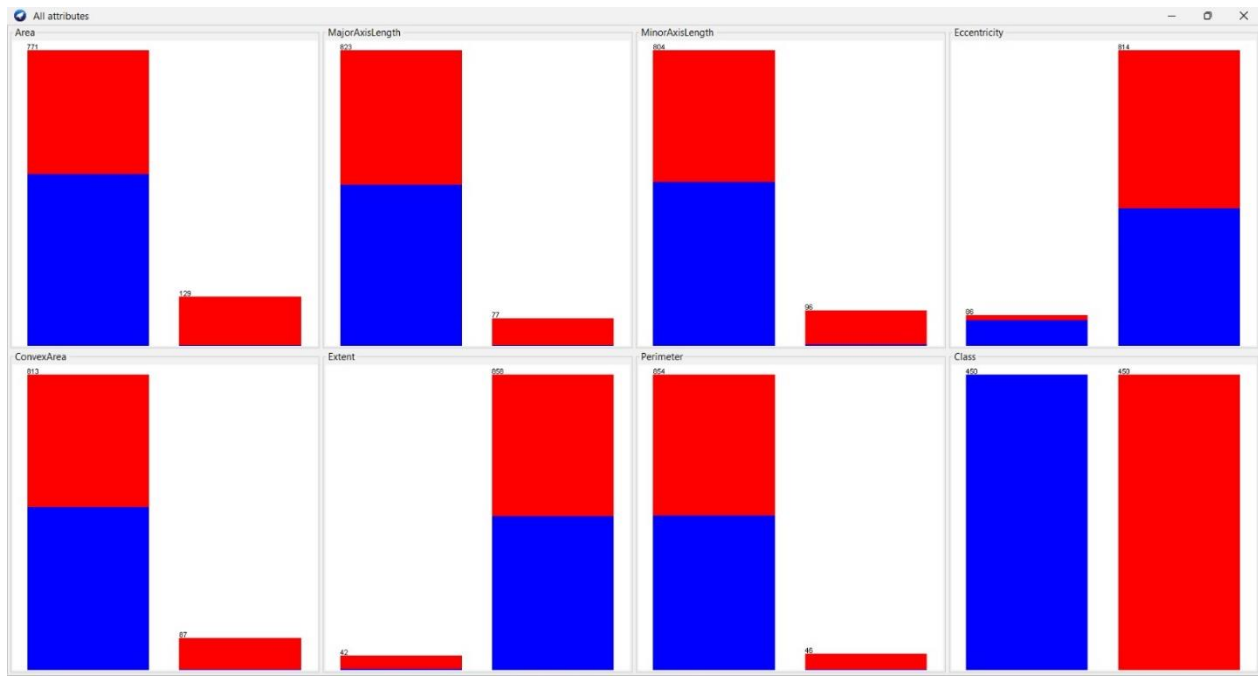


Figure 20: Attributes after discretizing

From choose option in Weka software, the discretization method was chosen, then all attributes were selected to be discretized into two ranges.

## 5-Fold Cross Validation

Using the 5-fold cross validation to train the model, and applying Naïve Bayes as a classifier, the result of cross matrix, accuracy, recall, precision, and f-source were calculated as shown in figure 21.

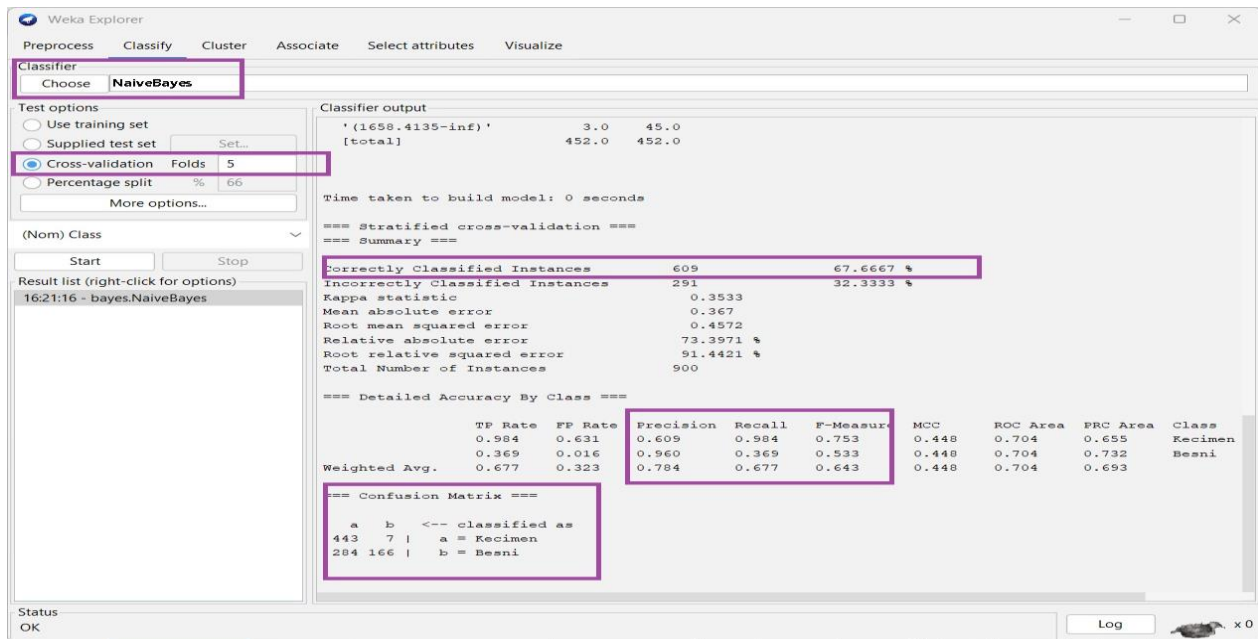


Figure 21: The result of using Naive Bayes as Classifier

- From figure21, from confusion matrix part. The matrix illustrates that, **TP =443**, **FN =7**, **FP =284** and **TN = 166**. From these values, precision, recall and f- measure values can be calculated. The value of the **precision is 0.609** , **Recall is 0.984** and **f-measure = 0.753**. Also, from the figure the **accuracy** was equals **67.667%**, which is not that good.

### 3. Changing hyper-parameters

Applying changing on batch size from 100 to 900 , num Decimal Places from 2 to 6 and turn on the use kernel Estimator.

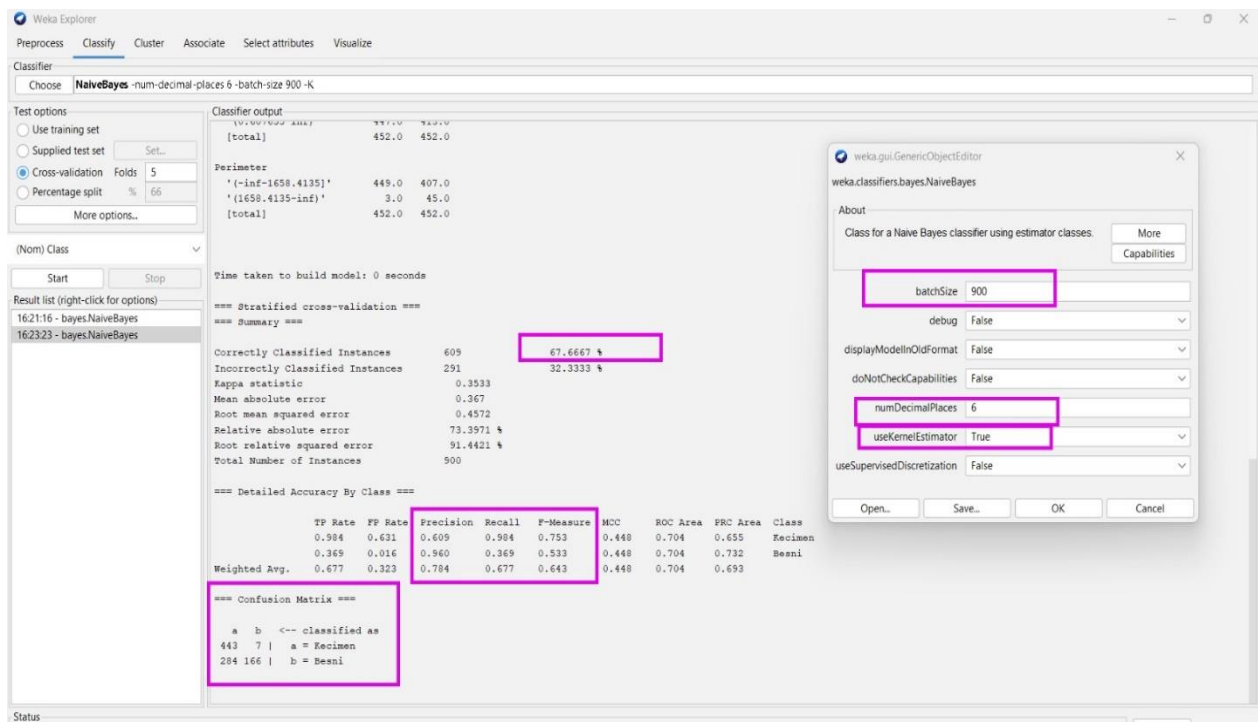


Figure 22: Naive Bayes results after changing Hyper Parameters

Notice from figure 22, the results are the same as Figure 21 results. No change the same accuracy and the other values.



## Multi Layer Perceptron (MLP)

### Discretization

The data was filtered ‘proposed’, by discretize all continuous attributes to discrete values with two different ranges.

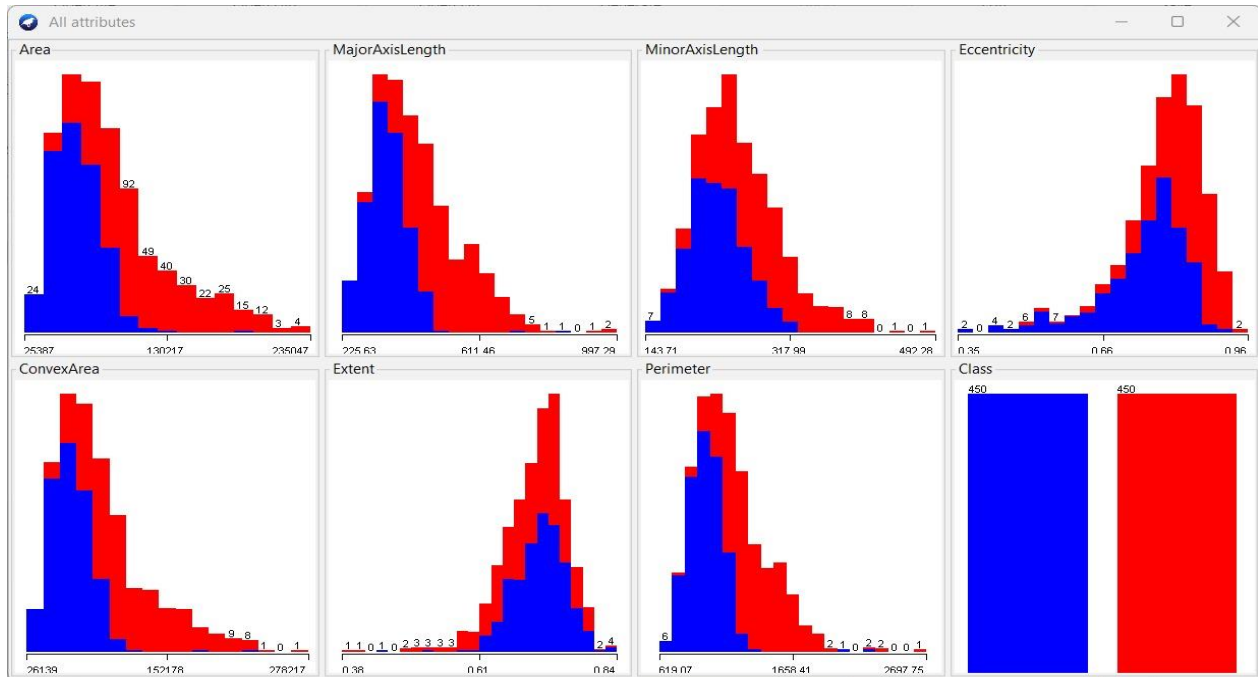


Figure 23: Attributes before discretization

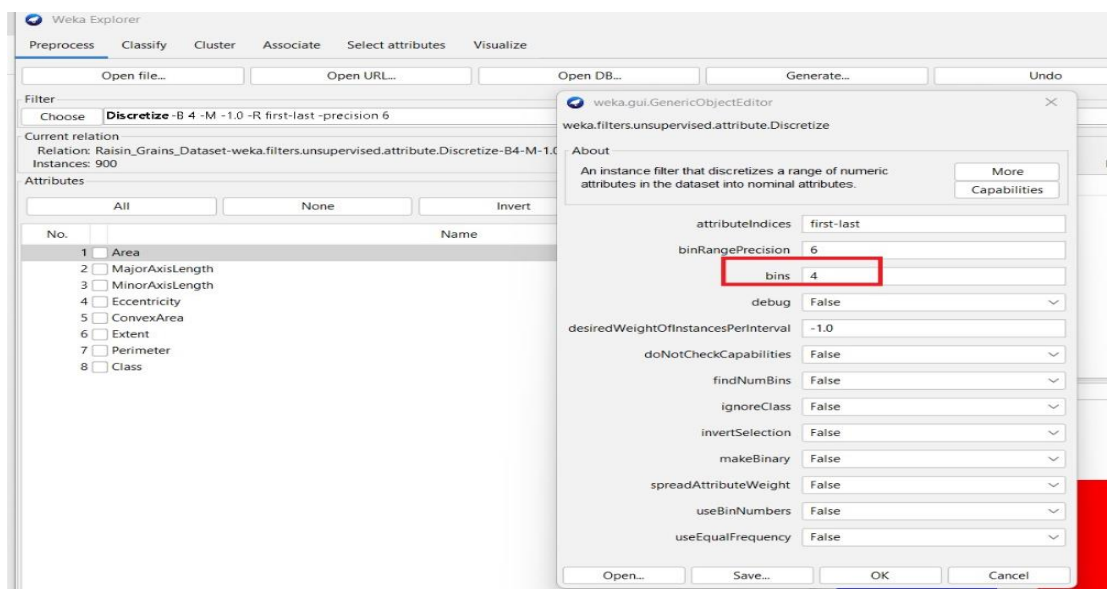


Figure 24: Settings for Prepressing

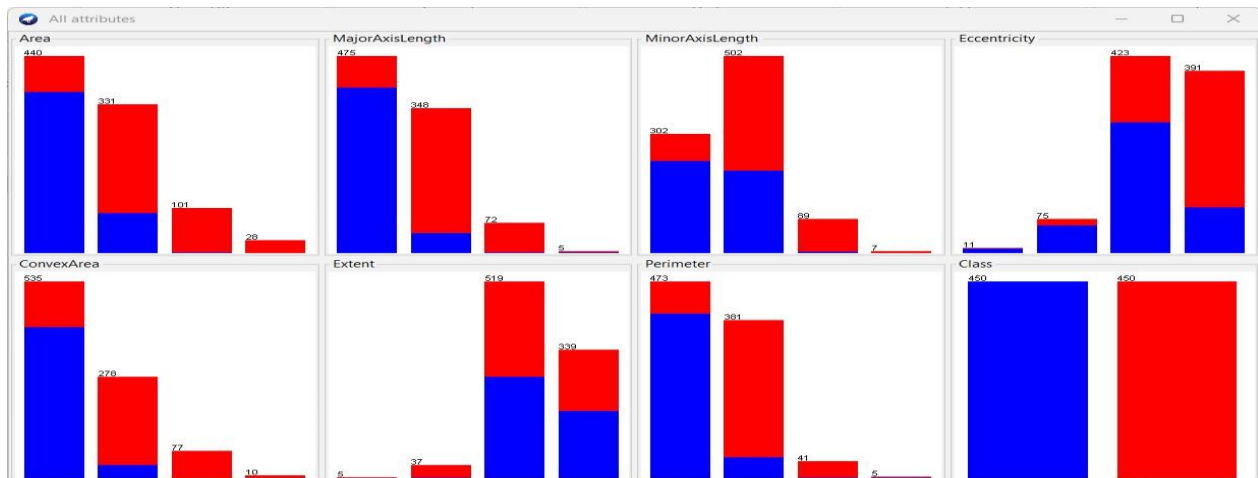


Figure 25: Attributes after discretization

## 5-Fold Cross Validation

Using the 5-fold cross validation to train the model, and applying Multi Layer Perceptron (MLP) as a classifier, the result of cross matrix, accuracy, recall, precision, and f-source were calculated.

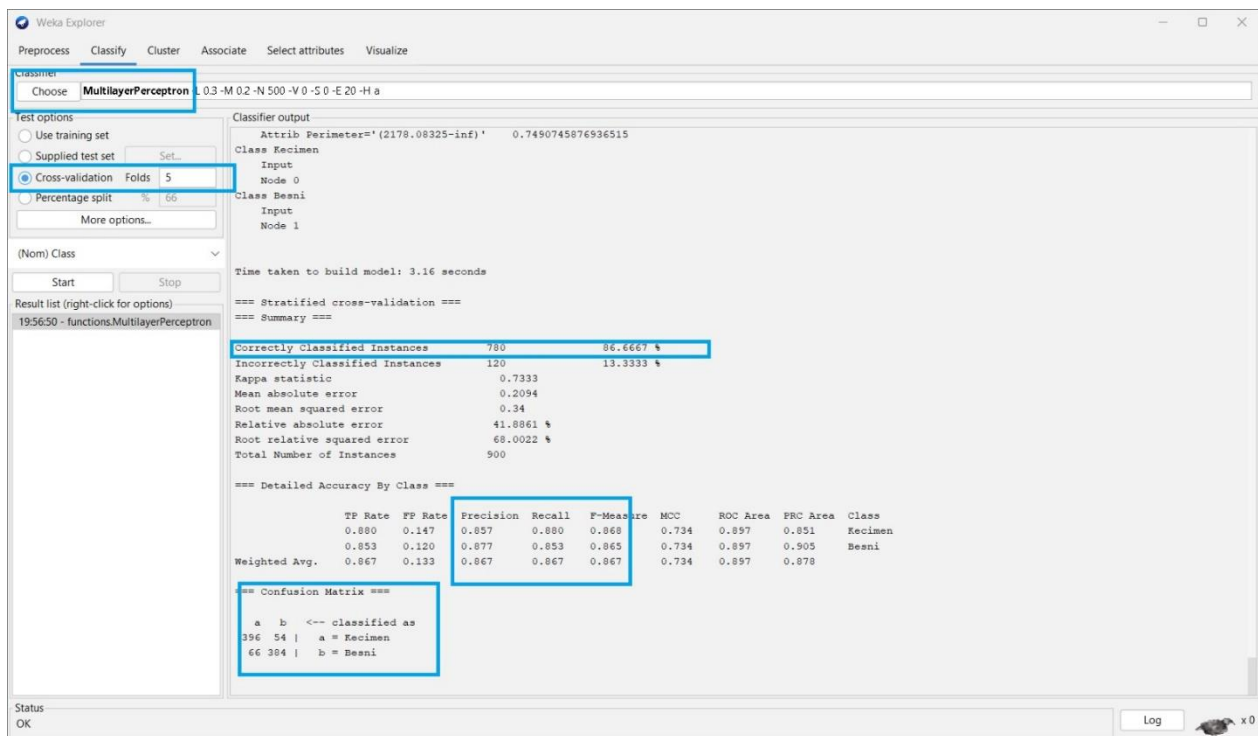


Figure 26: The result of using Multi Layer Perceptron (MLP) as Classifier

From figure 26, from confusion matrix part. The matrix illustrates that, **TP =396**, **FN =54**, **FP= 66** and **TN = 384**. From these values, precision, recall and f- measure values can be calculated. The value of the **precision is 0.857**, **Recall is 0.880** and **f-measure = 0.868**. Also, from the figure **the accuracy was equals 86.67%**, which is very good.

## Changing hyper-parameters

Changing the learning rate in Multi Layer Perceptron (MLP) from 0.3 to 0.1 and speed from 0 to 1 and training time from 500 to 1000.

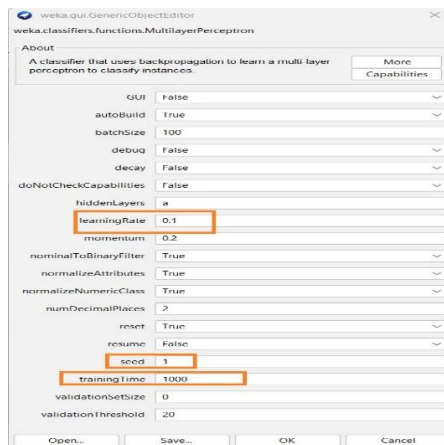


Figure 27: Changing hyper-parameter for MLP

The new Confusion matrix is

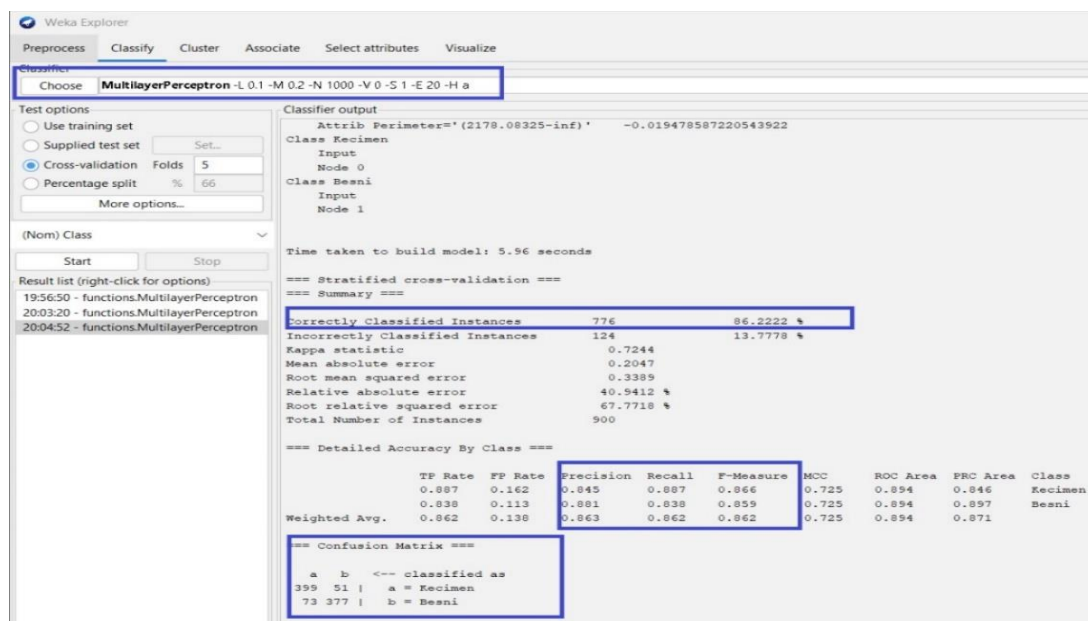


Figure 28: Confusion Matrix after Changing hyper-parameter for MLP

From figure 28, from confusion matrix part. The matrix illustrates that, **TP =399, FN =51, FP = 73** and **TN = 377**. From these values, precision, recall and f- measure values can be calculated. The value of **the precision is 0.845** , **Recall is 0.887** and **f-measure = 0.866**. Also, from the figure **the accuracy was equals 86.222%** , which is good.

## Conclusion

From the obtained results, Some Notes were taken, for the first part when the Decision Tree algorithm was applied, the accuracy increased when the number of bins are four and when the confidence factor is equal to 0.75. That test in this project has the **best accuracy by 87.11%** and with F-measure equals **0.874**. In the Naïve Bayes algorithm, the results were different. They were the worst, as the accuracy equals **67.667%** and the F-measure equals **0.753**. In the last algorithm Multi Layer Perceptron (MLP), the results were good, since the accuracy was equal to **86.67%** and the F-measure to **0.868**, even after playing with the parameters, the results were still good.