

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Artificial Intelligence ENCS 3340

Second Semester, 2021/2022

**PROJECT#2 Machine Learning for Classification**

**\*\*\*\*\*\*\*\***

**Prepared by: Osama Qutait**

***Student’s No.:* *1191072***

**Partner's name: Mahmoud Samara**

**Partner's *No.:* *1191602***

***Instructor:* *Dr.Adnan Yahya***

**Date: 12-6-2022**

Table of Contents

[1. Abstract 2](#_Toc105955963)

[2. Data Set 2](#_Toc105955964)

[3. First model (Decision tree) 3](#_Toc105955965)

[4. Second model (Naïve Bayes) 6](#_Toc105955966)

[5. Third model (multilayer perceptron) 8](#_Toc105955967)

[6. Conclusion 10](#_Toc105955968)

**List of Figures**

[**Figure ‎2‑1 Raisin Dataset** 2](#_Toc105954515)

[**Figure ‎2‑2 distrubtion for all attributes** 2](#_Toc105954516)

[**Figure ‎2‑3 Attribute Class information**...............................................................................................3 **Figure ‎2‑4 Attribute area information** 3](#_Toc105954517)

[**Figure ‎3‑1 new distribution for Area**...................................................................................................4 **Figure ‎3‑2 new distribution for Extent** 3](#_Toc105954518)

[**Figure ‎3‑3 Summary of Decision Tree** 4](file:///C:\Users\USER\Desktop\ENCS3340_PROJECT2_REPORT-1.docx#_Toc105954519)

[**Figure ‎3‑4 The Decision Tree** 4](file:///C:\Users\USER\Desktop\ENCS3340_PROJECT2_REPORT-1.docx#_Toc105954520)

[**Figure ‎3‑5 Summary of the newDecision Tree** 5](#_Toc105954521)

[**Figure ‎3‑6 New Decision Tree** 5](#_Toc105954522)

[**Figure ‎4‑1 new distribution for Perimeter** 6](#_Toc105954523)

[**Figure ‎4‑2 Classifier output** 6](#_Toc105954524)

[**Figure ‎4‑3 Naïve Bayes curve** 7](#_Toc105954525)

[**Figure ‎4‑4 Classifier output** 7](#_Toc105954526)

[**Figure ‎4‑5 New Naïve Bayes curve** 7](#_Toc105954527)

[**Figure ‎5‑1 new distribution for Area**...................................................................................................9 **Figure ‎5‑2 new distribution for ConvexArea** 8](#_Toc105954528)

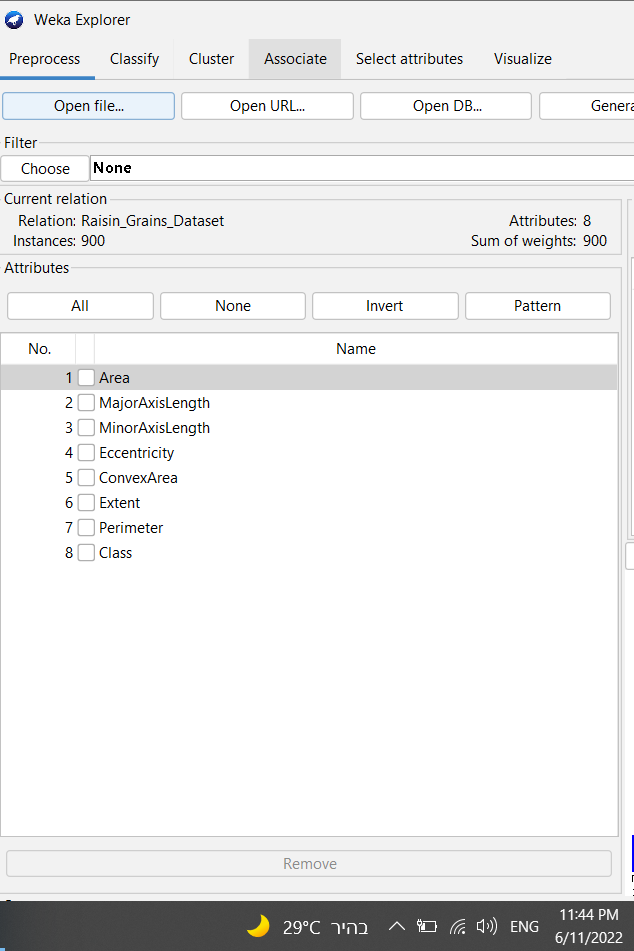
[**Figure 5‑3 Summary of multilayer perceptron** 8](#_Toc105954529)

# Abstract

First of all, this will be an implementation of Machine Learning for Classification. We will compare different machine learning algorithms for a classification task and test 3 different models. We will use WEKA3.8.6 program to processing, classifying and simulate project results and get the values for the project from (Raisin Dataset.affr) file that have taken from the link. Since the team numbers are 1191072, 1191602, So 2 mode 3 = 2 then we will take group number 2 (Raisin Dataset). We will make the following models:

1. Decision Tree
2. Naïve Bayes
3. Multilayer perceptron

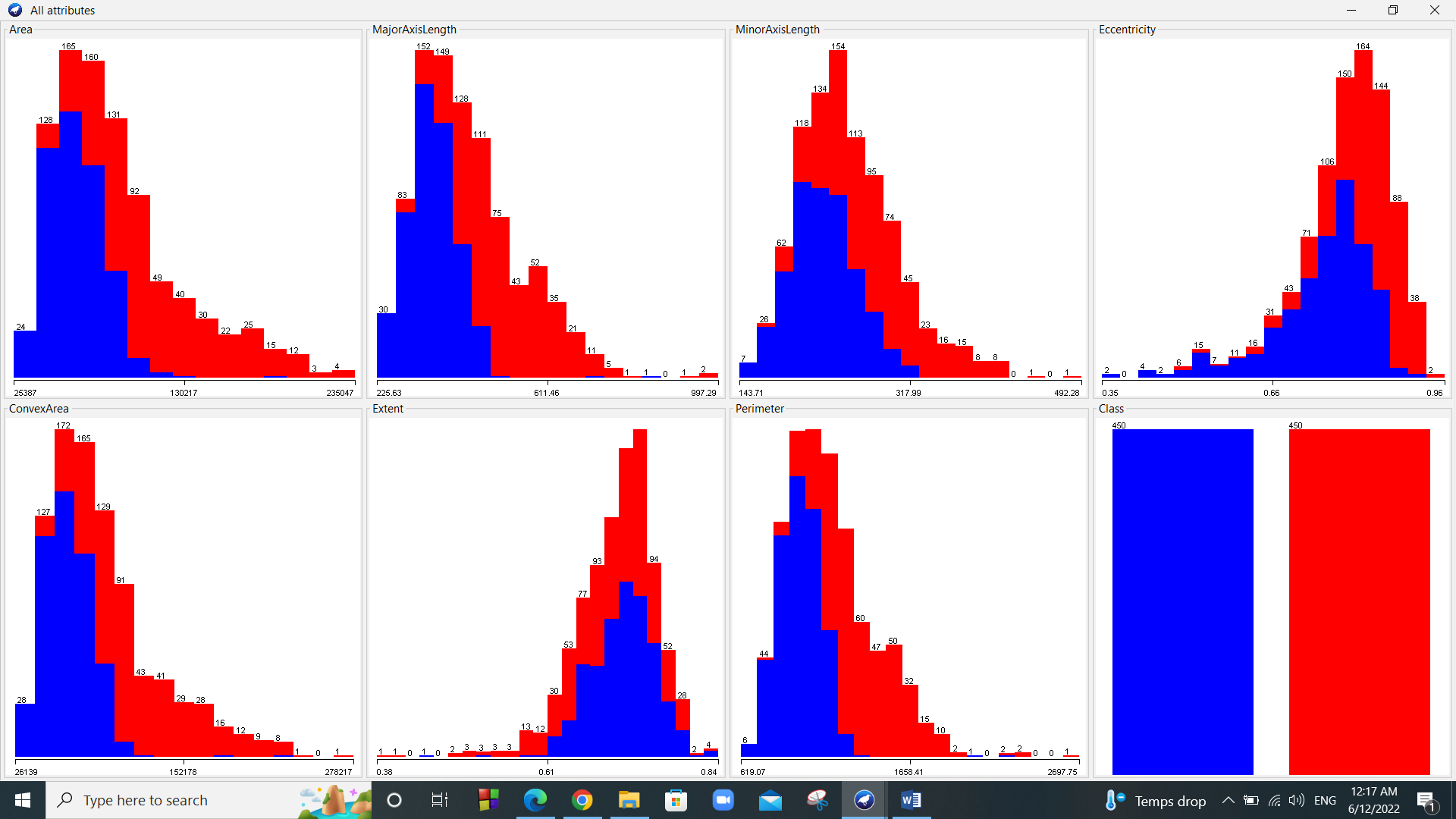
# Data Set



**Figure ‎2‑1 Raisin Dataset**

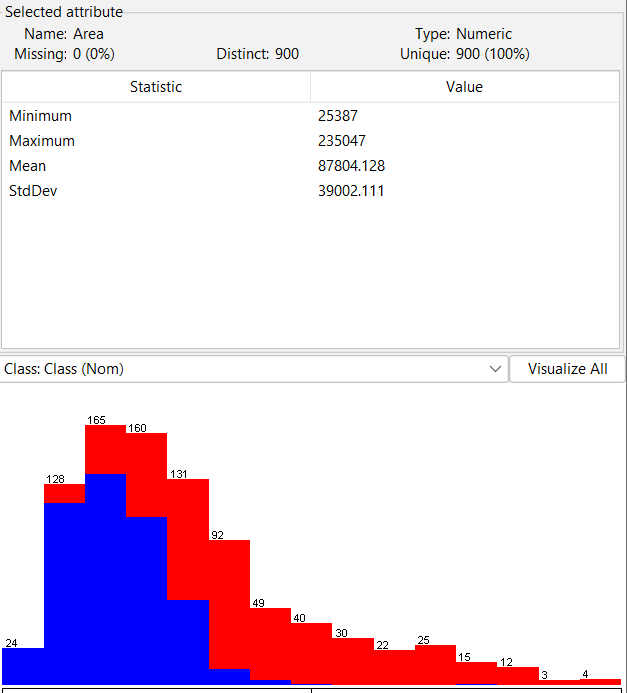
As we can see from the previous figure first we have open Raisin Dataset.ARFF file in WEKA program to read it. Now the number of instances and attributes has been initialized and were recorded in the screen shot, so number of instances is 900 and the number of attributes is 8 that are: 1.Area, 2.MajorAxisLength, 3.MinorAxisLength, 4.Eccentricity, 5.ConvexArea, 6.Extent 7.Perimeter (these 7 attributes have Numeric type (continuous) No.8.Class has Nominal type (discrete).

See the following figure that shows the classification for the different data for each attribute:



**Figure ‎2‑2 distrubtion for all attributes**

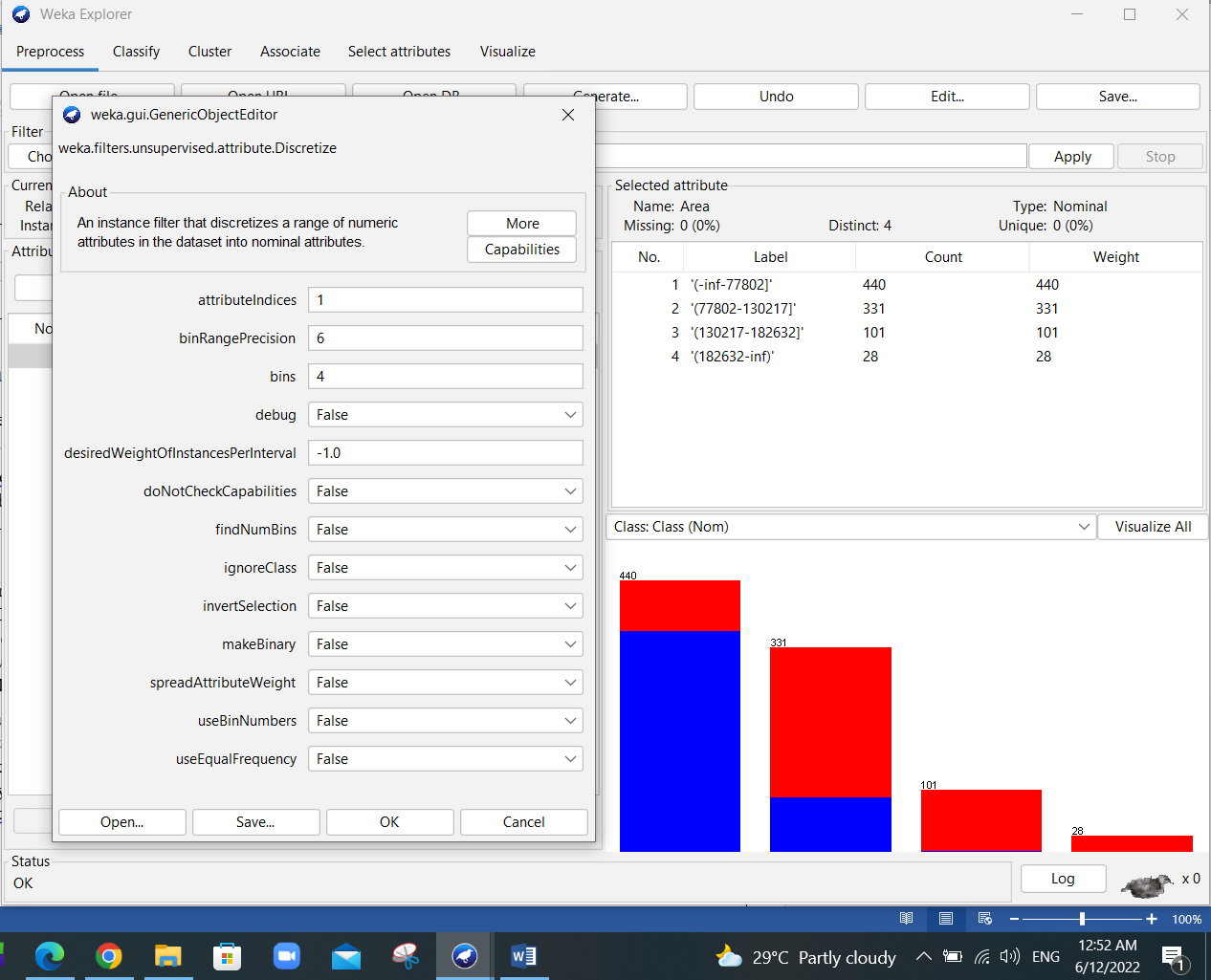
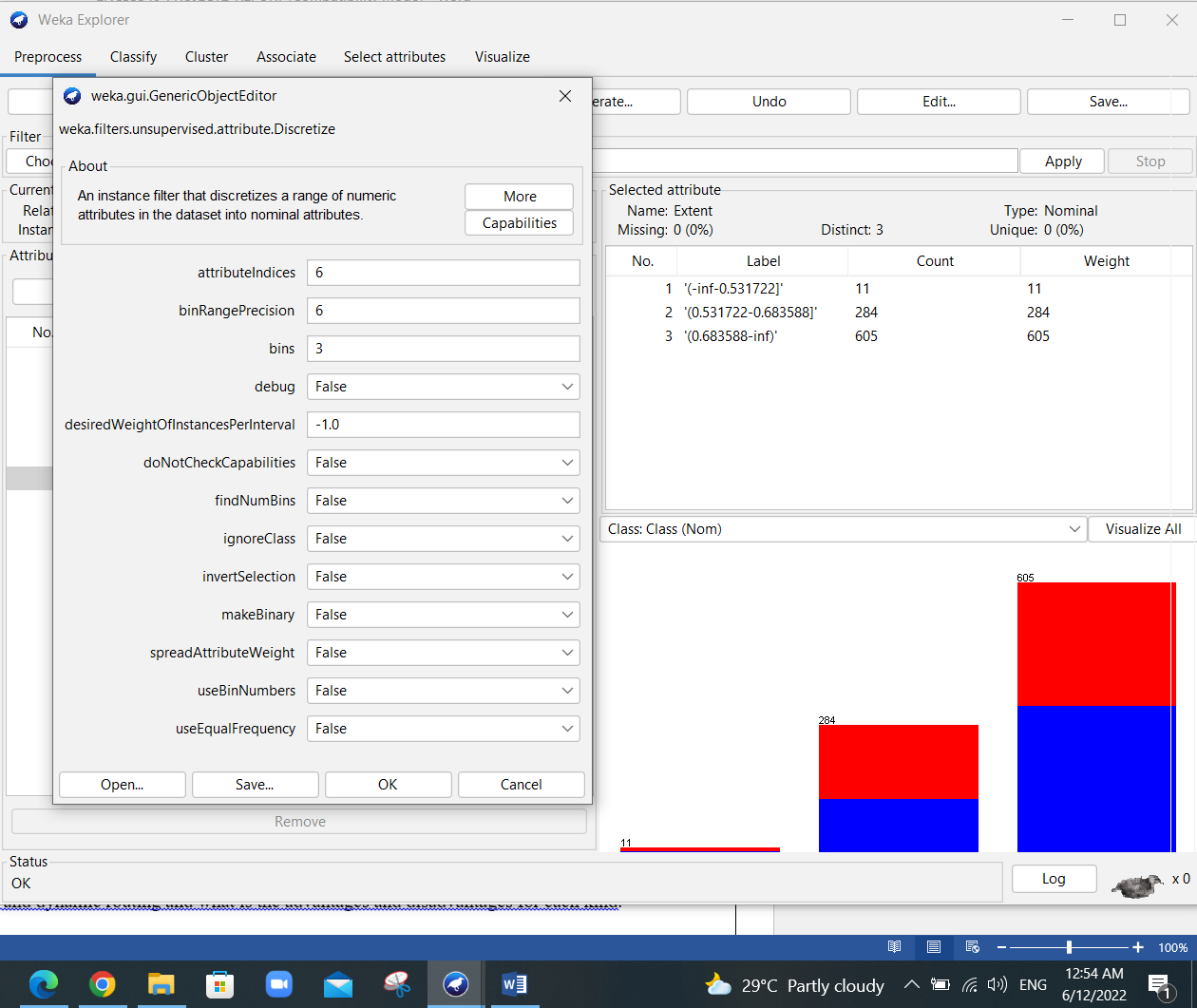
Also the preprocess page can show the details for each attribute for example: Attribute Area since it is continuous so it will show the different statics like (minimum value, maximum value, Mean and Standard deviation). On the other hand, attribute class it is discrete so it will show the kinds of labels it contains and the weight for each label from the total number.

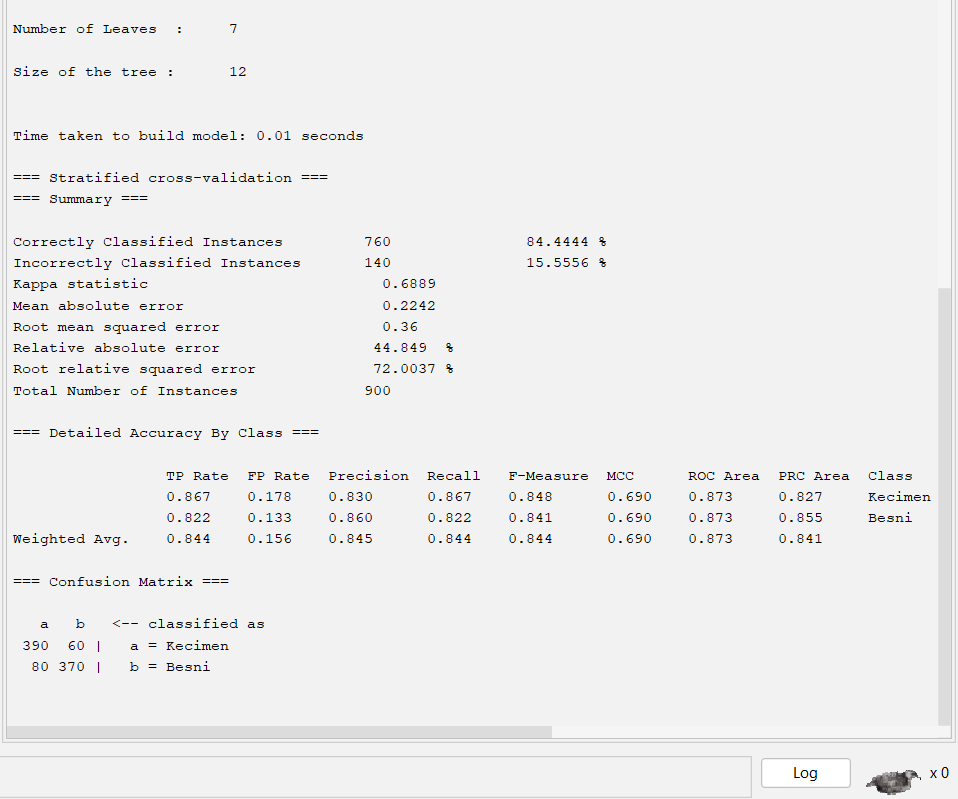
**Figure ‎2‑3 Attribute Class information Figure ‎2‑4 Attribute area information**

# First model (Decision tree)

Before starting our work in the decision tree model I will make discretization of two continuous attributes that are (Area and Extent) as the following: From Filter we will choose Discretize then I will choose which attribute I will change and the new number of bins. The following figures shows new distribution for Area and Extent after making discretization so the changed from continuous to discrete(Nominal type).

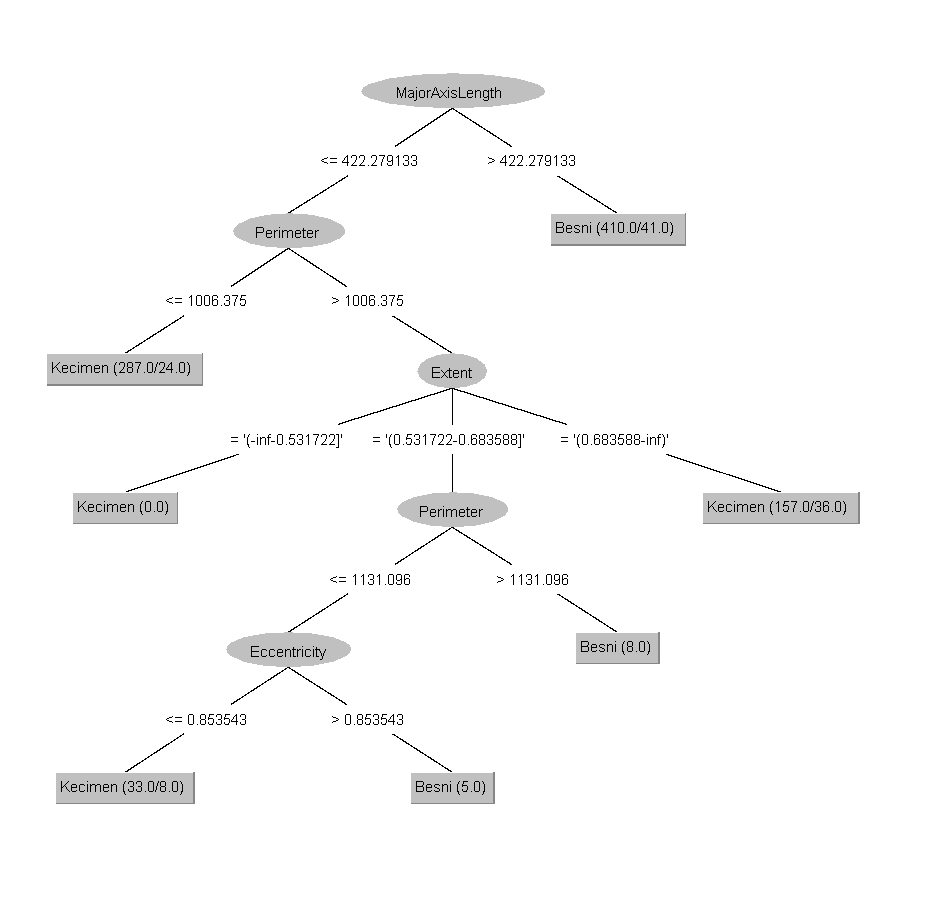
 

**Figure ‎3‑1 new distribution for Area Figure ‎3‑2 new distribution for Extent**

We applied the decision tree as the first algorithm for classifying the data set. We chose 5-fold cross validation as a test option:

From the figure 3-3 we see that the total number of leaves was 7 and the size of the tree is 12. In addition, we see that from the test we got 760 instances are correctly classified with a percentage of accuracy of 84.4444% and with 140 incorrectly classified instances with a percentage of accuracy of 15.5556%. Moreover, we see that the Confusion matrix results are initialized as the Class option results, for example Kecimen (a) is true so 390 instances out of 450 is true with rate of 0.876 and this rate is the true positive rate for Keciman and the False Positive is 80 with a rate of 0.178. The precision, recall and F-measure with their weighted averages were shown in the figure. The same thing for Besni with different numbers and rates.

**Figure ‎3‑3 Summary of Decision Tree**

For The precision we can calculate it as: **Precision = TruePositives / (TruePositives + FalsePositives),** for example for Kecimen Precision it is equal: (0.867 / (0.867+0.178)) = 0.830

For The Recall we can calculate it as: **Recall = TruePositives / (TruePositives + FalseNegatives)** for example forKecimen Recall it is equal: (0.867 / (0.867+0.133)) = 0.867

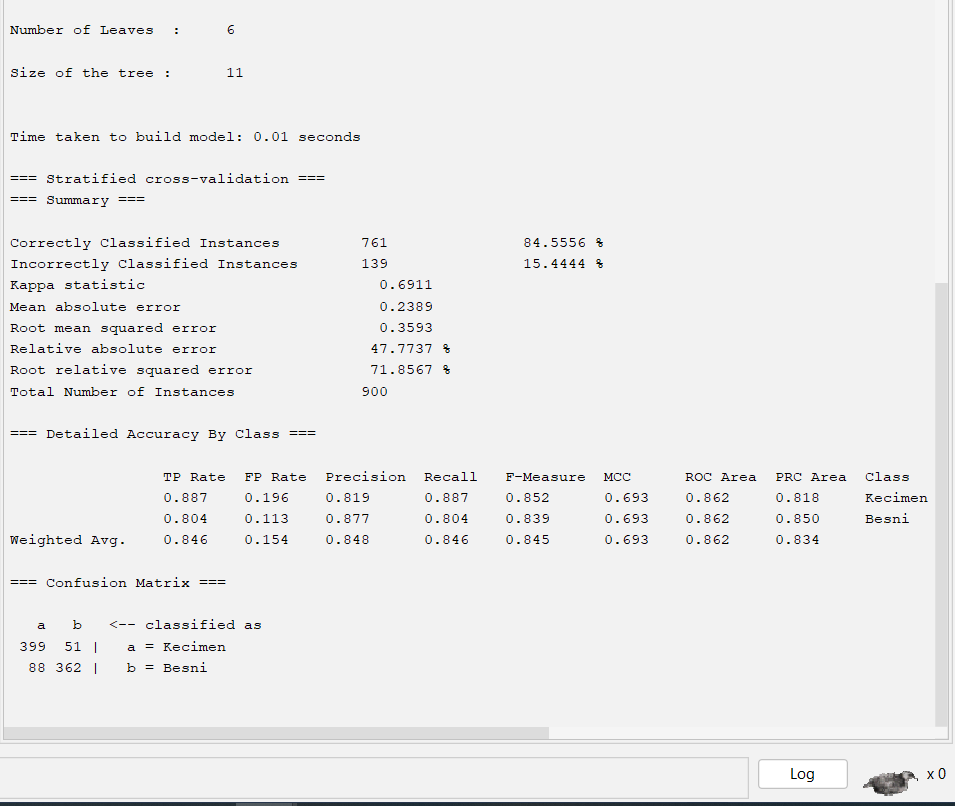
For The F-Measure we can calculate it as: **F-Measure = (2 \* Precision \* Recall) / (Precision + Recall)** for example forKecimen F-Measure it is equal: (2\*0.830\*0.867 / (0.867+0.830)) = 0.848

The report shows also all this measures and calculations and confusion matrix for Besni with different numbers and rates.

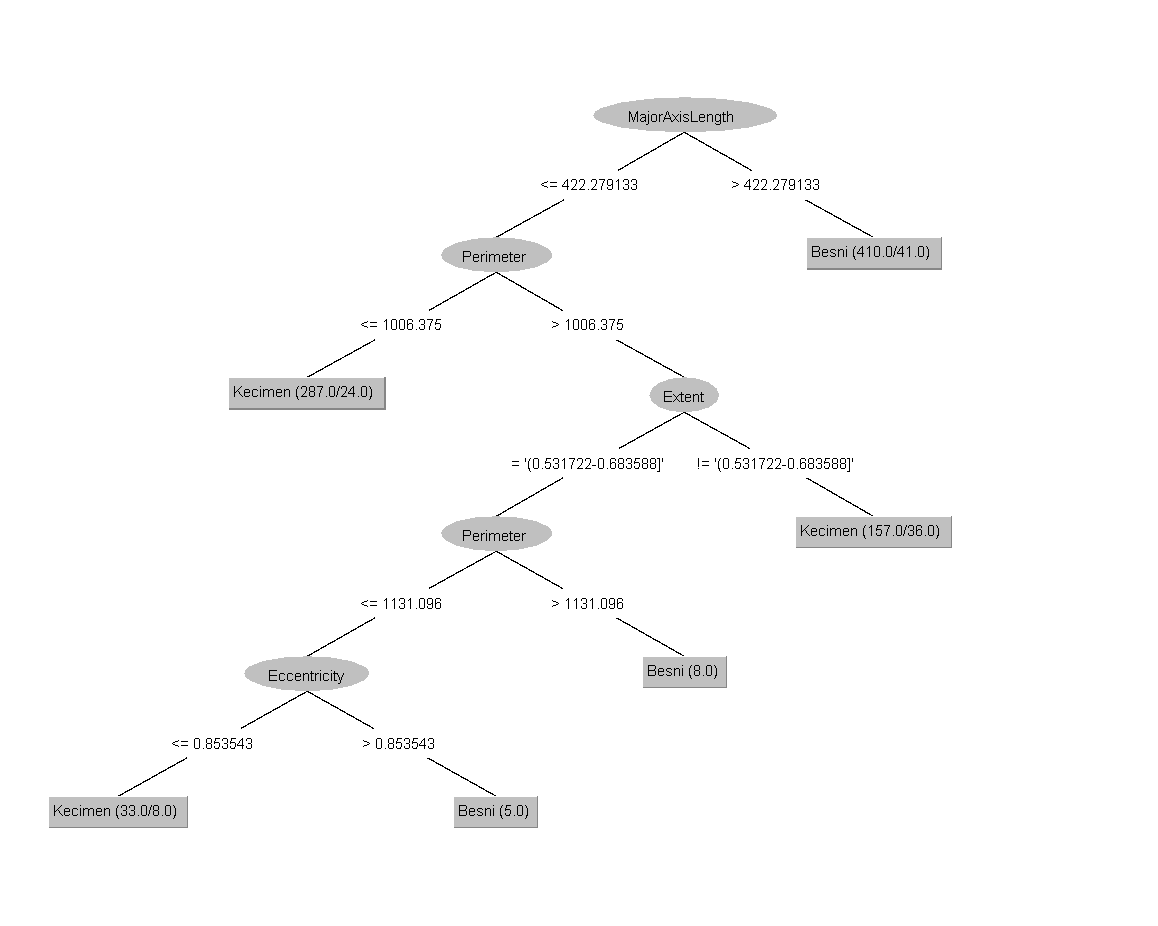
See figure 3-4 that shows the decision tree for our test and simulation with the number of leaves was 7 and the size of the tree is 12

**Figure ‎3‑4 The Decision Tree**

Now we will change the hyper-parameter for example change binary split to be true and the confidence factor from 0.25 to 0.1. So, the number of leaves, size of tree, precision, recall, F-measure with their weighted averages and percentage of correct/incorrect classification were also affected but the calculation rules will remain the same as shown in the figure below:



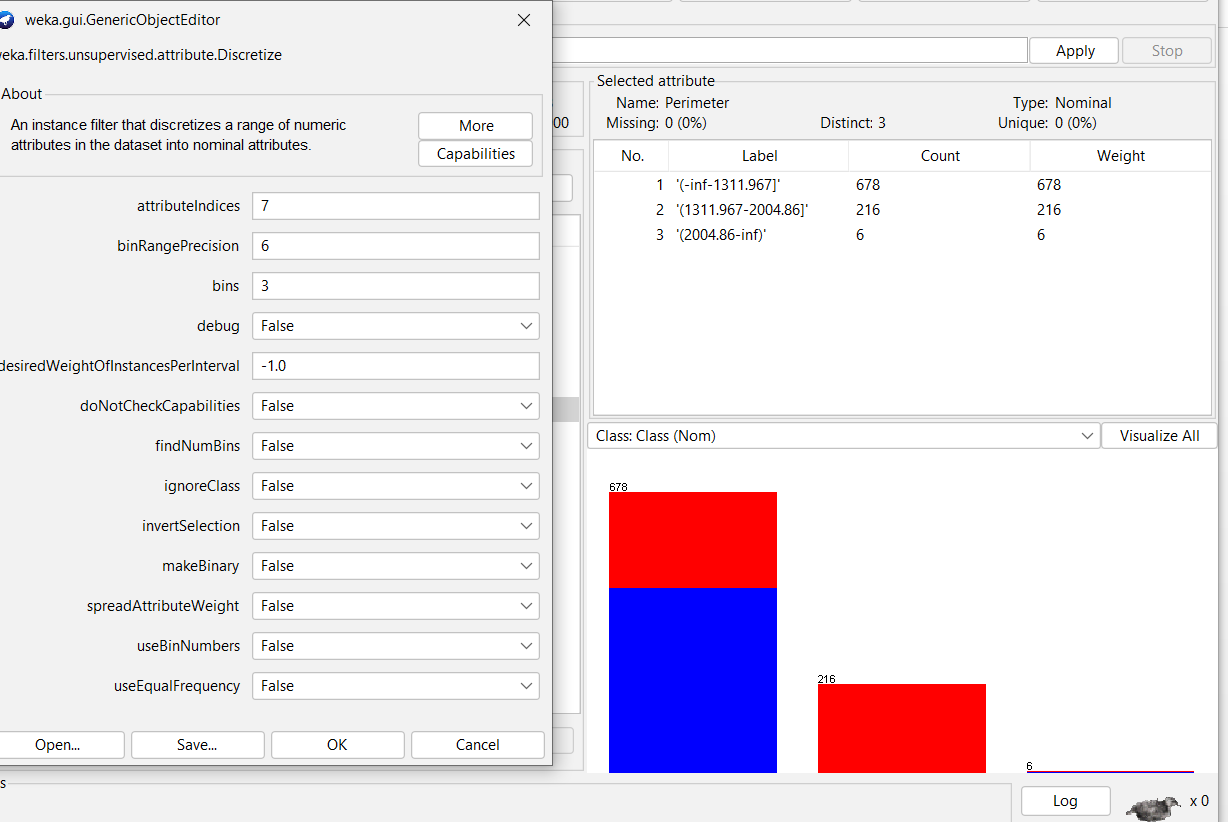
**Figure ‎3‑5 Summary of the newDecision Tree**



**Figure ‎3‑6 New Decision Tree**

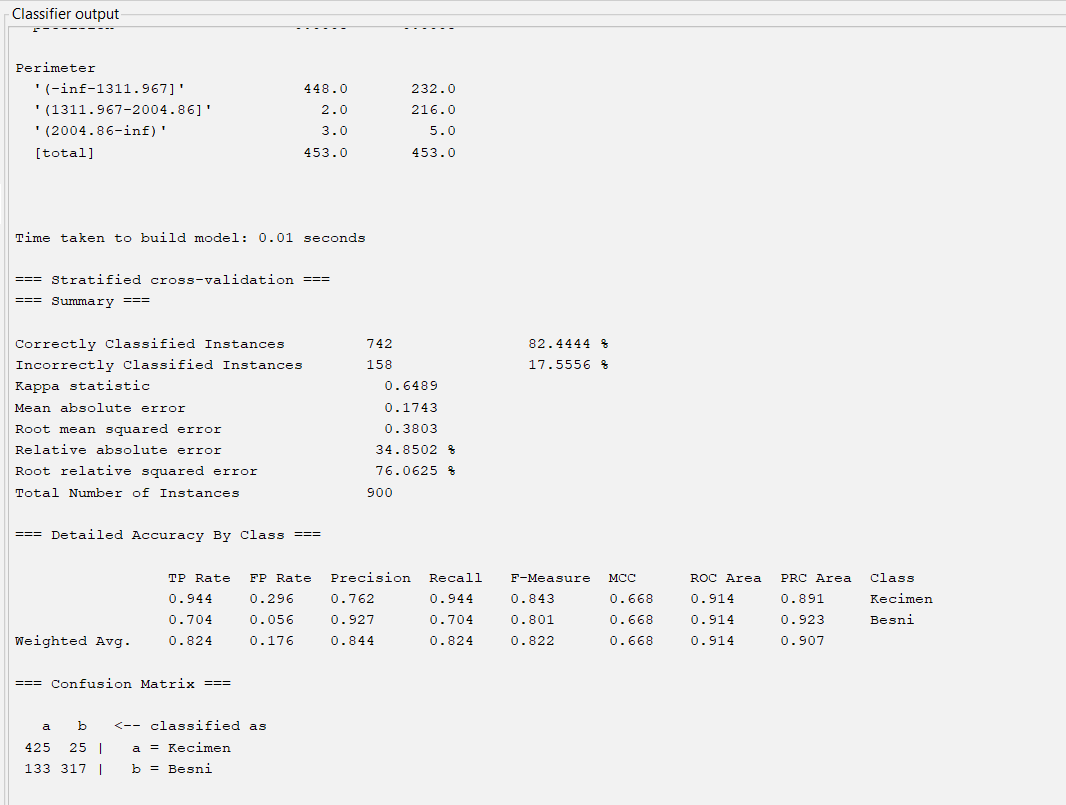
# Second model (Naïve Bayes)

I will make discretization to Perimeter attribute as what I did in the first model. The following figure shows new distribution after making discretization so it changed to discrete(Nominal type).



**Figure ‎4‑1 new distribution for Perimeter**

We applied the Naïve Bayes as the second algorithm for classifying the data set. We chose 5-fold cross validation as a test option:



**Figure ‎4‑2 Classifier output**

From the figure 4-2 we see It produces 5 equal sized sets. In addition, we see that from the test we got 742 instances are correctly classified with a percentage of accuracy of 82.4444% and with 158 incorrectly classified instances with a percentage of accuracy of 17.5556%. Moreover, we see that the Confusion matrix results are initialized as the Class option results, for example Kecimen (a) is true so 425 instances out of 450 is true with rate of 0.944 and this rate is the true positive rate for Keciman and the False Positive is 25 with a rate of 0.296. The precision, recall and F-measure with their weighted averages were shown in the figure. The same thing for Besni with different numbers and rates.

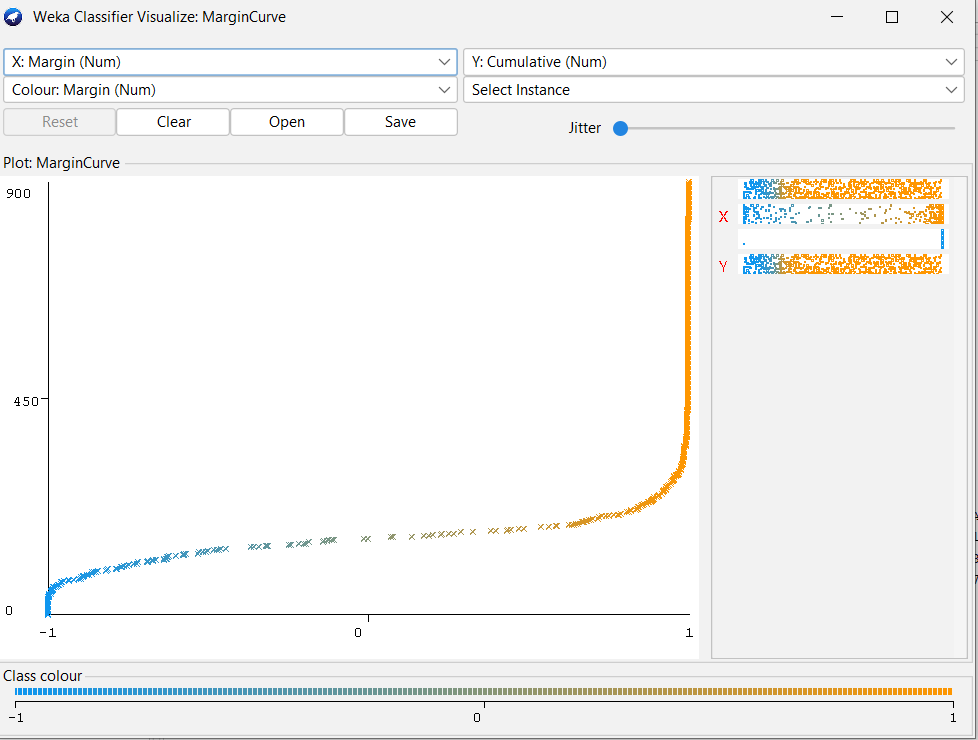
For The precision we can calculate it as: **Precision = TruePositives / (TruePositives + FalsePositives),** for example for Kecimen Precision it is equal: (0.944 / (0.944+0.296)) = 0.762

For The Recall we can calculate it as: **Recall = TruePositives / (TruePositives + FalseNegatives)** for example forKecimen Recall it is equal: (0.867 / (0.867+0.133)) = 0.944

For The F-Measure we can calculate it as: **F-Measure = (2 \* Precision \* Recall) / (Precision + Recall)** for example forKecimen F-Measure it is equal: (2\*0.944\*0.762 / (0.944+0.762)) = 0.843

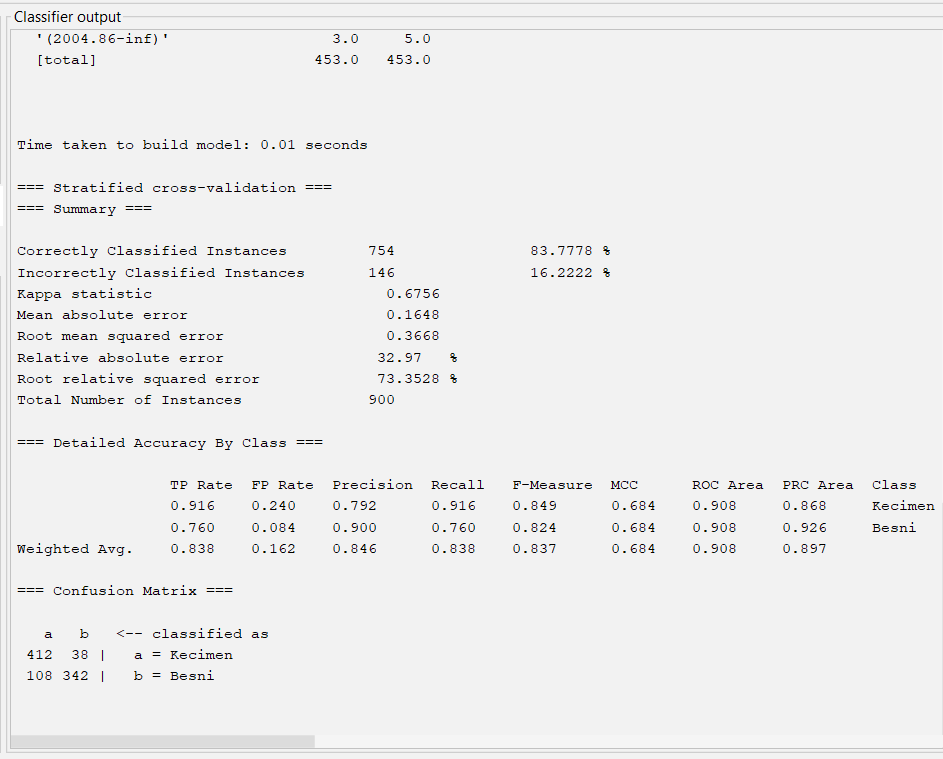
The report shows also all this measures and calculations and confusion matrix for Besni with different numbers and rates.

See figure 4-3 that shows the Naïve Bayes curve:

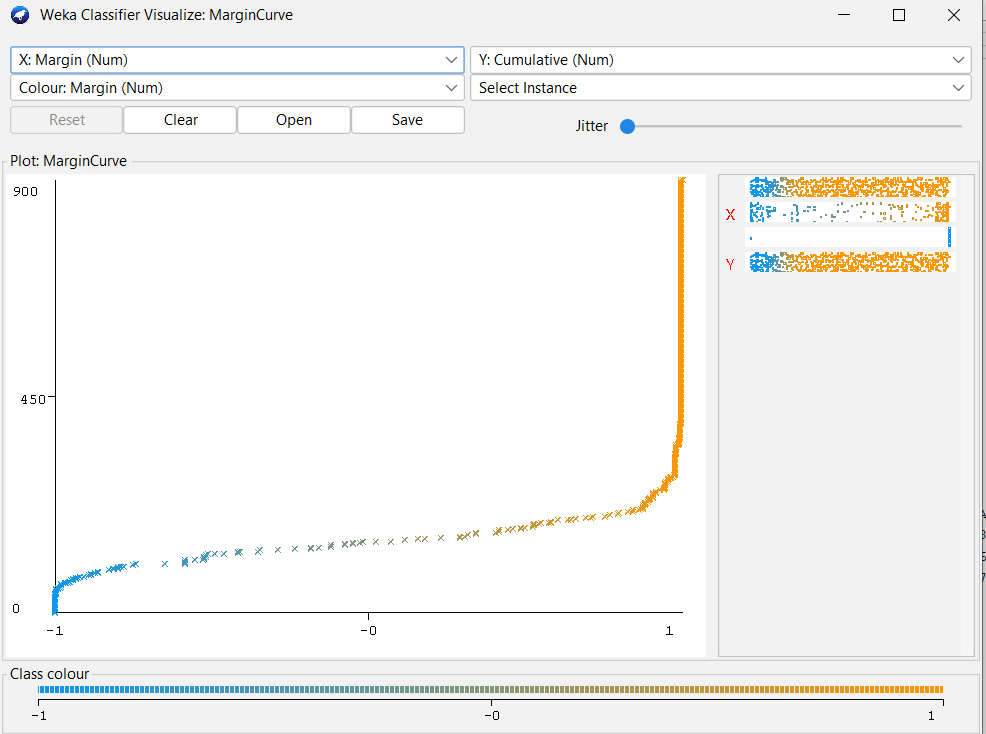


**Figure ‎4‑3 Naïve Bayes curve**

Now we will change the num of decimal places from 2 to 5, batch size from 100 to 200. So, the precision, recall, F-measure with their weighted averages and percentage of correct/incorrect classification were also affected but the calculation rules will remain the same as shown in the figure below:



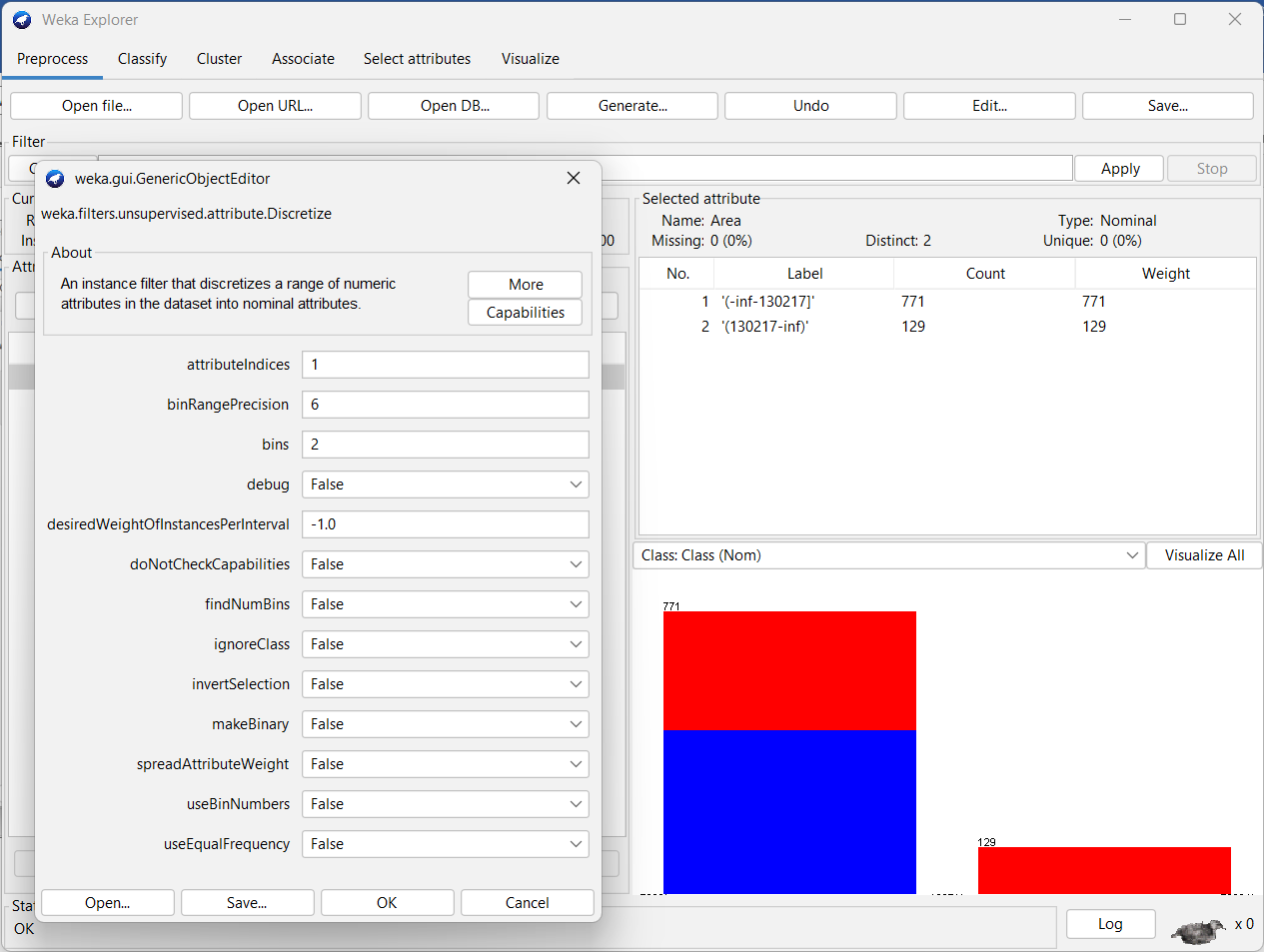
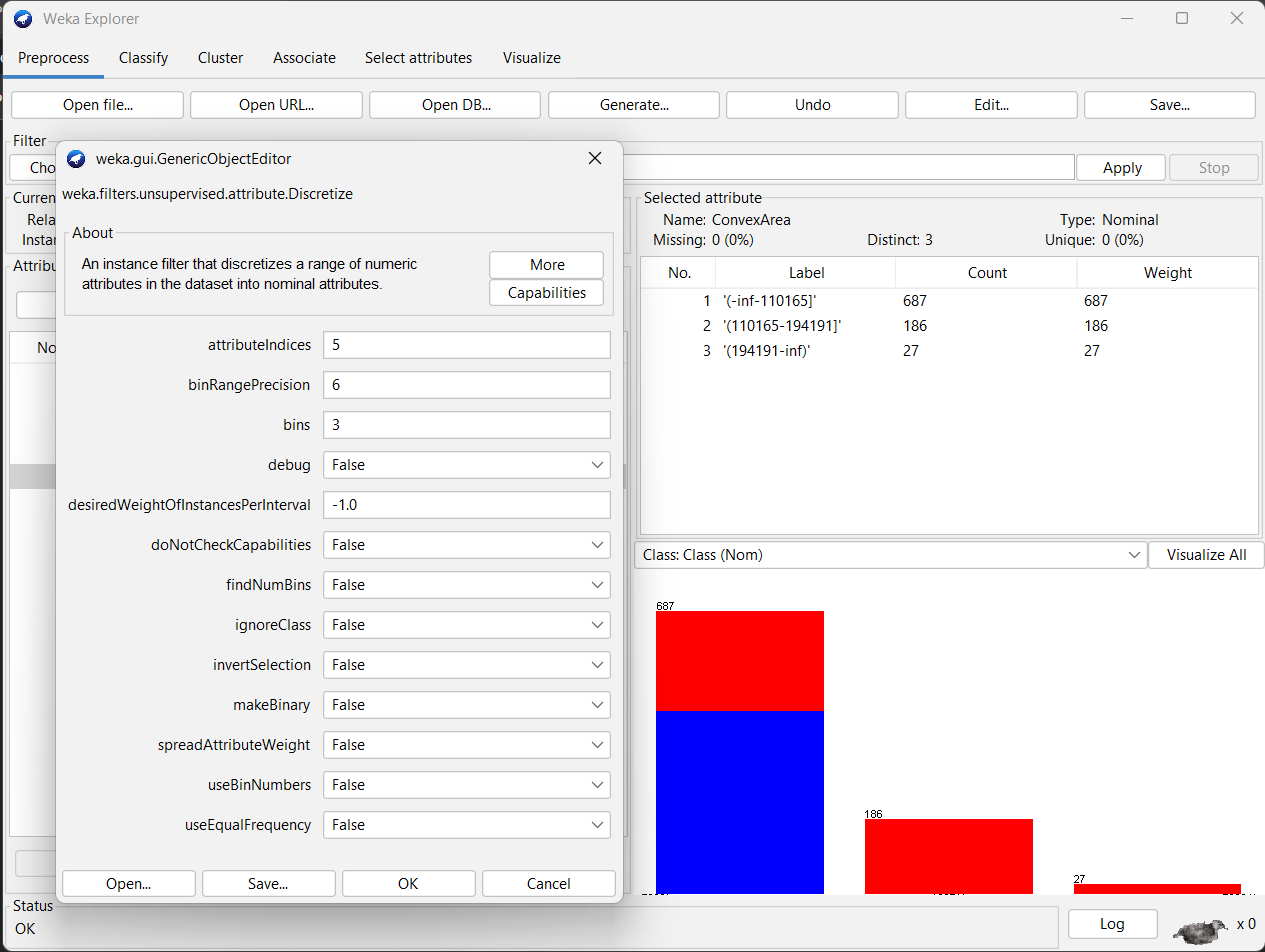
**Figure ‎4‑4 Classifier output**



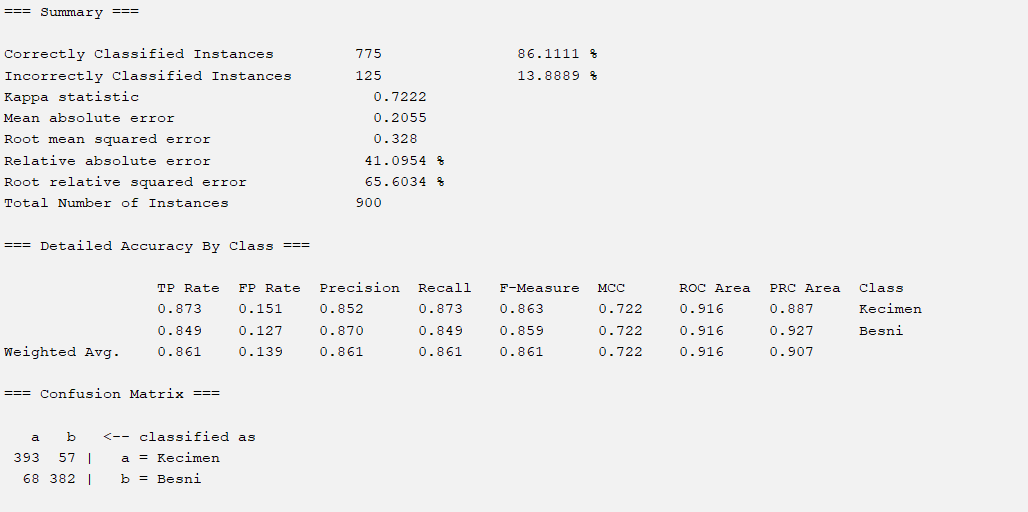
**Figure ‎4‑5 New Naïve Bayes curve**

# Third model (multilayer perceptron)

Before starting our work in the multilayer perceptron model I will make discretization of two continuous attributes that are (Area and ConvexArea) as the following: From Filter we will choose Discretize then I will choose which attribute I will change and the new number of bins. The following figures shows new distribution for Area and Extent after making discretization so the changed from continuous to discrete(Nominal type).



**Figure ‎5‑1 new distribution for Area Figure ‎5‑2 new distribution for ConvexArea**



**Figure 5‑3 Summary of multilayer perceptron**

From the figure 5-3 we see that from the test we got 775 instances are correctly classified with a percentage of accuracy of 86.1111% and with 125 incorrectly classified instances with a percentage of accuracy of 13.8889%. Moreover, we see that the Confusion matrix results are initialized as the Class option results, for example Kecimen (a) is true so 393 instances out of 450 is true with rate of 0.873 and this rate is the true positive rate for Keciman and the False Positive is 68 with a rate of 0.151. The precision, recall and F-measure with their weighted averages were shown in the figure. The same thing for Besni with different numbers and rates.

For The precision we can calculate it as:

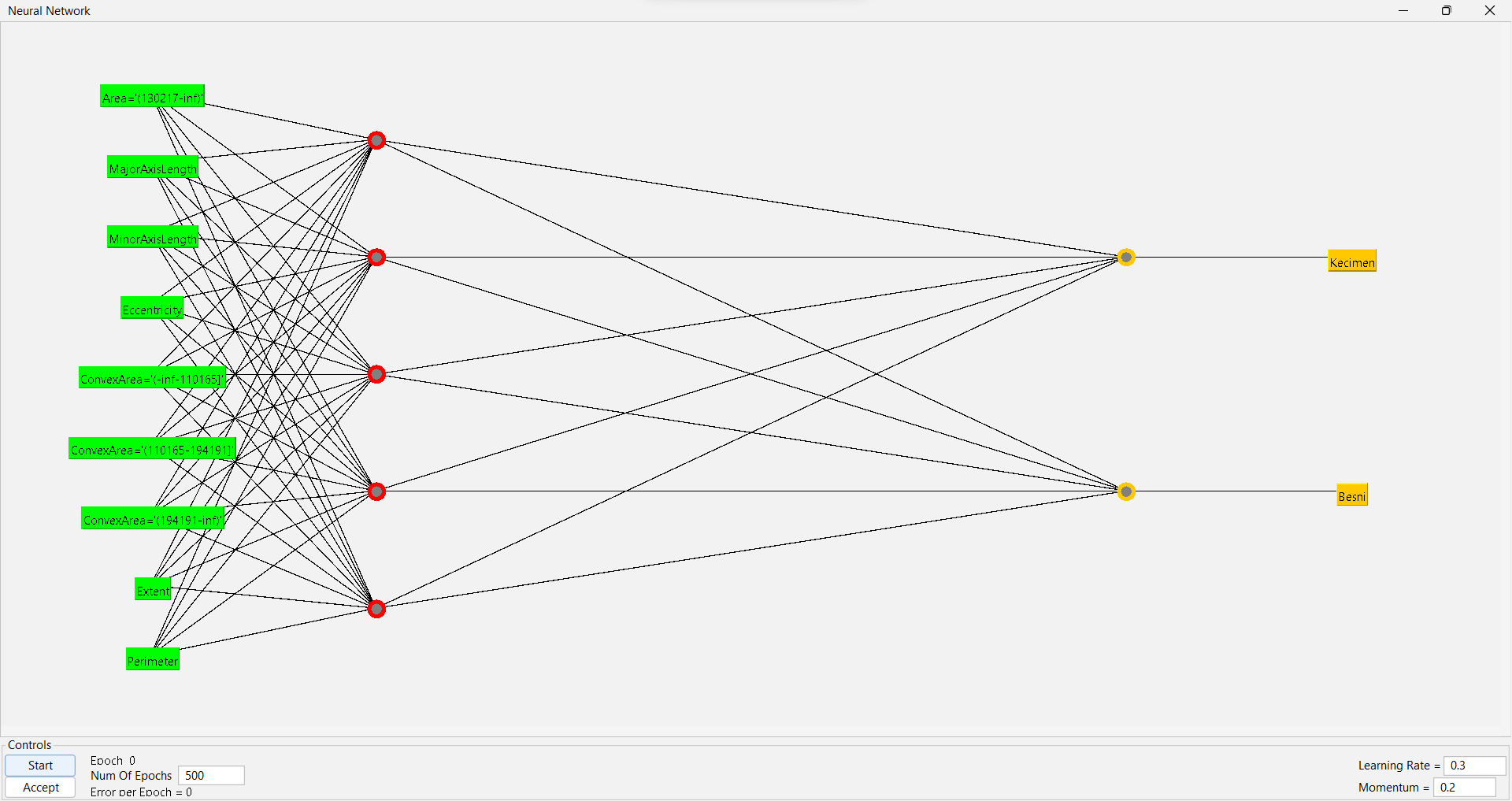
**Precision = TruePositives / (TruePositives + FalsePositives),** for example for Kecimen Precision it is equal: (0.873 / (0.873+0.151)) = 0.852

For The Recall we can calculate it as: **Recall = TruePositives /(TruePositives+FalseNegatives)** for example forKecimen Recall it is equal: (0.873/ (0.873+0.126)) = 0.873

For The F-Measure we can calculate it as: **F-Measure = (2 \* Precision \* Recall) / (Precision + Recall)** for example forKecimen F-Measure it is equal:

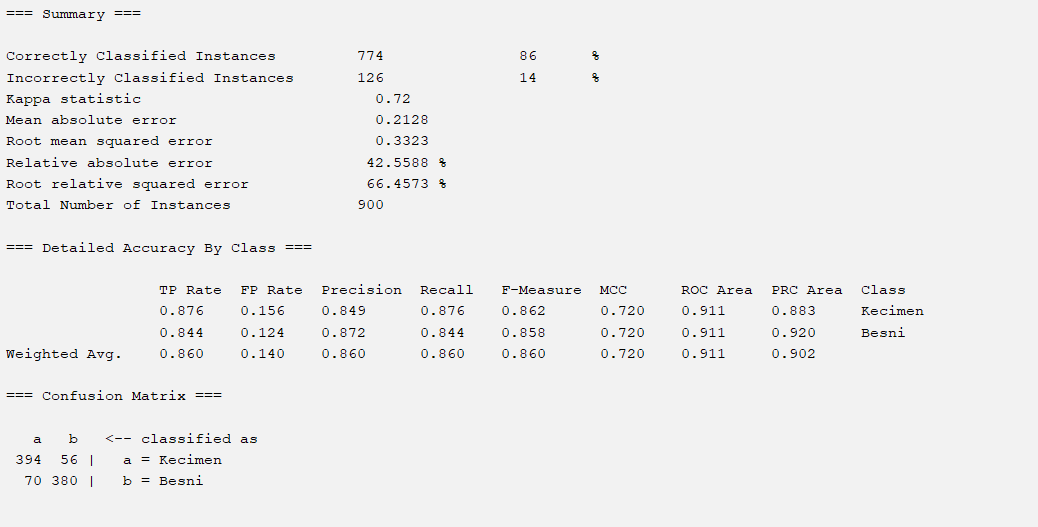
(2\*0.852\*0.87 / (0.852+0.87)) = 0.863

The report shows also all this measures and calculations and confusion matrix for Besni with different numbers and rates.

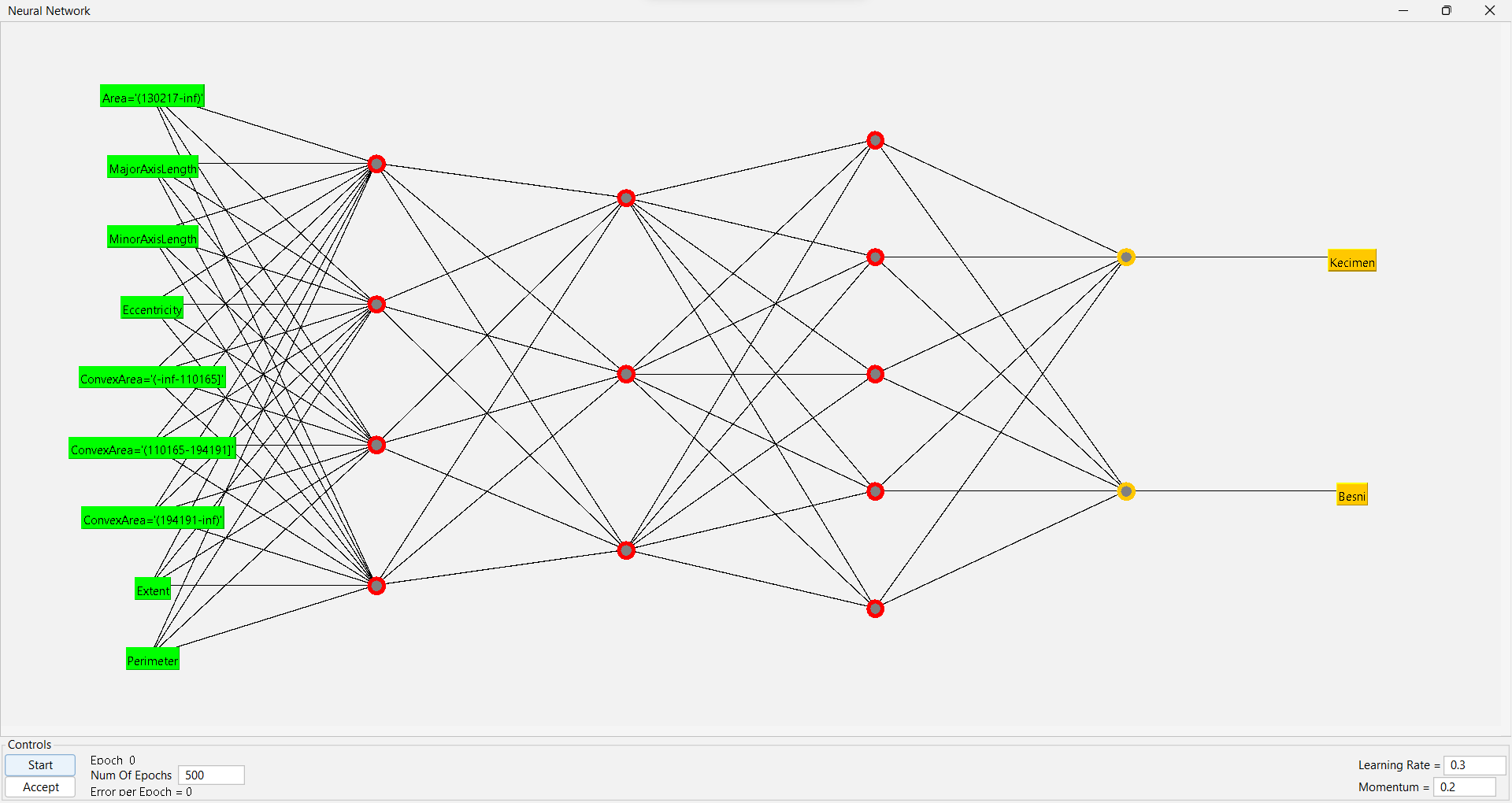


**Figure 5‑4 multilayer perceptron system**

Now we will change the num of hidden layer. So, the precision, recall, F-measure with their weighted averages and percentage of correct/incorrect classification were also affected but the calculation rules will remain the same as shown in the figure below:



**Figure 5‑5 Summary of multilayer perceptron**



**Figure 5‑6 multilayer perceptron system**

# Conclusion

In the end, we can say that we learnt how to use a new tool that is WEKA and used different models to test same data and notice the differences between them how do they work and in the results. As we saw in the results and summary, **Decision Tree** is a supervised algorithm It is using a binary tree and the target values are presented in the tree leaves and we knew the size of tree. On the other hand, Naïve Bayes Tree uses decision tree as the general structure and deploys naïve Bayesian classifiers at leaves it classifiers work better than decision trees when the sample data set is small. So since data is big we can see depend on the percentage of the correct instances for each model that decision tree is better than Naïve Bayes in our case. In Multi-Layer Perceptron the connections between neurons are so-called weights. Their values are selected during the training process. Finally, we can notice that the results for the same dataset has changed depend on the type of model we choose, also in the same model the results changed if we change hyper-parameter and we can notice that when 5 cross validation option test it test all the instances and