Faculty of Engineering and Technology

Electrical and Computer Engineering Department

ENCS3340

Artificial Intelligence

Machine Learning for Classification Project

Student's Name: Lojain Abdalrazaq. **ID Number:** 1190707.
Student's Name: Arwa Doha. **ID Number:** 1190324.

Instructor's Name: Dr. Adnan Yahya.

Section: 1.

June 10, 2022

Abstract

The aim of this project is to learn how to use machine learning tools to test different algorithms for classification tasks for different models such as Decision Tree, Naïve Bayes. In addition, to compare them for a classification task using WEKA tool. The dataset, attributes, experiments, and the results will be discussed and described. And finally, the performance of all the tested methods will be compared.

Early stage diabetes risk prediction Dataset

First of all, according to our student's numbers, the last digit in the least student ID is 4, which means that our dataset will be the number 1 **early stage diabetes risk prediction Dataset.**

The following figure shows the dataset that we have uploaded to the WEKA tool, it is shown that the dataset has 17 attributes, and 520 examples. The attributes are divided into two types, the numeric attributes such as **Age attribute**, and nominal attributes such as **Gender.**

In the case of numeric attributes such as age, the minimum and maximum values are defined. Also, the mean and standard deviation is shown in figure 1.

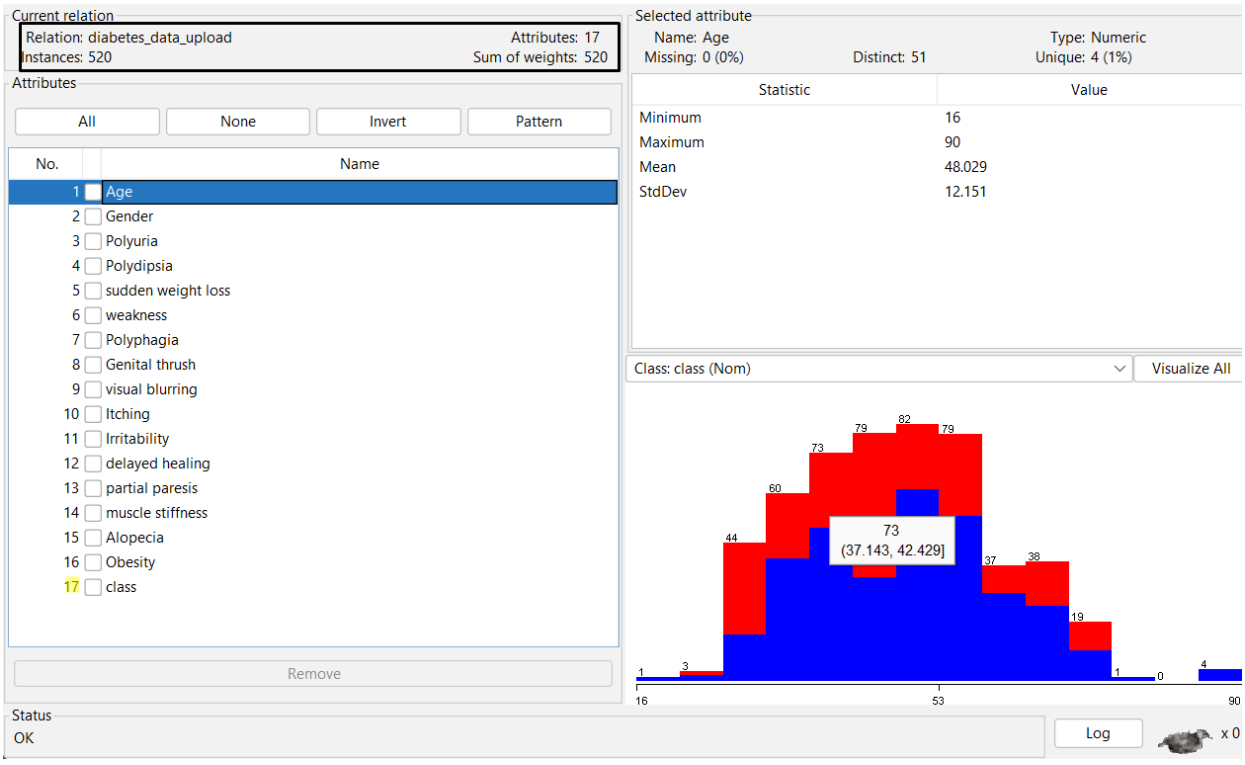


Figure 1 Uploading the diabetes dataset into WEKA tool.

The following figure shows that all the atrrtibutes are **nominal** except the Age attribute is continues attribute (**numerical**).

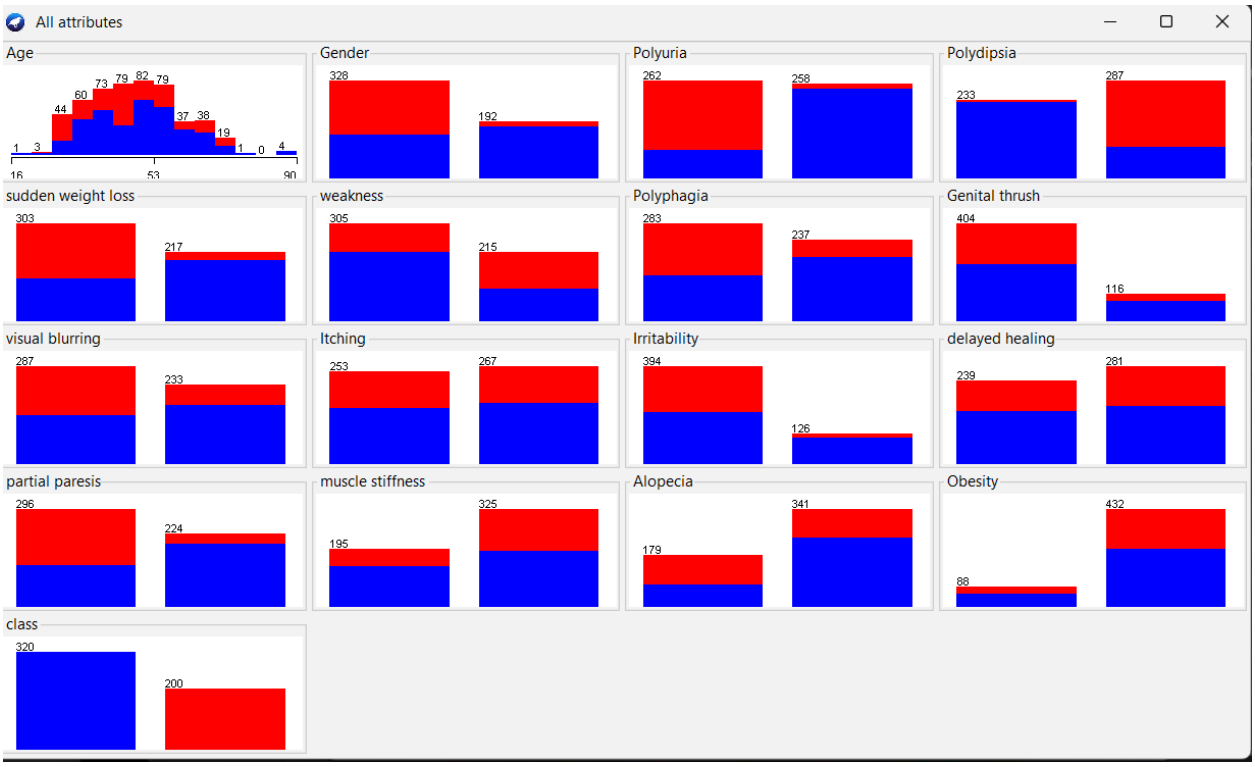
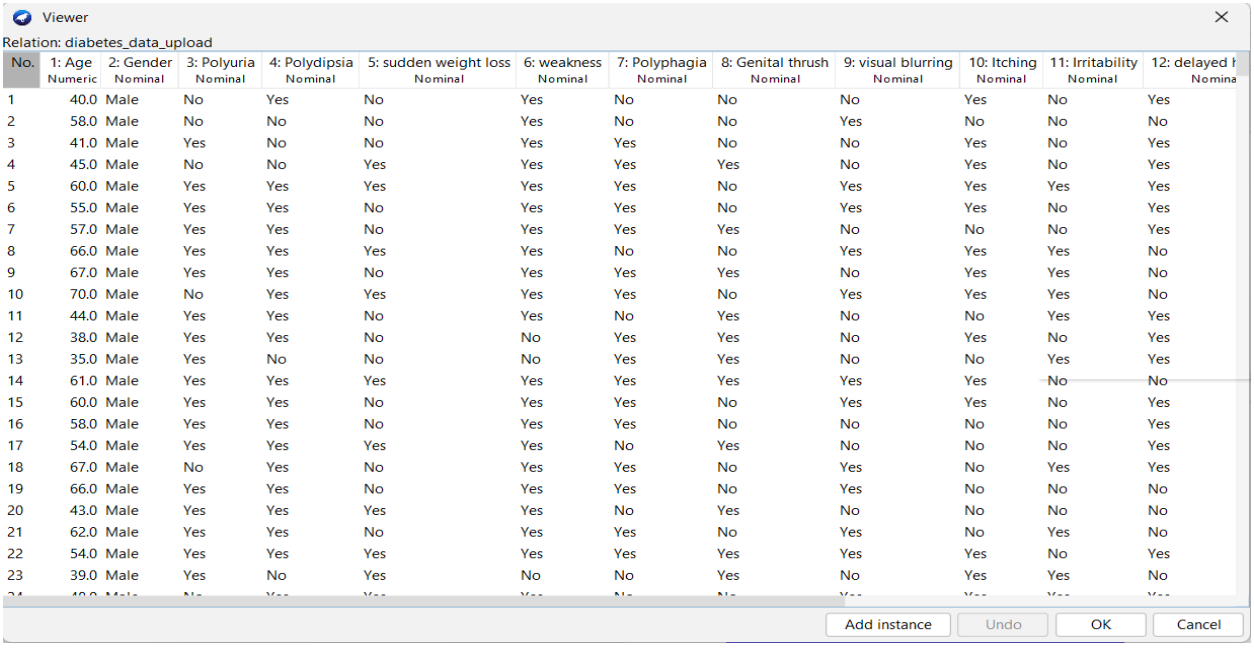


Figure 2 Dataset attributes (nominal and numerical).

And the table of the **early stage diabetes risk prediction Dataset** is represented in WEKA tool is shown in the following figure:


Viewer

Relation: diabetes_data_upload

No.	1: Age Numeric	2: Gender Nominal	3: Polyuria Nominal	4: Polydipsia Nominal	5: sudden weight loss Nominal	6: weakness Nominal	7: Polyphagia Nominal	8: Genital thrush Nominal	9: visual blurring Nominal	10: Itching Nominal	11: Irritability Nominal	12: delayed h Nomina
1	40.0	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes
2	58.0	Male	No	No	No	Yes	No	No	Yes	No	No	No
3	41.0	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes
4	45.0	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes
5	60.0	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
6	55.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes
7	57.0	Male	Yes	Yes	No	Yes	Yes	Yes	No	No	No	Yes
8	66.0	Male	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No
9	67.0	Male	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	No
10	70.0	Male	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No
11	44.0	Male	Yes	Yes	No	Yes	No	Yes	No	No	Yes	Yes
12	38.0	Male	Yes	Yes	No	No	Yes	Yes	No	Yes	No	Yes
13	35.0	Male	Yes	No	No	No	Yes	Yes	No	No	Yes	Yes
14	61.0	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
15	60.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes
16	58.0	Male	Yes	Yes	No	Yes	Yes	No	No	No	No	Yes
17	54.0	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	Yes
18	67.0	Male	No	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes
19	66.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	No	No
20	43.0	Male	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No
21	62.0	Male	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	No
22	54.0	Male	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
23	39.0	Male	Yes	No	Yes	No	No	Yes	No	Yes	Yes	No

Add instance | Undo | OK | Cancel

Figure 3 The diabetes risk prediction Dataset table in WEKA tool.

1. Decision Tree

The first step is to preprocess at least one continous attribute, the following figure represents preprocess the **Age attribute** using **discretization** into two bins, the first value from (-∞-53) while the second one from (53-∞) as shown:

Figure 5 Preprocess the Age attribute using discretization.

Now, after preprocecing the continua attrubutes, the Decision Tree model was tested using 5-fold cross validation, which is standard evaluation technique, and allow us to devide the applied dataset into 5 parts or pieces(**folds**). The classification output is as the following:

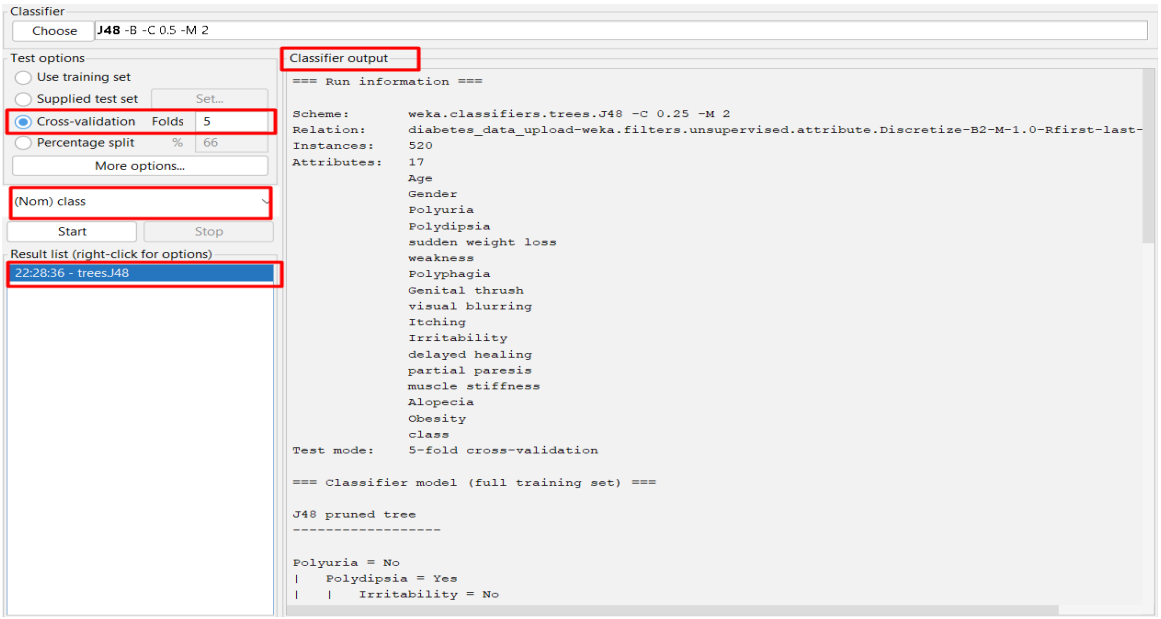


Figure 6 Decision Tree classifier output.

Now, from the running information, it is shown that the number of leaves is 20, while the Size of the tree is 39.

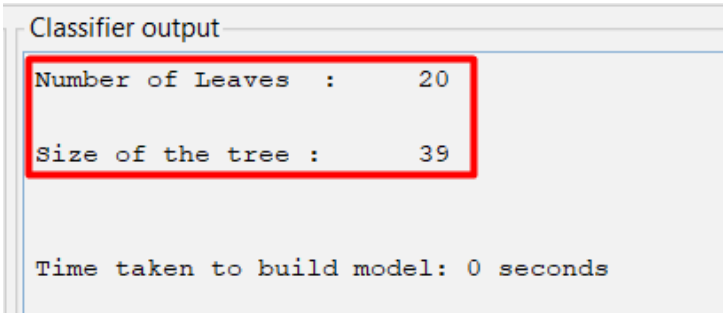


Figure 7 The number of leaves and the size of the tree.

The following figure shows that the number of instances that are correctly classified is 489 instances with 94.0385% percentage. While the number of incorrectly classified instances is 31with 5.9615% percentage.

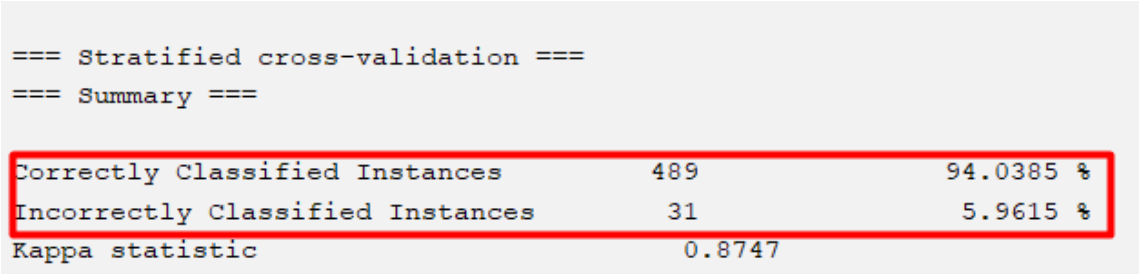


Figure 8 Correctly and incorrectly Classified Instances.

From the following results, the confusion matrix shows the rows (**a=Positive** and **b=Negative**) which are the prediction, while the columns represents the true classes.

- ✓ **302:** represents the true positive.
- ✓ **18:** represents the false positive.
- ✓ **13:** represents the false negative.
- ✓ **187:** represents the true negative.

In addition, precision and recall are as the following:

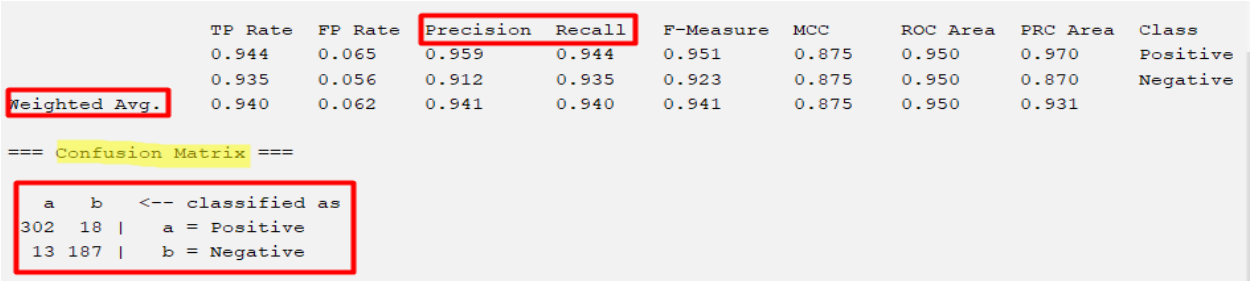


Figure 9 The precision, recall and the confusion matrix of the tree.

While the visualized tree is as the following:

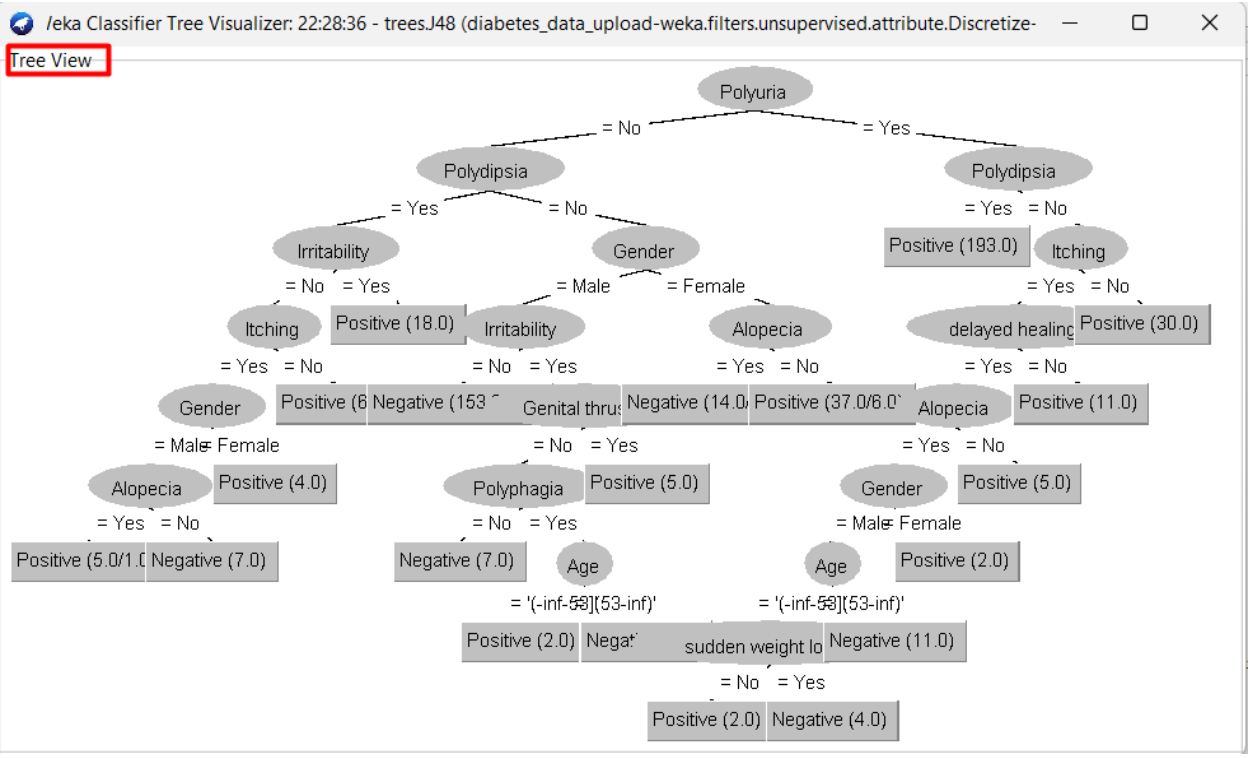


Figure 10 Visualized tree.

To change one hyper parameter, we have changed the binarySplits from false to true, but no changes happened in the classifier output.

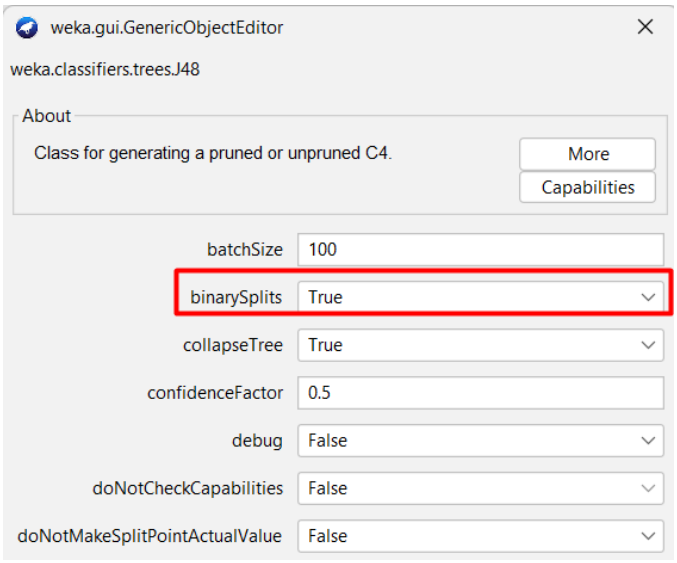


Figure 11 changing the hyper-parameter binarysplits to true.

While when changing the hyper-parameter confidence factor to 0.5, the number of leaves changed to 23, and the size of the tree changed to 45.

```
Number of Leaves  :     23

Size of the tree :     45
```

Figure 12 The number of leaves and the size of the tree when changing the hyper-parameter.

But, the number of instances that are correctly classified, and the number of incorrectly classified instances still the same as shown in the following figure:

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

Correctly Classified Instances         489               94.0385 %
Incorrectly Classified Instances        31                5.9615 %
```

Figure 13 Correctly and incorrectly Classified Instances when changing the hyper-parameter.

Finally the confusion matrix, precision and recall after changing the hyper parameter are shown:

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

Correctly Classified Instances         489               94.0385 %
Incorrectly Classified Instances        31                5.9615 %
Kappa statistic                          0.8756
Mean absolute error                      0.0734
Root mean squared error                  0.232
Relative absolute error                 15.4976 %
Root relative squared error             47.6883 %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.931     0.045     0.971       0.931   0.951      0.877    0.955     0.971     Positive
                 0.955     0.069     0.897       0.955   0.925      0.877    0.955     0.878     Negative
Weighted Avg.    0.940     0.054     0.942       0.940   0.941      0.877    0.955     0.935

=== Confusion Matrix ===

   a   b   <-- classified as
 298  22 |   a = Positive
   9 191 |   b = Negative
```

Figure 14 The confusion matrix, precision and recall.

And finally the visualized tree:

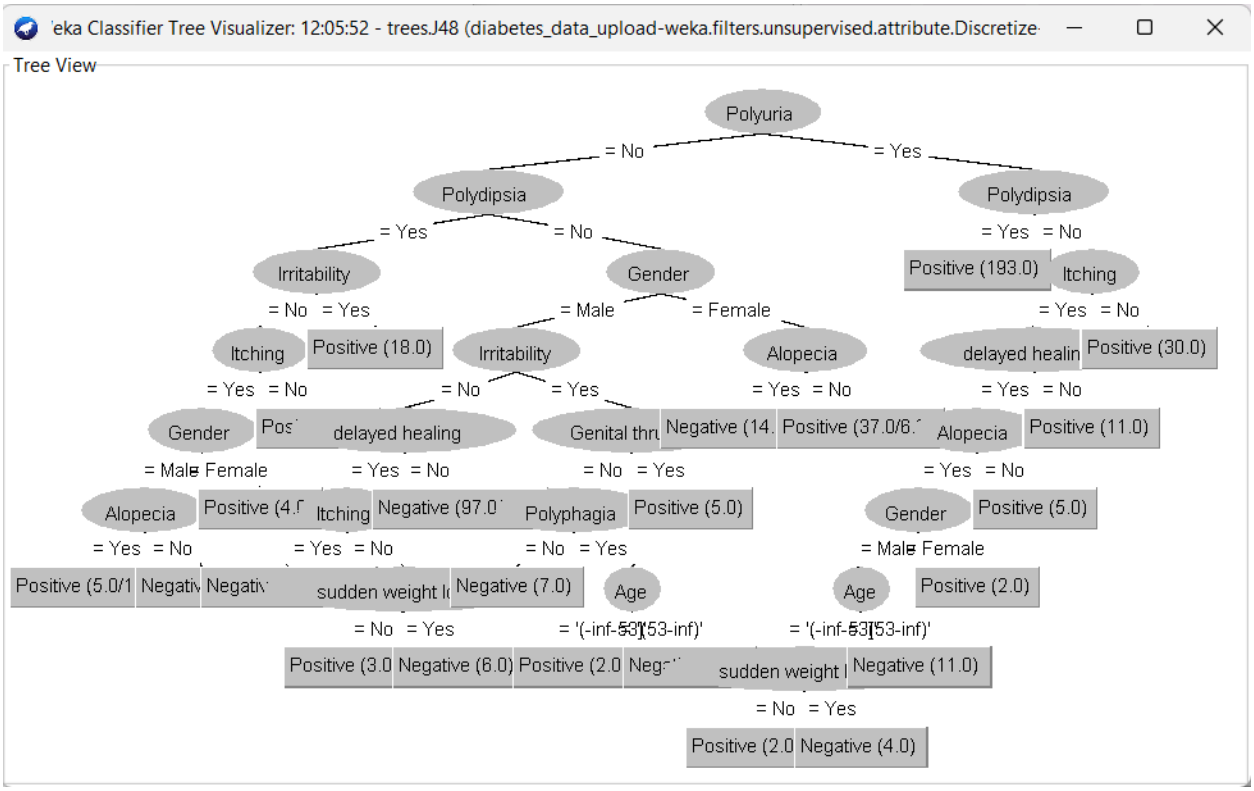


Figure 15 Visualized tree after applying the change of hyper parameter.

2. Naïve Bayes

After preprocecing the continous attrubutes, the Naïve Bayes model was tested using 5-fold cross validation. The classification output is as the following:

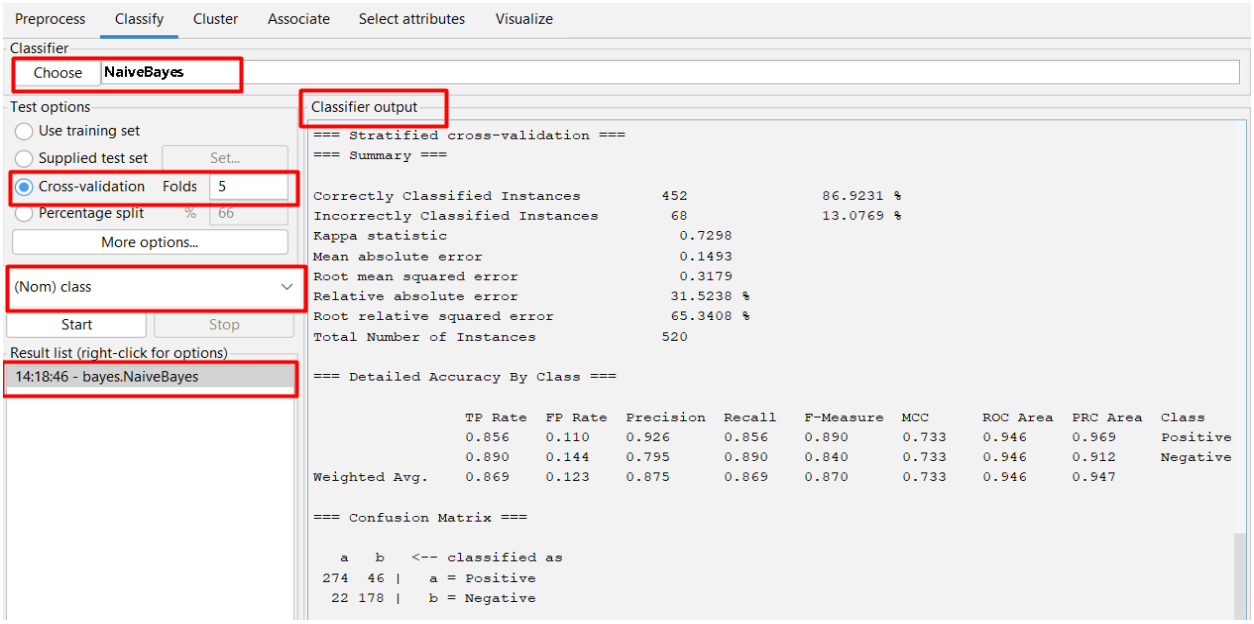


Figure 16 Naïve Bayes Classifier output.

From the results of this model, the number of correctly classified instances is 452 with 86.9231% percentage, while the incorrectly classified instances is 68 with 13.0769% percentage as shown:

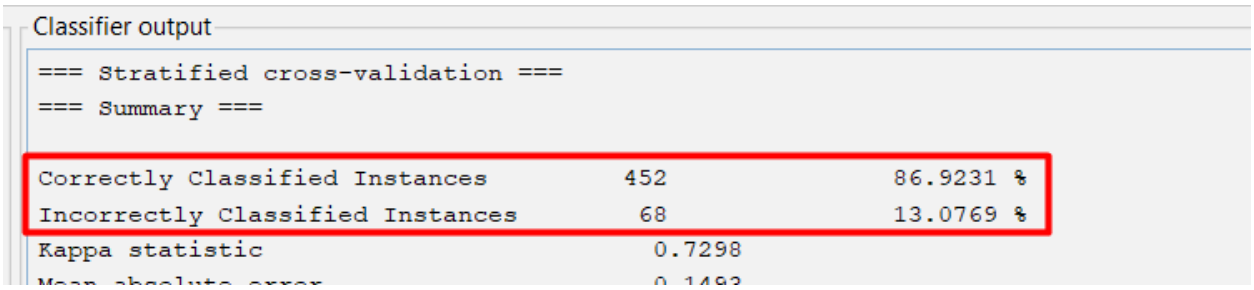


Figure 17 Correctly and incorrectly Classified Instances.

And the confusion matrix shows the rows (**a=Positive** and **b=Negative**) which are the prediction, while the columns represents the true classes as mentioned before.

- ✓ **274:** represents the true positive.
- ✓ **46:** represents the false positive.
- ✓ **22:** represents the false negative.
- ✓ **178:** represents the true negative.

In addition, precision and recall are as the following:

```
=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.856     0.110     0.926       0.856   0.890      0.733    0.946     0.969     Positive
                 0.890     0.144     0.795       0.890   0.840      0.733    0.946     0.912     Negative
Weighted Avg.    0.869     0.123     0.875       0.869   0.870      0.733    0.946     0.947

=== Confusion Matrix ===

   a   b   <-- classified as
 274  46 |   a = Positive
  22 178 |   b = Negative
```

Figure 18 The precision, recall and the confusion matrix of the Naive model.

After changing the hyper-parameter **kernel estimator** to true value and changing the **batch-size** to the double (100 to 200), no effect happened, and no change on the results as shown in the following figure:

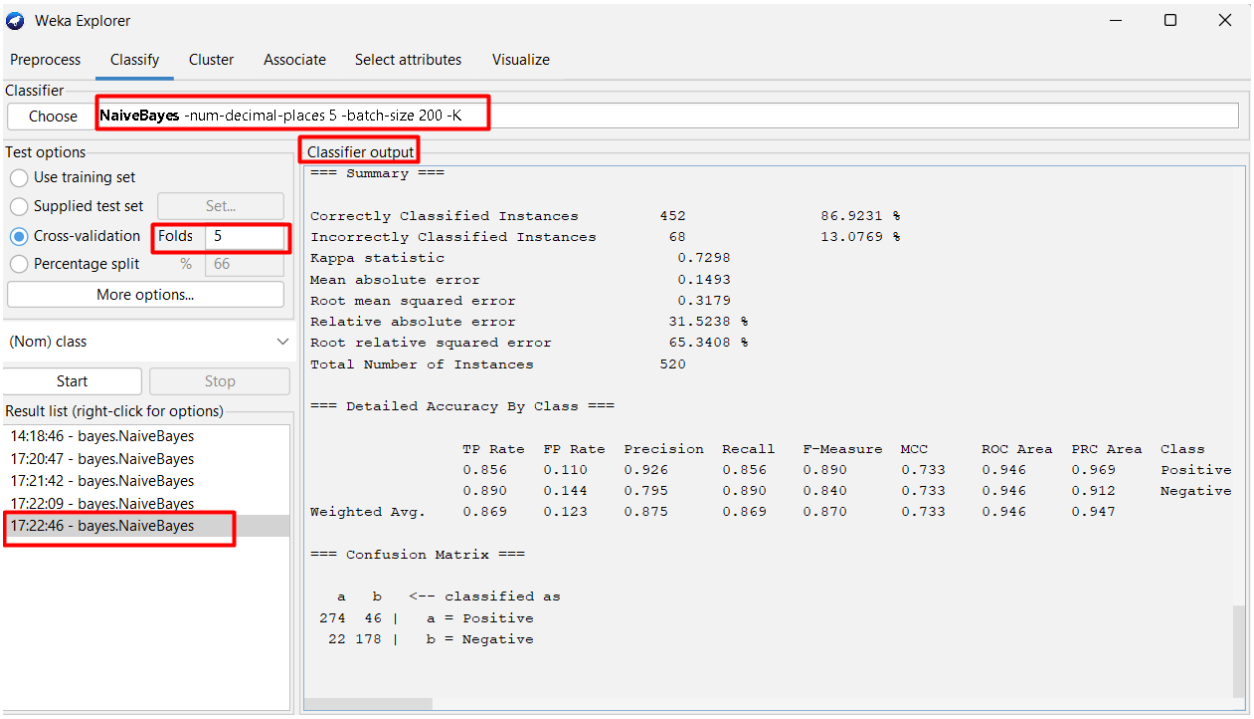


Figure 19 Changing the hyper-parameter of Nayive Bayas.

3. Multilayer Perceptron

The same as previous steps, and after applying preprocecing the continous attributes, the Multilayer Perceptron model was tested using 5-fold cross validation. The classification output is as the following:

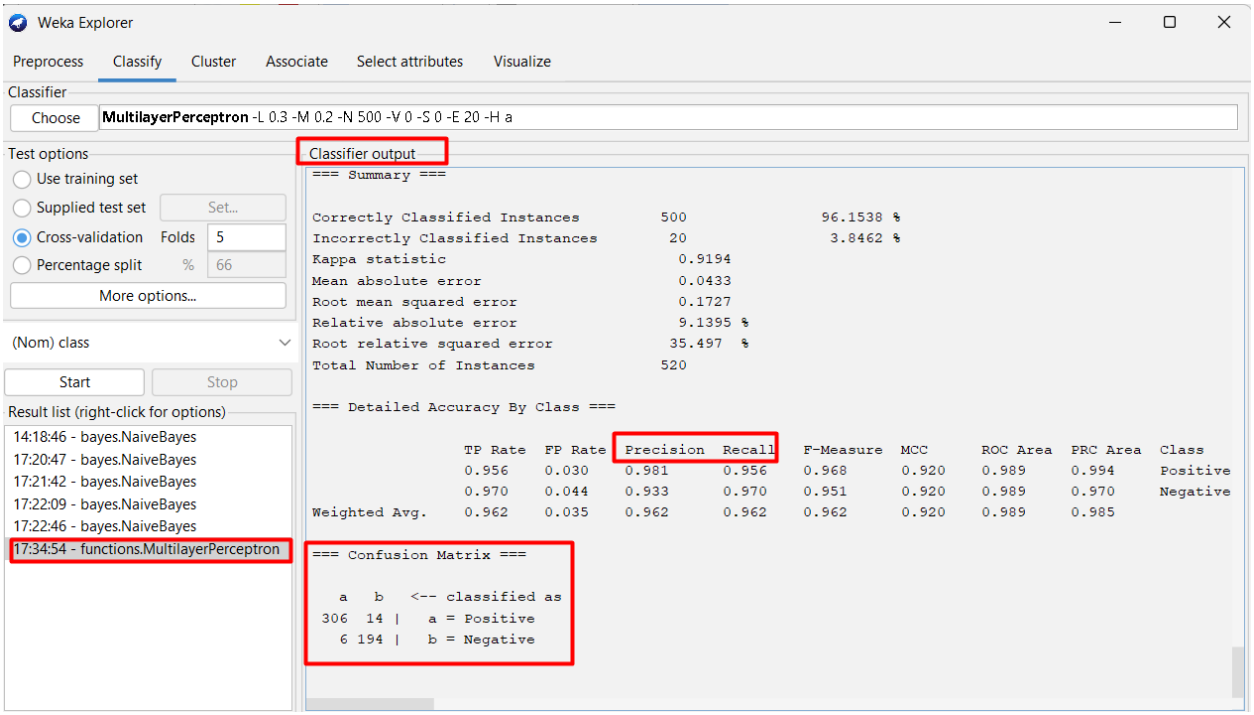


Figure 20 The output classification of Multilayer Perceptron model.

From the previous graph, it is shown that the correctly classified instances are 500 with 96.1538%. While the incorrectly classified instances are 20 with 3.8462%.

The confusion matrix, the precision and the recall of the **Multilayer Perceptron** are shown in the following figure:

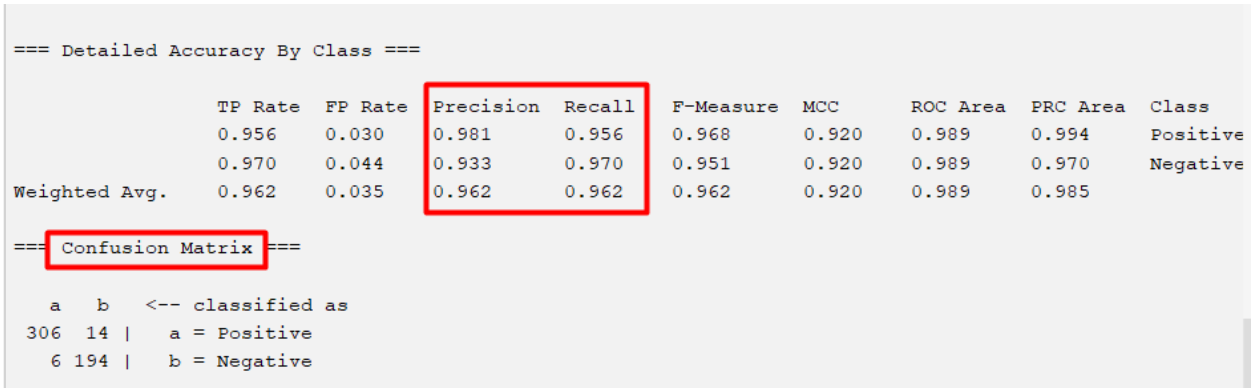


Figure 21 confusion matrix, the precision and the recall of the Multilayer Perceptron.

The following figure represent the Neural Network of out dataset:

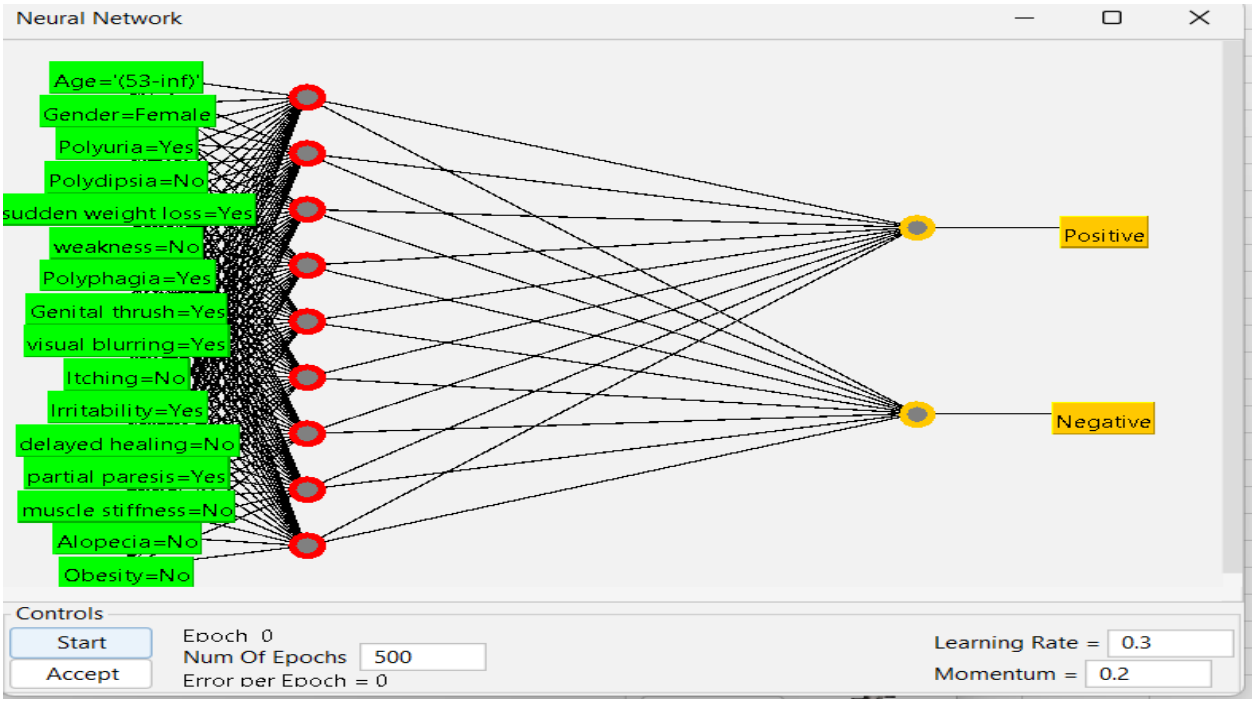


Figure 22 Neural Network

In addition, the confusion matrix shows the rows (**a=Positive** and **b=Negative**) which are the prediction, while the columns represents the true classes as mentioned before.

- ✓ **306:** represents the true positive.
- ✓ **14:** represents the false positive.
- ✓ **6:** represents the false negative.
- ✓ **194:** represents the true negative.

```
=== Confusion Matrix ===

   a   b   <-- classified as
 306  14 |   a = Positive
   6 194 |   b = Negative
```

Figure 23 The confusion matrix of Multilayer Perceptron model.

After changing the **decay** hyper-parameter from false to true, the results became as the following:

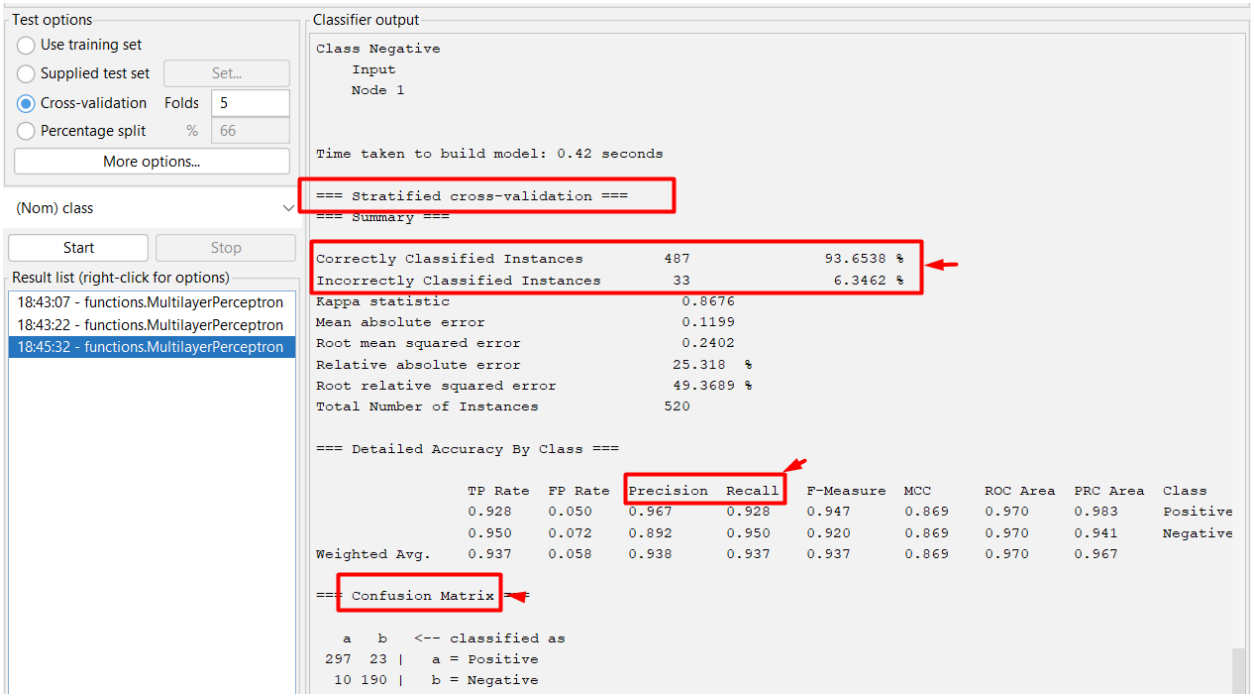


Figure 24 The confusion matrix, precision and recall after changing the parameter.

From the previous figure, and after applying the hyper parameter change, the correctly classified instances became 407 with 93.65% instead of 500 with 96.1538%, while the incorrectly classified instances are 33 with 6.34% instead of 20 with 3.8462%.

Conclustion

Overall, this project has covered some machine learning algorithms, and learning how to compare a classification task using WEKA. And we test 3 models: Decision Tree, Naïve Bayes, and Multilayer Perceptron.

And we noticed that the percentage of correctly classified is obviously change from one method to another, and they are for this three methods, in the following order: 94.0385%, 86.9231%, and 96.1538%. And this result shows that Multilayer Perceptron is a better than both Decision tree and Naïve Bayes model.