جامعة بيرزيت

BIRZEIT UNIVERSITY

Faculty of Engineering & Technology – Computer Systems Engineering Department

Second Semester 2023-2024

Artificial Intelligent ENCS3340

Project #2:
Machine Learning for Classification

Prepared by:

Rasha Daoud - 1210382

Nadia Thaer - 1210021

Instructor: Aziz Qaroush

Section: 3

Date: 18st June 2024

Table of Contents

List of Figures:

Introduction

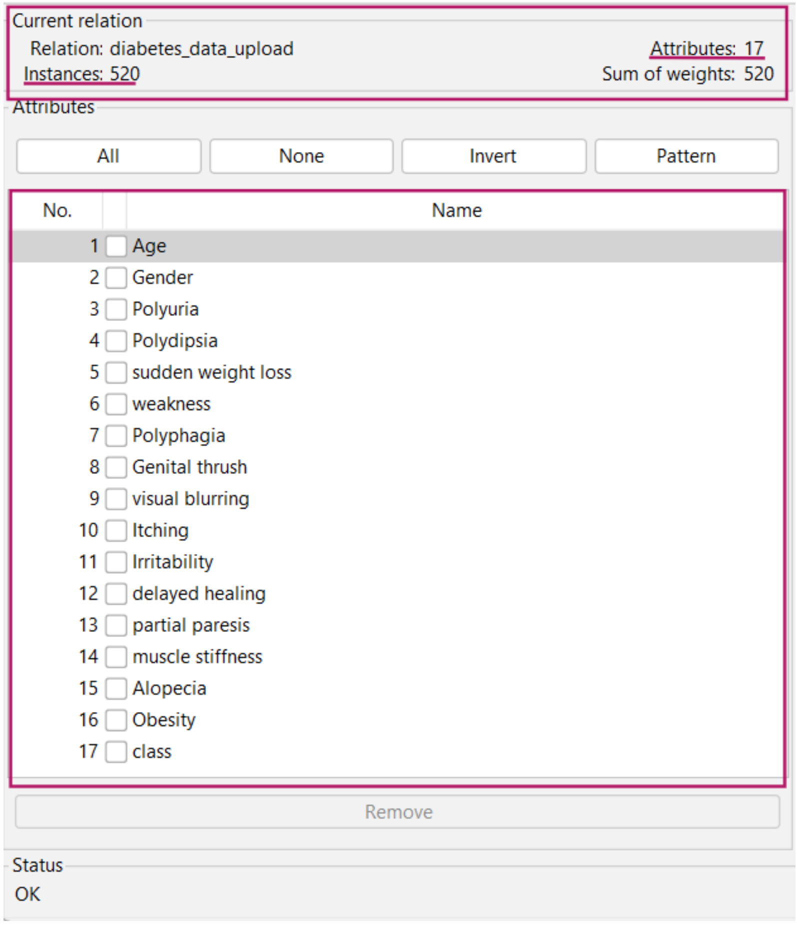
The Dataset used:

Id #1: 1210382

Id #2: 1210021 ➡ 1%3=1

As requested, based on the last digit of the lowest student ID number in the team, we select dataset #1 which is about "Early stage diabetes risk prediction Dataset".

The Attributes:



Current relation

Relation: diabetes_data_upload

Attributes: 17

Instances: 520

Sum of weights: 520

Attributes

No.		Name
1	□	Age
2	□	Gender
3	□	Polyuria
4	□	Polydipsia
5	□	sudden weight loss
6	□	weakness
7	□	Polyphagia
8	□	Genital thrush
9	□	visual blurring
10	□	Itching
11	□	Irritability
12	□	delayed healing
13	□	partial paresis
14	□	muscle stiffness
15	□	Alopecia
16	□	Obesity
17	□	class

Status

OK

Figure 1: Dataset on WEKA.

Number of attributes= 17, Number of instance= 520

The set of attributes are listed in the second box.

Dataset Description

The "Early Stage Diabetes Risk Prediction Dataset" consists of health-related attributes aimed at predicting the risk of diabetes. The dataset includes the following attributes:

1. **Age**: Age of the patient (years).
2. **Gender**: Male or Female.
3. **Polyuria**: Frequent urination (Yes/No).
4. **Polydipsia**: Excessive thirst (Yes/No).
5. **Sudden weight loss**: Sudden weight loss (Yes/No).
6. **Weakness**: General body weakness (Yes/No).
7. **Polyphagia**: Excessive hunger (Yes/No).
8. **Genital thrush**: Yeast infection (Yes/No).
9. **Visual blurring**: Blurred vision (Yes/No).
10. **Itching**: Itching (Yes/No).
11. **Irritability**: Irritability (Yes/No).
12. **Delayed healing**: Delay in wound healing (Yes/No).
13. **Partial paresis**: Muscle weakness (Yes/No).
14. **Muscle stiffness**: Muscle stiffness (Yes/No).
15. **Alopecia**: Hair loss (Yes/No).
16. **Obesity**: Obesity (Yes/No).
17. **Class**: Diabetes risk (Positive/Negative).

The goal is to predict the "Class" attribute based on the other features.

Discretization

In all three branches, the process of discretization in the first part is the same:

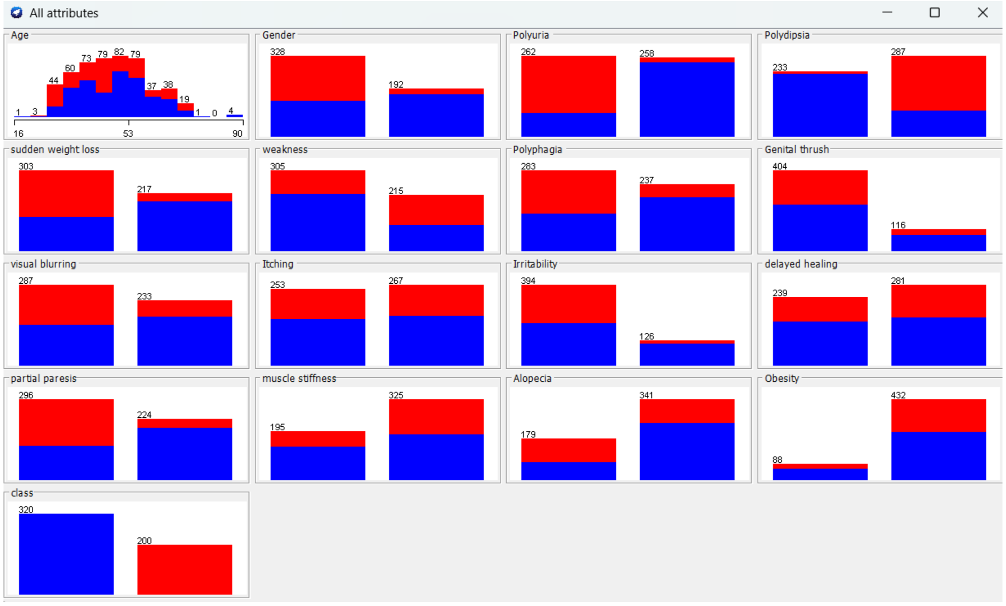
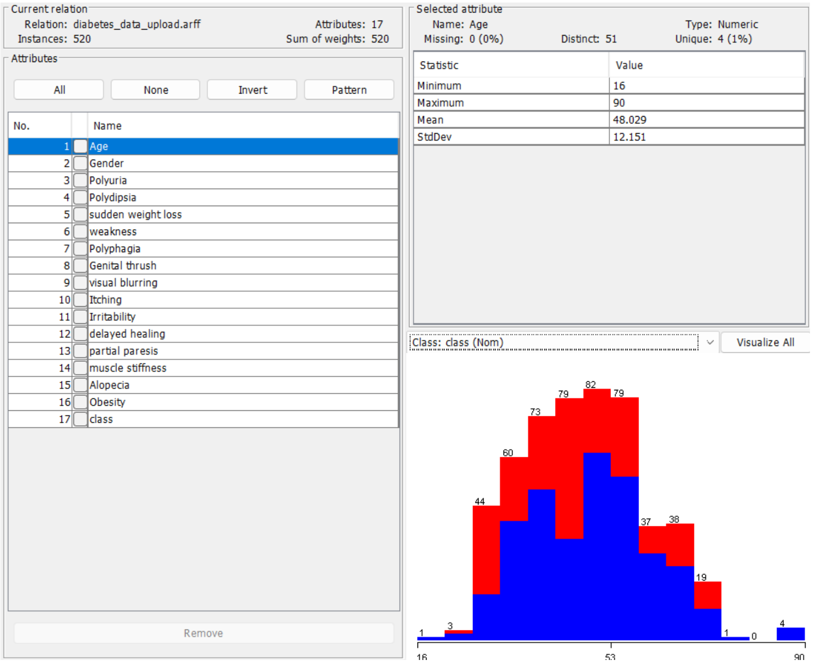


Figure 2: Distribution for all attributes before Discretization.

For the first attribute "Age" it is the only continuous attribute, so we can discretize it:


Current relation

Relation: diabetes_data_upload.arff   Attributes: 17
Instances: 520   Sum of weights: 520

Attributes

All   None   Invert   Pattern

No.	Name
1	Age
2	Gender
3	Polyuria
4	Polydipsia
5	sudden weight loss
6	weakness
7	Polyphagia
8	Genital thrush
9	visual blurring
10	Itching
11	Irritability
12	delayed healing
13	partial paresis
14	muscle stiffness
15	Alopecia
16	Obesity
17	class

Remove

Selected attribute

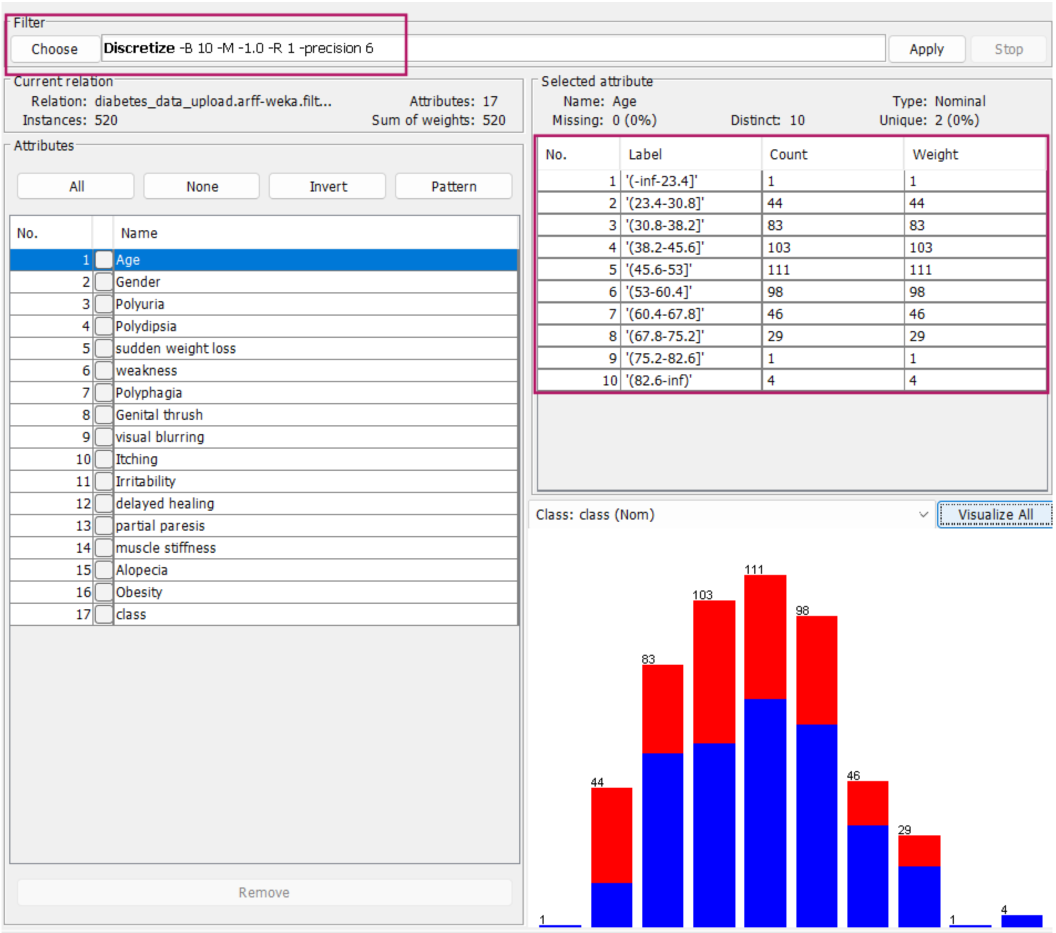
Name: Age   Type: Numeric
Missing: 0 (0%)   Distinct: 51   Unique: 4 (1%)

Statistic	Value
Minimum	16
Maximum	90
Mean	48.029
StdDev	12.151

Class: class (Nom)   Visualize All

Figure 3: Age attribute before discretization.

Applying Discretize filter on "Age" attribute using 10 bins, give us the following result:



Discretize -B 10 -M -1.0 -R 1 -precision 6

Current relation

Relation: diabetes_data_upload.arff-weka.filt...
Attributes: 17
Instances: 520
Sum of weights: 520

Attributes

No.		Name
1	□	Age
2	□	Gender
3	□	Polyuria
4	□	Polydipsia
5	□	sudden weight loss
6	□	weakness
7	□	Polyphagia
8	□	Genital thrush
9	□	visual blurring
10	□	Itching
11	□	Irritability
12	□	delayed healing
13	□	partial paresis
14	□	muscle stiffness
15	□	Alopecia
16	□	Obesity
17	□	class

Selected attribute

Name: Age
Type: Nominal
Missing: 0 (0%)
Distinct: 10
Unique: 2 (0%)

No.	Label	Count	Weight
1	'(-inf-23.4]'	1	1
2	'(23.4-30.8]'	44	44
3	'(30.8-38.2]'	83	83
4	'(38.2-45.6]'	103	103
5	'(45.6-53]'	111	111
6	'(53-60.4]'	98	98
7	'(60.4-67.8]'	46	46
8	'(67.8-75.2]'	29	29
9	'(75.2-82.6]'	1	1
10	'(82.6-inf)'	4	4

Class: class (Nom)

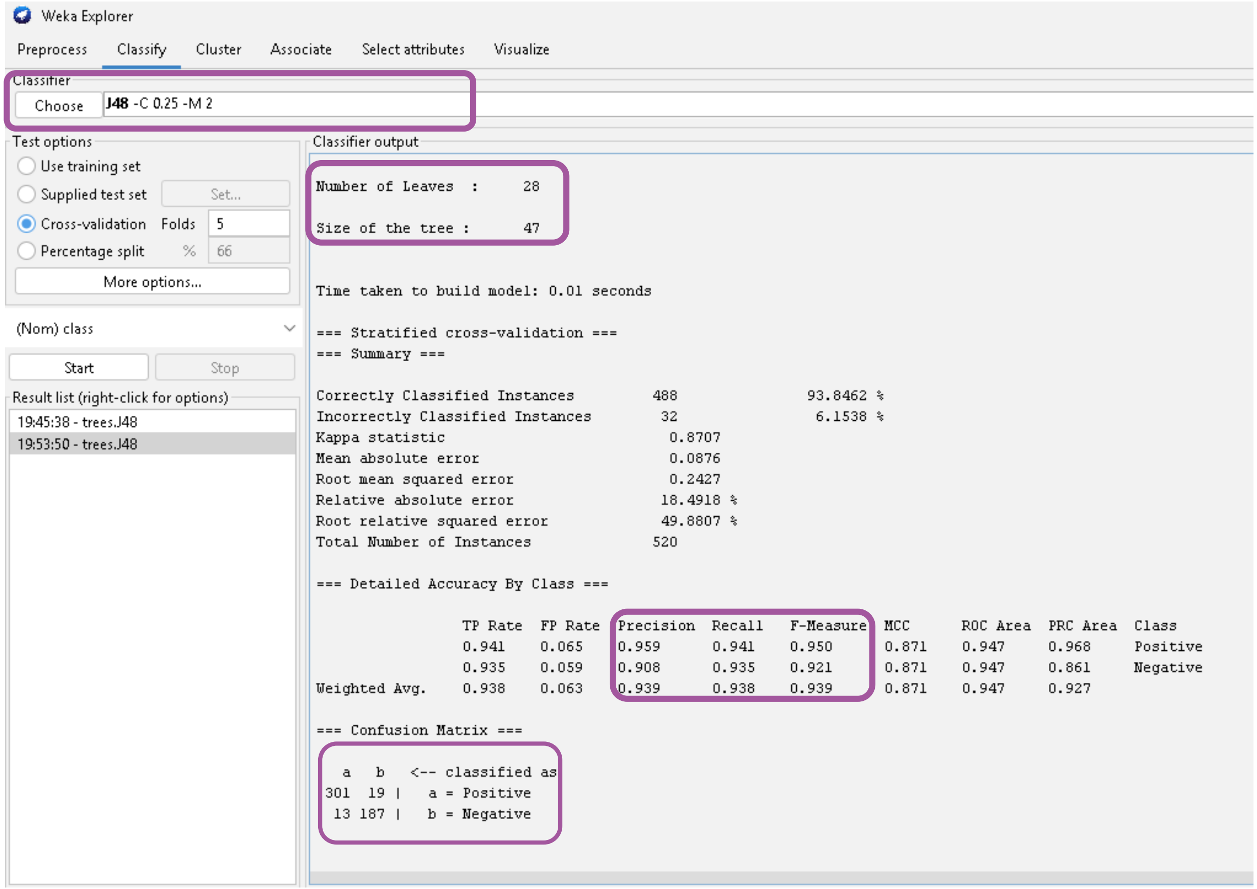
Figure 4: Age attribute after Discretization with 10 bins.

The discretize filter applied to the attribute shown above in the first box. And in the second one, the new discrete values and their count are shown in detail.

Decision Tree

5-folds 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 below.



Classifier: J48 -C 0.25 -M 2

Test options:
- ○ Use training set
- ○ Supplied test set
- ● Cross-validation Folds 5
- ○ Percentage split % 66

(Nom) class

Result list:
- 19:45:38 - trees.J48
- 19:53:50 - trees.J48

```
Number of Leaves  :     28

Size of the tree :      47


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances         488               93.8462 %
Incorrectly Classified Instances        32                6.1538 %
Kappa statistic                          0.8707
Mean absolute error                      0.0876
Root mean squared error                  0.2427
Relative absolute error                 18.4918 %
Root relative squared error             49.8807 %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.941    0.065    0.959      0.941    0.950      0.871    0.947     0.968     Positive
                 0.935    0.059    0.908      0.935    0.921      0.871    0.947     0.861     Negative
Weighted Avg.    0.938    0.063    0.939      0.938    0.939      0.871    0.947     0.927

=== Confusion Matrix ===

   a   b   <-- classified as
 301  19 |   a = Positive
  13 187 |   b = Negative
```

Figure 5: Test Decision Tree using 5-fold cross validation.

From the Result of Classify

- The Number of Leaves: 28, Size of the tree: 47
- The Classifier model use is → j48 (Type of Decision tree)
- From the figure, it is shown that the correctly classified instances are 93.8462 %
- The Result Found from the Confusion Matrix:

TP =301, FN =19, FP = 13 and TN = 187.

As the precision is 0.959, recall 0.941, F-Measure = 0.950. These results could be acceptable.

Tree View:

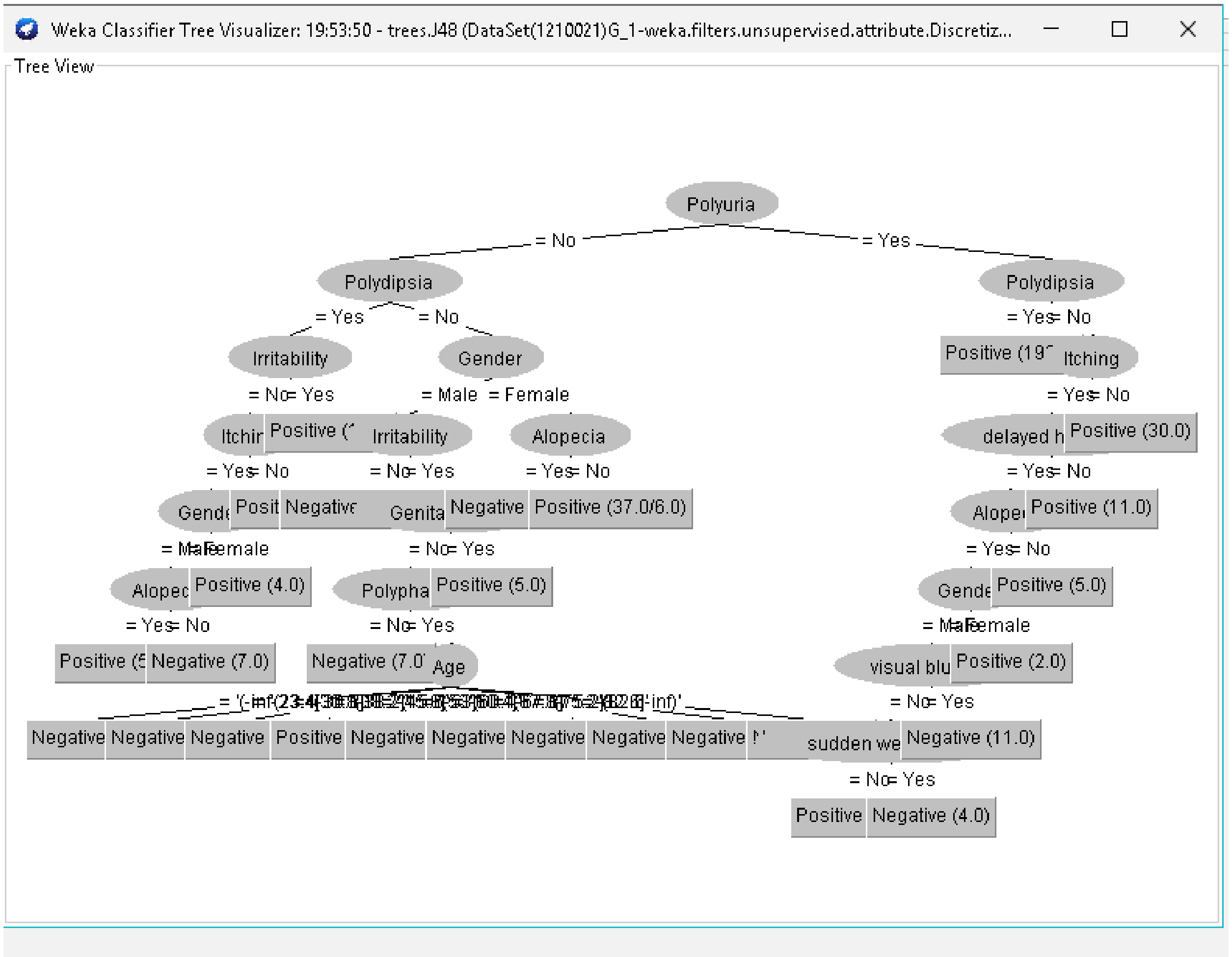
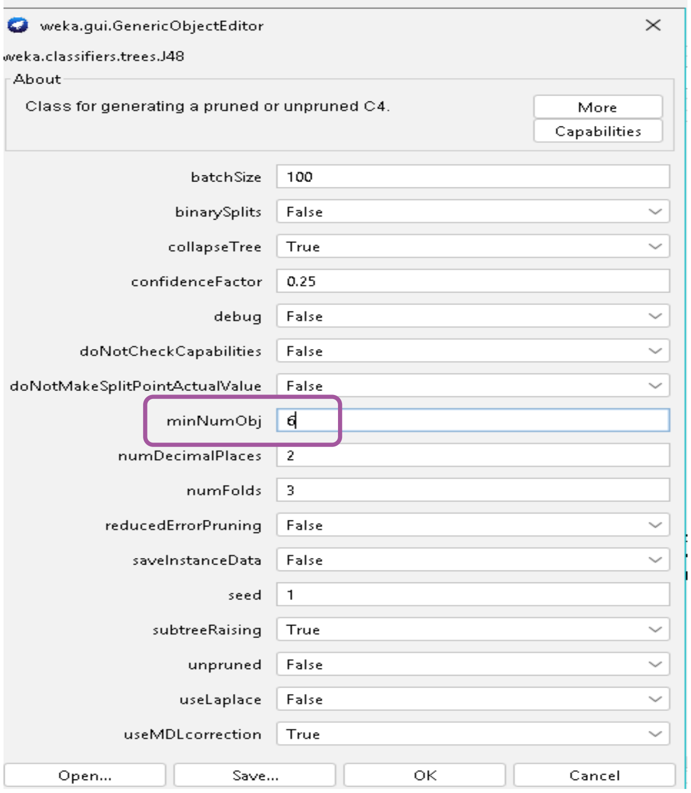


Figure 6: Tree, bins=10

Changing hyper-parameters.

Changing was done on 10 bins data output, and the MinNumObj was 2 Then set 6



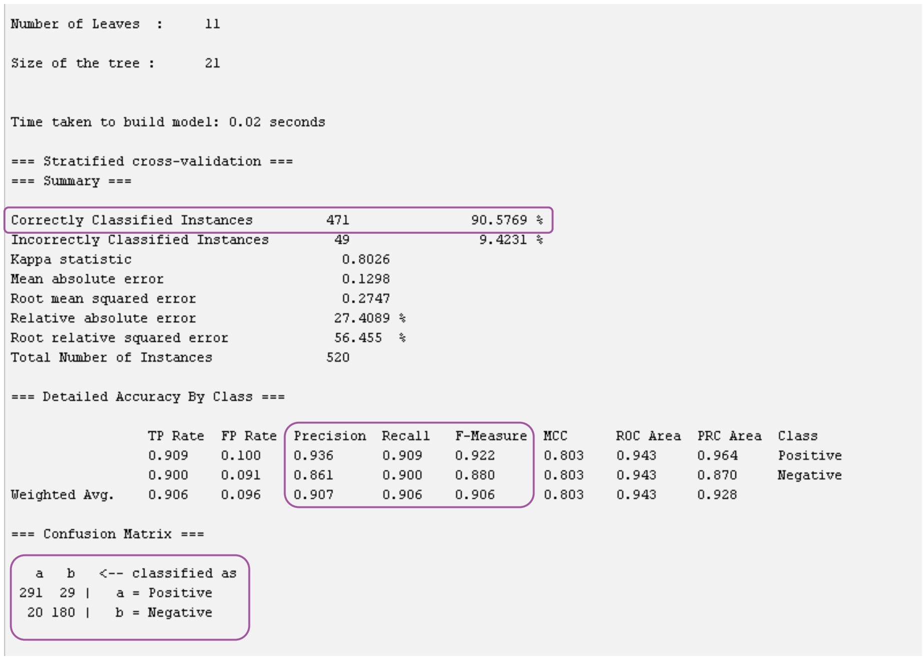
weka.classifiers.trees.J48

About

Class for generating a pruned or unpruned C4.

batchSize	100
binarySplits	False
collapseTree	True
confidenceFactor	0.25
debug	False
doNotCheckCapabilities	False
doNotMakeSplitPointActualValue	False
minNumObj	6
numDecimalPlaces	2
numFolds	3
reducedErrorPruning	False
saveInstanceData	False
seed	1
subtreeRaising	True
unpruned	False
useLaplace	False
useMDLcorrection	True

Figure 7:Change minNumObj to 6.



```
Number of Leaves  :     11

Size of the tree :      21


Time taken to build model: 0.02 seconds

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

Correctly Classified Instances         471               90.5769 %
Incorrectly Classified Instances        49                9.4231 %
Kappa statistic                          0.8026
Mean absolute error                      0.1298
Root mean squared error                  0.2747
Relative absolute error                 27.4089 %
Root relative squared error             56.455  %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.909    0.100    0.936      0.909    0.922      0.803    0.943     0.964     Positive
                 0.900    0.091    0.861      0.900    0.880      0.803    0.943     0.870     Negative
Weighted Avg.    0.906    0.096    0.907      0.906    0.906      0.803    0.943     0.928

=== Confusion Matrix ===

   a   b   <-- classified as
 291  29 |   a = Positive
  20 180 |   b = Negative
```

Figure 8: Result After Change minNumObj in Decision Tree.

Notice the changes:

- The Number of Leaves: 11, Size of the tree: 21
- From the figure, it is shown that the correctly classified instances are 90.5769 %
- The Result Found from the Confusion Matrix: TP =291, FN =29, FP = 20 and TN = 180.

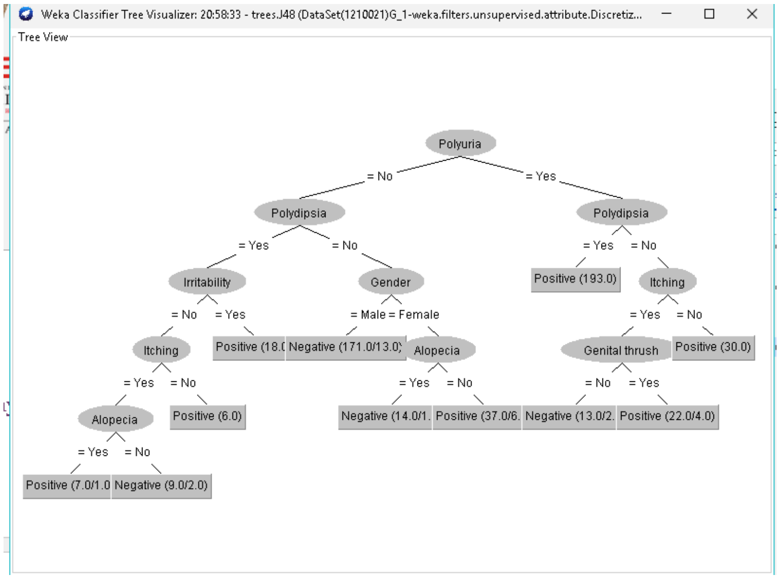


Figure 9: Tree after changing minNumObj.

As the precision is 0.936, recall 0.909, F-Measure = 0.922.

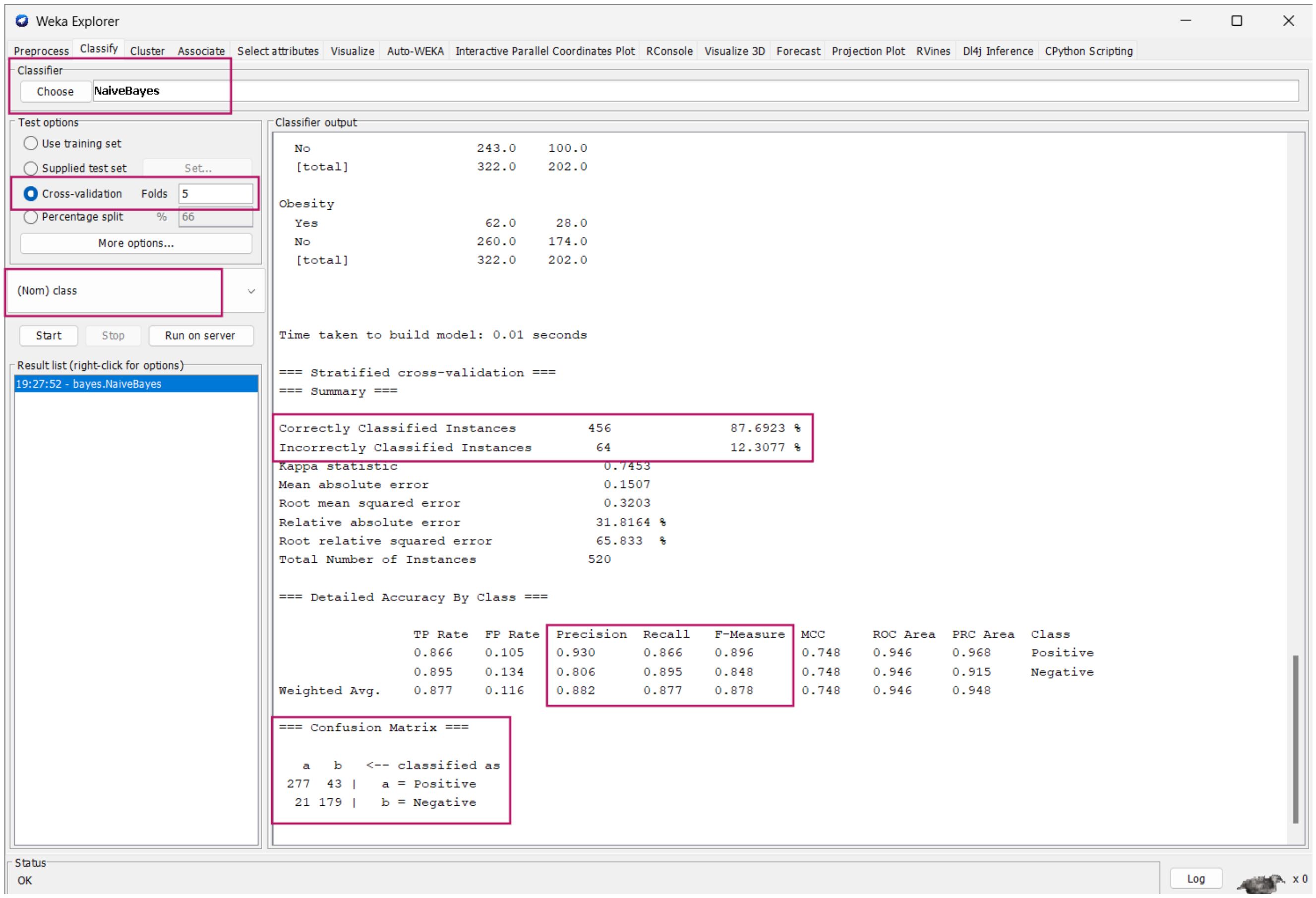
These results could be acceptable.

Notice when increase the number of (minNumObj) the correctly classified instances are will decrease and this will effect on the Confusion Matrix.

Naïve Bayes

5-folds 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 below.



Classifier: NaiveBayes

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 5

○ Percentage split % 66

(Nom) class

Result list (right-click for options)

19:27:52 - bayes.NaiveBayes

Status: OK

Classifier output

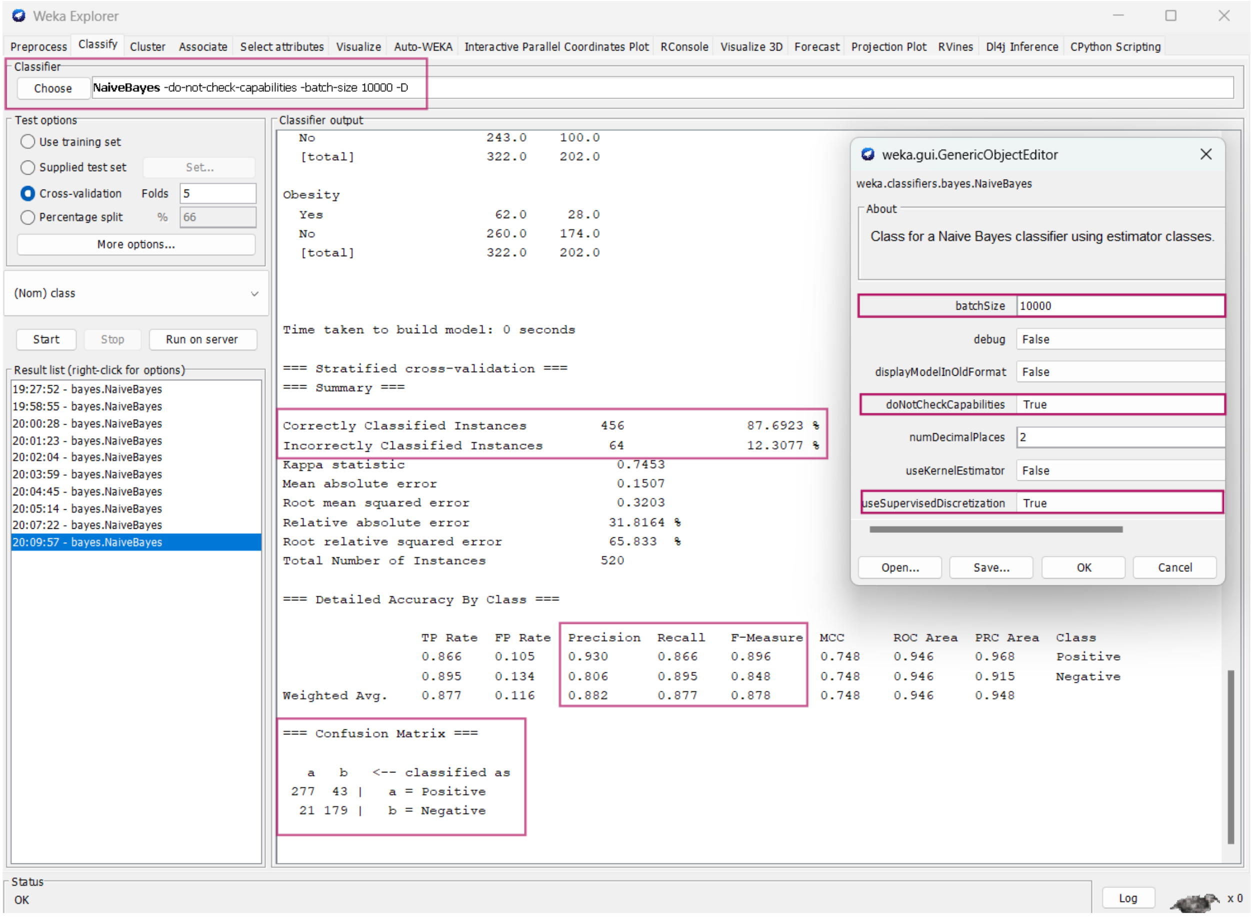
```
  No                     243.0    100.0
  [total]                322.0    202.0

Obesity
  Yes                     62.0     28.0
  No                     260.0    174.0
  [total]                322.0    202.0


Time taken to build model: 0.01 seconds

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

Correctly Classified Instances         456               87.6923 %
Incorrectly Classified Instances        64               12.3077 %
Kappa statistic                          0.7453
Mean absolute error                      0.1507
Root mean squared error                  0.3203
Relative absolute error                 31.8164 %
Root relative squared error             65.833  %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.866    0.105    0.930      0.866    0.896      0.748    0.946     0.968     Positive
                 0.895    0.134    0.806      0.895    0.848      0.748    0.946     0.915     Negative
Weighted Avg.    0.877    0.116    0.882      0.877    0.878      0.748    0.946     0.948

=== Confusion Matrix ===

   a   b   <-- classified as
 277  43 |   a = Positive
  21 179 |   b = Negative
```

Figure 10: The result of using Naive Bayes as Classifier.

From figure above, from confusion matrix part. The matrix illustrates that, TP =277, FN =43, FP =21, and TN = 179. From these values, precision, recall and f- measure values can be calculated. The value of the precision is 0.882, Recall is 0.877 and f-measure = 0.878 all in average, and they specify for both positive and negative class above. Also, from the figure the accuracy was equals 87.69%, Which is considered good.

Changing hyper-parameters.

Applying changing on batchSize from 100 to 10000, and the doNotCheckCapabilities to true, and the useSupervisedDiscretization to true.



Classifier: **NaiveBayes** -do-not-check-capabilities -batch-size 10000 -D

Cross-validation Folds: 5

Percentage split %: 66

(Nom) class

Result list:

- 19:27:52 - bayes.NaiveBayes
- 19:58:55 - bayes.NaiveBayes
- 20:00:28 - bayes.NaiveBayes
- 20:01:23 - bayes.NaiveBayes
- 20:02:04 - bayes.NaiveBayes
- 20:03:59 - bayes.NaiveBayes
- 20:04:45 - bayes.NaiveBayes
- 20:05:14 - bayes.NaiveBayes
- 20:07:22 - bayes.NaiveBayes
- 20:09:57 - bayes.NaiveBayes

```
  No                      243.0    100.0
  [total]                 322.0    202.0

Obesity
  Yes                      62.0     28.0
  No                      260.0    174.0
  [total]                 322.0    202.0



Time taken to build model: 0 seconds

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

Correctly Classified Instances         456               87.6923 %
Incorrectly Classified Instances        64               12.3077 %
Kappa statistic                          0.7453
Mean absolute error                      0.1507
Root mean squared error                  0.3203
Relative absolute error                 31.8164 %
Root relative squared error             65.833  %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.866    0.105    0.930      0.866    0.896      0.748    0.946     0.968     Positive
                 0.895    0.134    0.806      0.895    0.848      0.748    0.946     0.915     Negative
Weighted Avg.    0.877    0.116    0.882      0.877    0.878      0.748    0.946     0.948

=== Confusion Matrix ===

   a   b   <-- classified as
 277  43 |   a = Positive
  21 179 |   b = Negative
```

weka.classifiers.bayes.NaiveBayes

About: Class for a Naive Bayes classifier using estimator classes.

batchSize	10000
debug	False
displayModelInOldFormat	False
doNotCheckCapabilities	True
numDecimalPlaces	2
useKernelEstimator	False
useSupervisedDiscretization	True

Status: OK

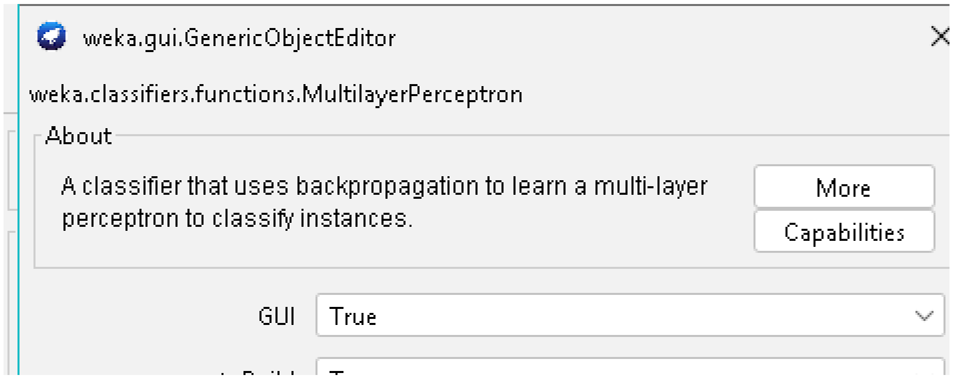
Figure 11:The result after change hyper-parameters.

The values after changing hyper-parameters does not change from the values before changing hyper-parameters as shown in the figure above.

MLP

5-folds cross validation.

To apply the 5 –fold cross validation on the MLP the setup will be:



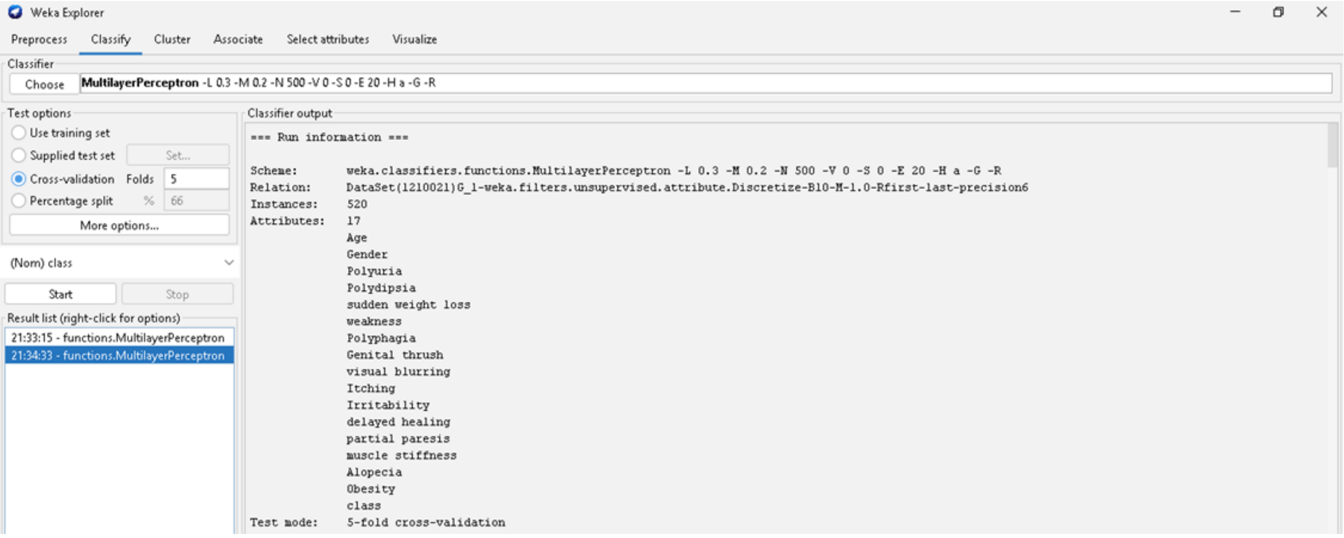
weka.classifiers.functions.MultilayerPerceptron

A classifier that uses backpropagation to learn a multi-layer perceptron to classify instances.

GUI True

Figure 12: Setup for applying 5-fold cross validation.

When starting the classify, the following is the result:



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

● Cross-validation Folds 5

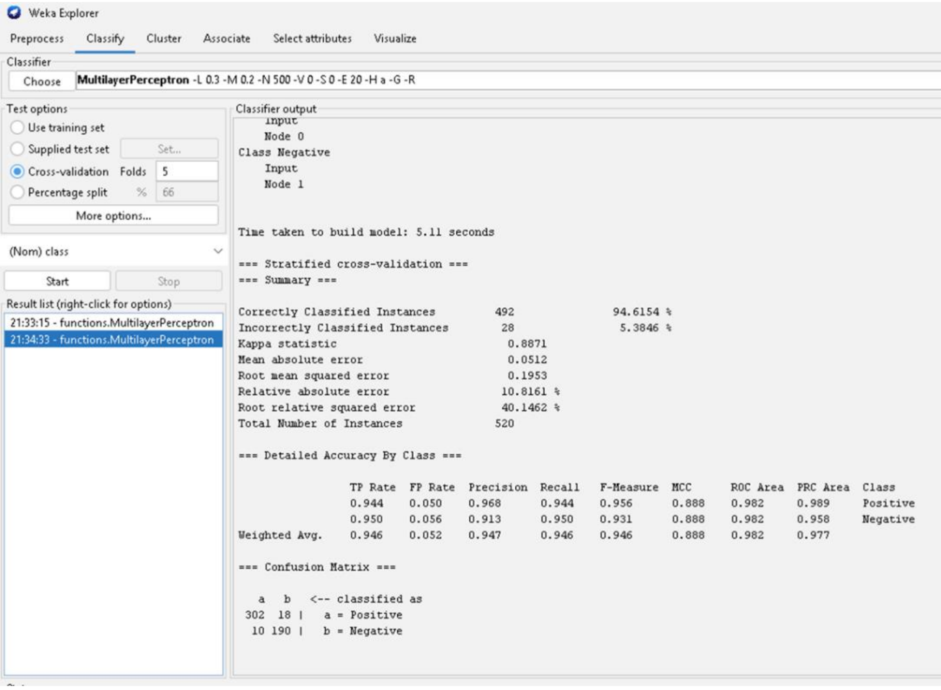
○ Percentage split % 66

(Nom) class

21:33:15 - functions.MultilayerPerceptron
21:34:33 - functions.MultilayerPerceptron

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R
Relation:     DataSet(1210021)G_1-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last-precision6
Instances:    520
Attributes:   17
              Age
              Gender
              Polyuria
              Polydipsia
              sudden weight loss
              weakness
              Polyphagia
              Genital thrush
              visual blurring
              Itching
              Irritability
              delayed healing
              partial paresis
              muscle stiffness
              Alopecia
              Obesity
              class
Test mode:    5-fold cross-validation
```

Figure 13: MLP 5-folds cross validation result part-1.



MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a -G -R

Cross-validation Folds 5

Percentage split % 66

(Nom) class

21:33:15 - functions.MultilayerPerceptron
21:34:33 - functions.MultilayerPerceptron

```
    Input
    Node 0
Class Negative
    Input
    Node 1


Time taken to build model: 5.11 seconds

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

Correctly Classified Instances         492               94.6154 %
Incorrectly Classified Instances        28                5.3846 %
Kappa statistic                          0.8871
Mean absolute error                      0.0512
Root mean squared error                  0.1953
Relative absolute error                 10.8161 %
Root relative squared error             40.1462 %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.944    0.050    0.968      0.944    0.956      0.888    0.982     0.989     Positive
                 0.950    0.056    0.913      0.950    0.931      0.888    0.982     0.958     Negative
Weighted Avg.    0.946    0.052    0.947      0.946    0.946      0.888    0.982     0.977

=== Confusion Matrix ===

   a   b   <-- classified as
 302  18 |   a = Positive
  10 190 |   b = Negative
```

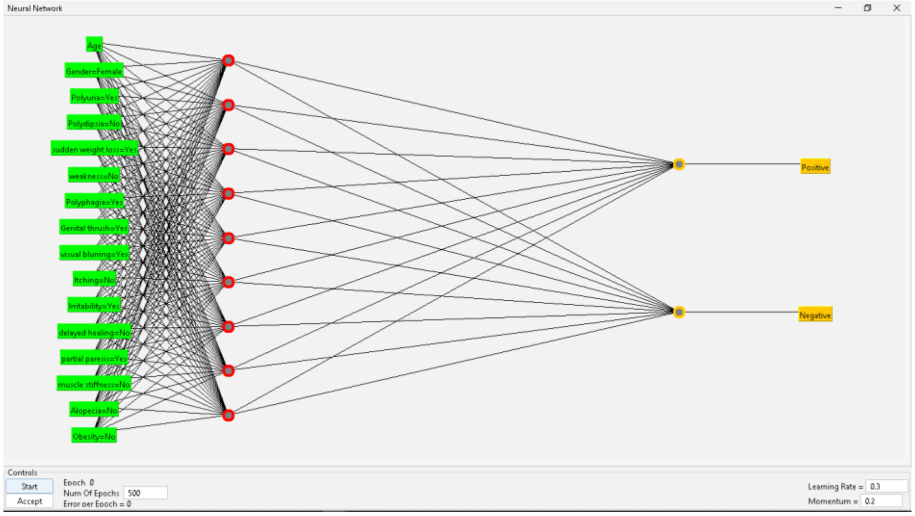
Figure 14: MLP 5-folds cross validation part-2.

From this result:

The classification model achieved an overall accuracy of 94.62%, correctly classifying 492 instances out of 520. Specifically, it accurately identified 302 instances of the 'Positive' class, while 10 instances were incorrectly classified as 'Negative'. Conversely, for the 'Negative' class, the model correctly classified 190 instances, with 18 instances misclassified as 'Positive'. These results reflect a high level of precision and recall for both classes, indicating robust performance in distinguishing between the classes based on the given matrix: TP =302, FN =18, FP = 10 and TN = 190.

So The Result Could Be acceptable.

To show the NN set the GUI True the Neural Network:

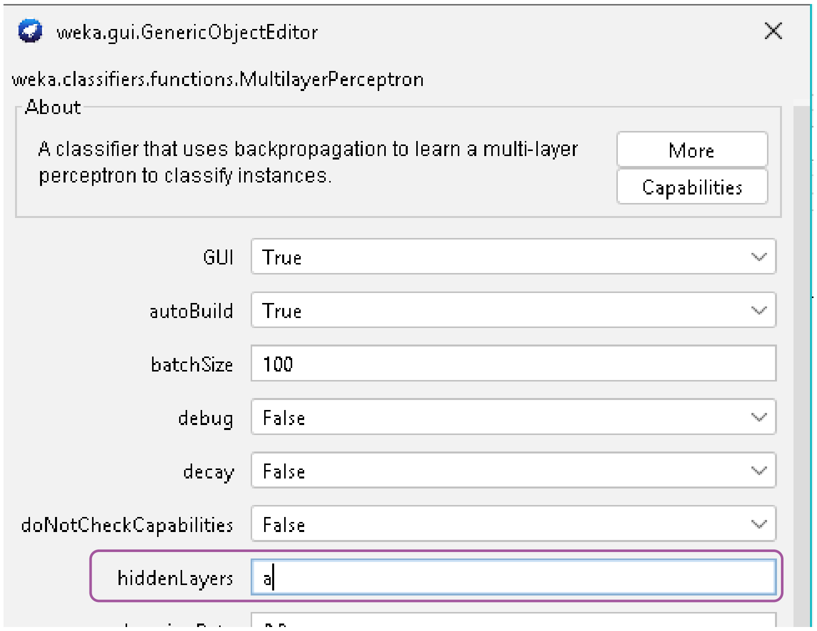


Epoch 0
Num Of Epochs 500
Error per Epoch = 0
Learning Rate = 0.3
Momentum = 0.2

Figure 15: Neural Network.

Changing hyper-parameters

By Changing the hidden layers From **a →2**



weka.gui.GenericObjectEditor

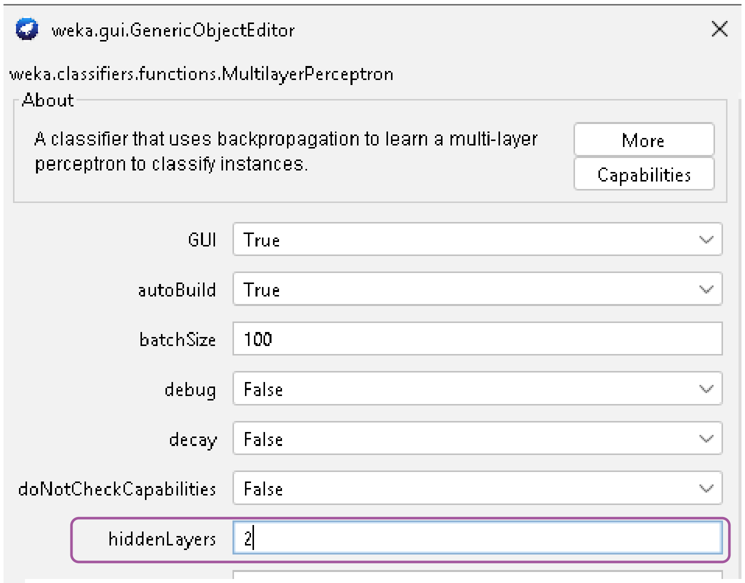
weka.classifiers.functions.MultilayerPerceptron

About

A classifier that uses backpropagation to learn a multi-layer perceptron to classify instances.

GUI	True
autoBuild	True
batchSize	100
debug	False
decay	False
doNotCheckCapabilities	False
hiddenLayers	a

Figure 16: Before changing the parameter.



weka.gui.GenericObjectEditor

weka.classifiers.functions.MultilayerPerceptron

About

A classifier that uses backpropagation to learn a multi-layer perceptron to classify instances.

GUI	True
autoBuild	True
batchSize	100
debug	False
decay	False
doNotCheckCapabilities	False
hiddenLayers	2

Figure 17: After changing the parameter.

The NN become:

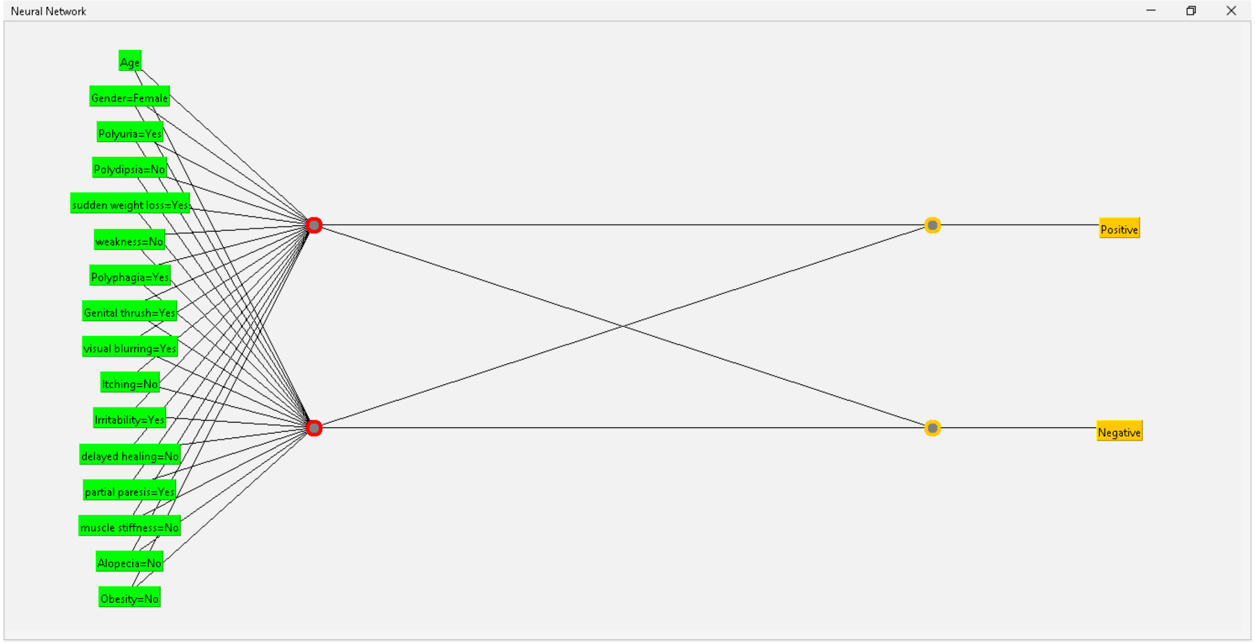
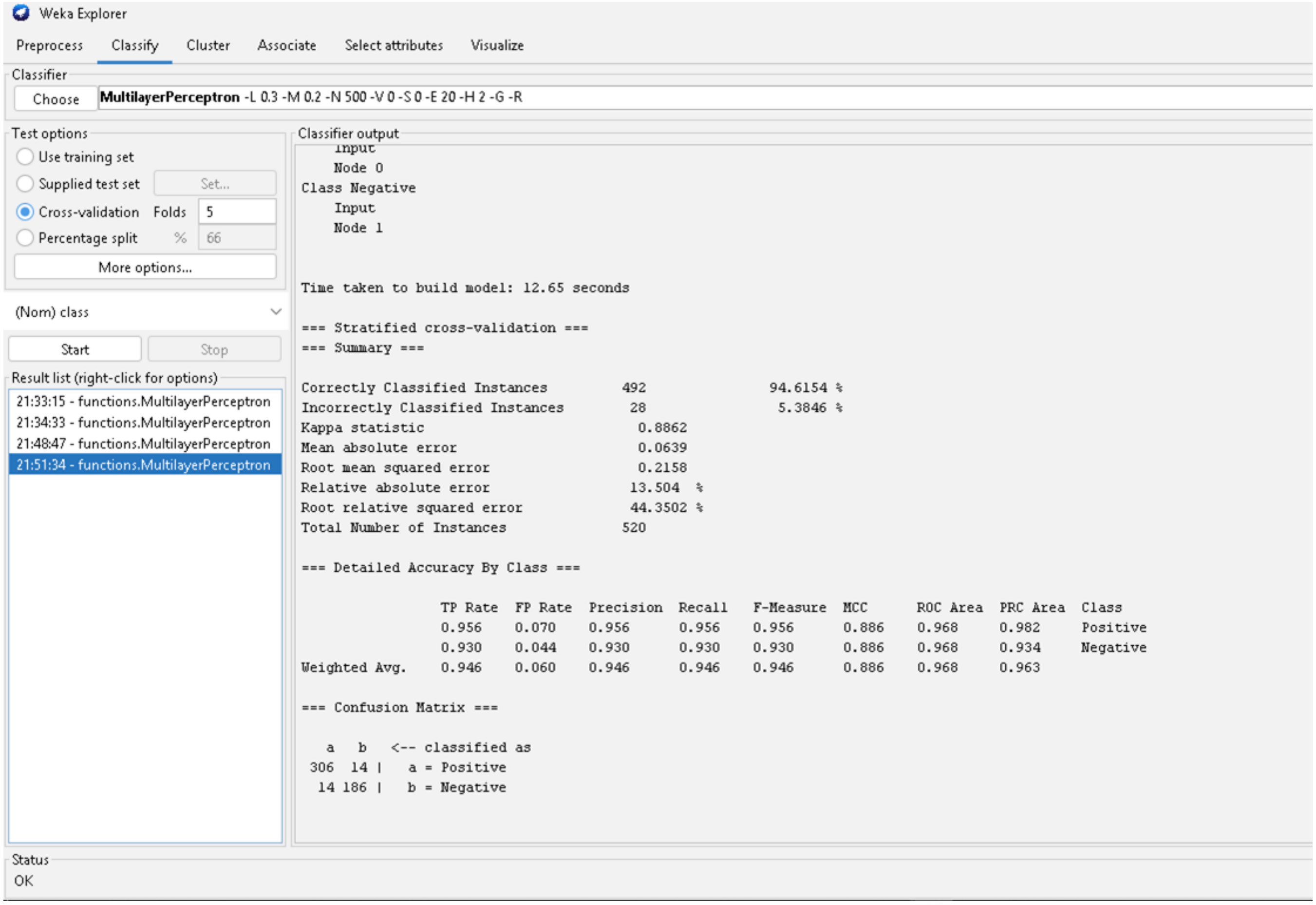


Figure 18: Neural Network After Changing.



Choose **MultilayerPerceptron** -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H 2 -G -R

- ○ Use training set
- ○ Supplied test set
- ● Cross-validation Folds 5
- ○ Percentage split % 66

(Nom) class

Result list (right-click for options)

- 21:33:15 - functions.MultilayerPerceptron
- 21:34:33 - functions.MultilayerPerceptron
- 21:48:47 - functions.MultilayerPerceptron
- 21:51:34 - functions.MultilayerPerceptron

```
    Input
    Node 0
Class Negative
    Input
    Node 1


Time taken to build model: 12.65 seconds

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

Correctly Classified Instances         492               94.6154 %
Incorrectly Classified Instances        28                5.3846 %
Kappa statistic                          0.8862
Mean absolute error                      0.0639
Root mean squared error                  0.2158
Relative absolute error                 13.504  %
Root relative squared error             44.3502 %
Total Number of Instances              520

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.956    0.070    0.956      0.956    0.956      0.886    0.968     0.982     Positive
                 0.930    0.044    0.930      0.930    0.930      0.886    0.968     0.934     Negative
Weighted Avg.    0.946    0.060    0.946      0.946    0.946      0.886    0.968     0.963

=== Confusion Matrix ===

   a   b   <-- classified as
 306  14 |   a = Positive
  14 186 |   b = Negative
```

Status
OK

Figure 19: The result of using MLP as classifier after change.

From the result of classify:

The model achieved an overall accuracy of 94.62%, correctly classifying 492 instances. Specifically, it accurately identified 306 instances of the 'Positive' class (indicating diabetes) and 186 instances of the 'Negative' class (indicating no diabetes). The model's performance metrics, including precision, recall, and F-measure, were consistently high for both classes, reflecting its effectiveness in distinguishing between individuals with and without diabetes based on the provided attributes.

TP =306, FN =14, FP = 14 and TN = 186.

As the precision is 0.956, recall 0.956, F-Measure = 0.956. These results could be acceptable.

The effect of changing:

Both results show good accuracy in identifying diabetes. Result Two found slightly more cases of diabetes and had fewer misses, but result one made fewer mistakes by falsely saying someone had diabetes when they didn't. Choosing between them would depend on whether it's more important to find all cases of diabetes or to avoid saying someone has diabetes when they don't.

conclusion

Among the three classifiers, the optimized Multi-Layer Perceptron (MLP) demonstrated the highest performance in terms of accuracy, precision, recall, and F-measure, making it the most effective model for this diabetes classification task. The Decision Tree also showed strong performance, especially in terms of precision, but was more sensitive to hyper-parameter adjustments. Naïve Bayes, while stable, had slightly lower performance metrics compared to the other two methods. Overall, the choice of classifier can depend on the specific requirements of the task at hand. If achieving the highest possible accuracy and balanced performance is crucial, the MLP is the preferred choice. For scenarios where interpretability and stability are important, the Decision Tree and Naïve Bayes classifiers offer viable alternatives.

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 67.77% and the F-measure equals 0.753. In the last algorithm 'The Hoeffding Tree algorithm", the results were good, since the accuracy was equal to 85.33% and the F-measure to 0.858, even after playing with the parameters, the results were still good.