Classifying United States congressional voting result based on different attributes to predict future voters

Objective:

We have 1984 United States congressional voting records dataset called 'vote'. The output attribute named class have two values: - 'Democrat' & 'Republic'. There are also 17 attributes showing the characteristics of the voters. Using Weka data mining tool, our objective is to implement classification algorithms based on provided dataset to predict future voters.

We have divided our task into several steps.

In step-1, 'voter' dataset will be preprocessed to select 12 best attributes.

In step-2, 4 different classification algorithms will produce 8 models (per algorithm 2 models: 2.1 to 2.8). As well as all models will be applied on pre-processed data set from step-1. Also, all the models will be learned and tested by splitting the dataset in a training and a test dataset, each of which consisting of 75% and 25% of instances, respectively. Result for each model will be included in this step also.

In step-3, we will calculate the accuracy, precision, recall, sensitivity, and specificity for each model, for the class 'democrat'.

In step-4, we will choose the best, the second best and the third best model along with justification from step-2.

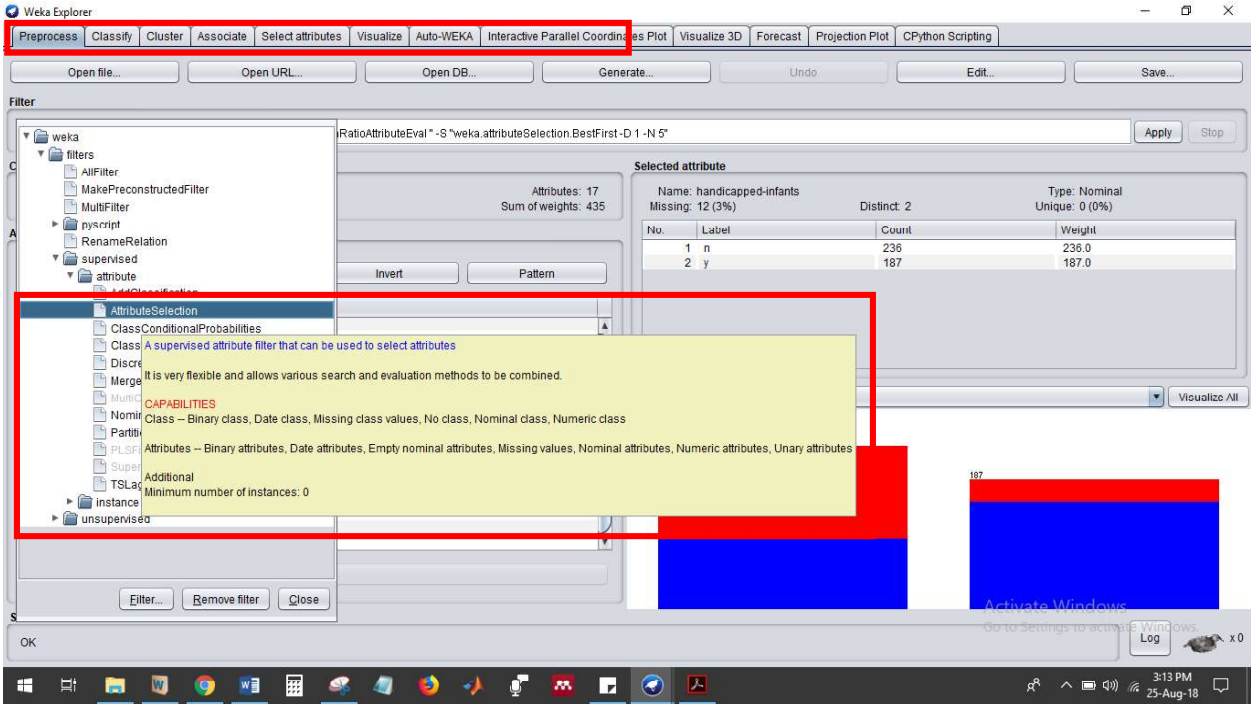
In step-5, we will mention three characteristics of 'democrat' voters as well as production rules of building those characteristics based on the decision tree built and displayed in step-2.1 & step-2.2.

Let's discuss these steps in detail.

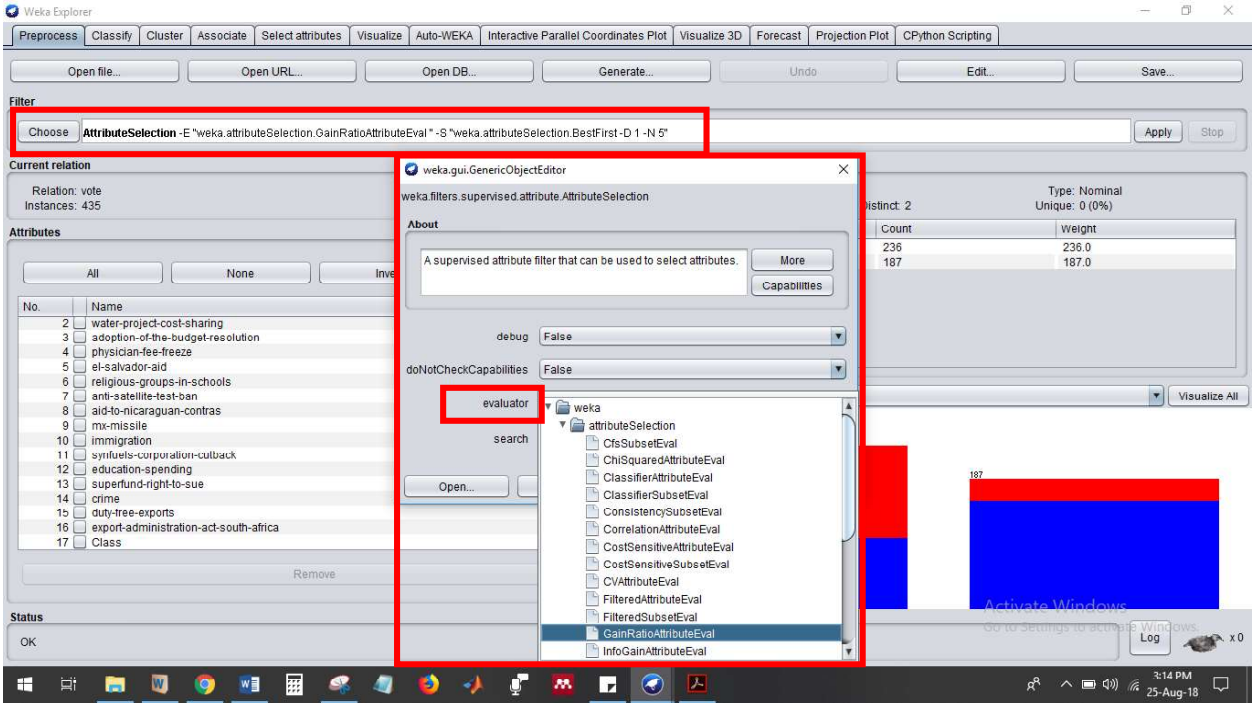
1) Step-1:

Here, in this step, dataset will be preprocessed to select 12 best attributes.

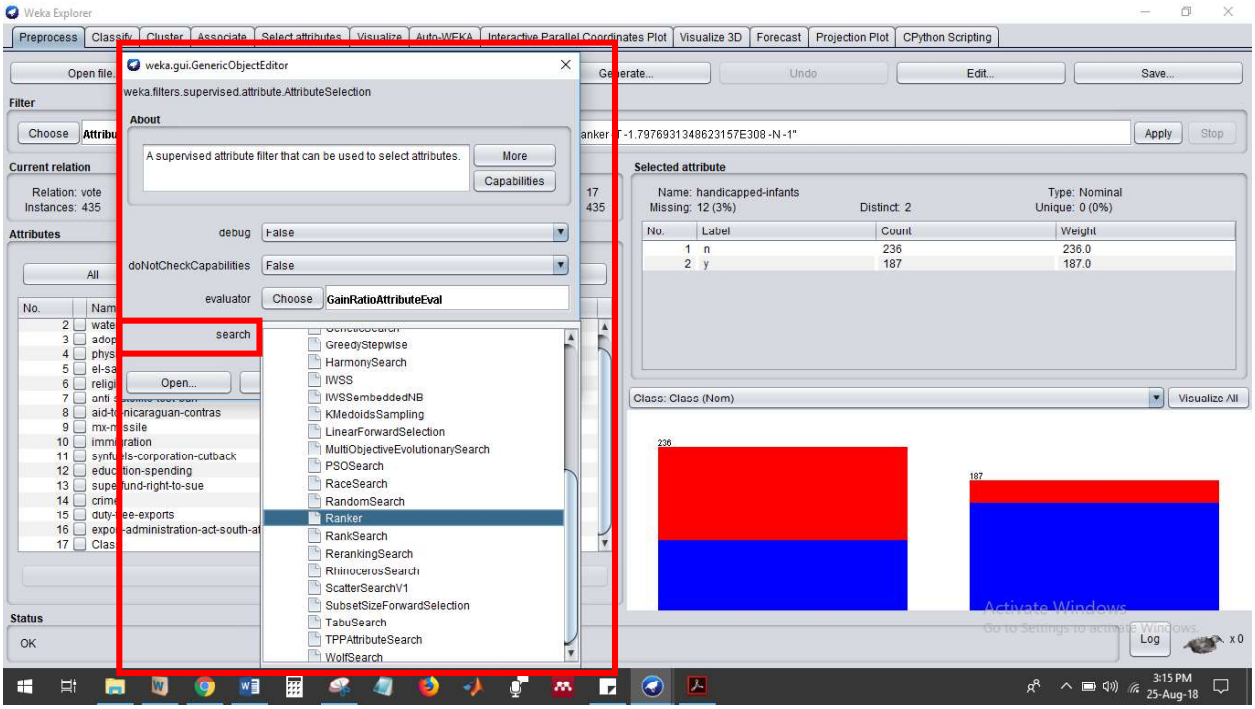
After opening data file in preprocess step, "***attributeSelection***" has been chosen as filter.



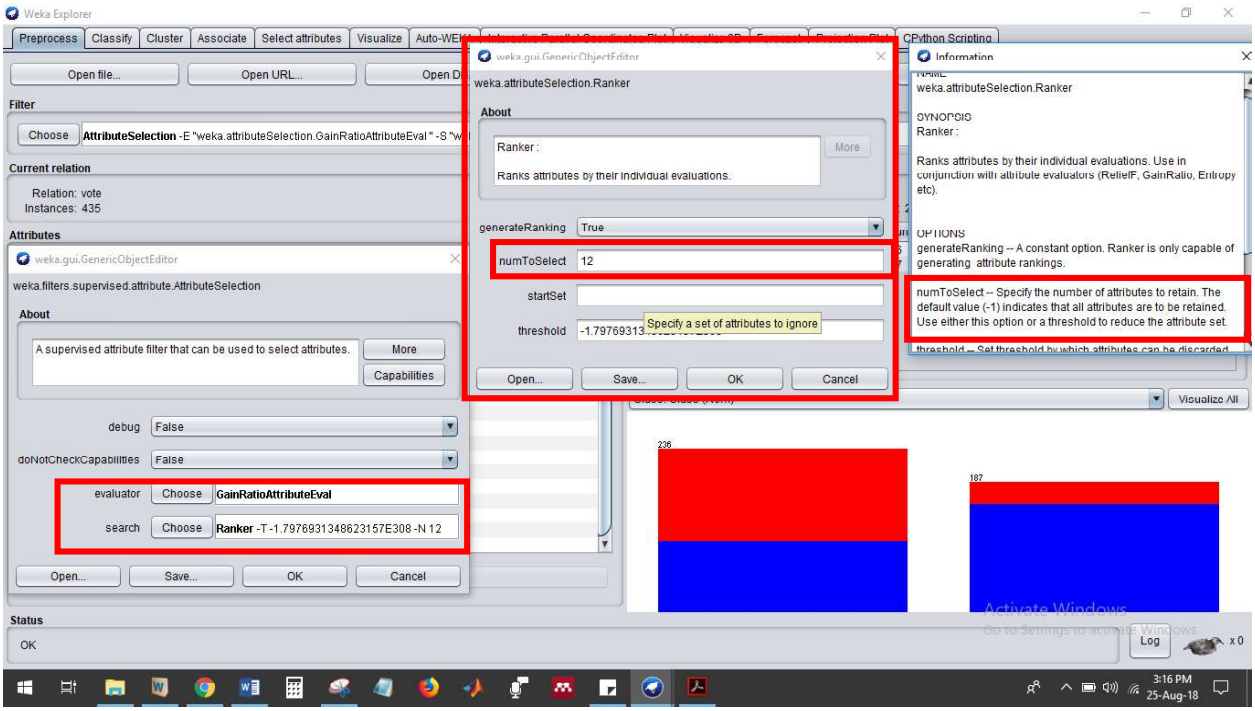
After that, "***GainRatioAttributeEval***" algorithm has been chosen as evaluator.



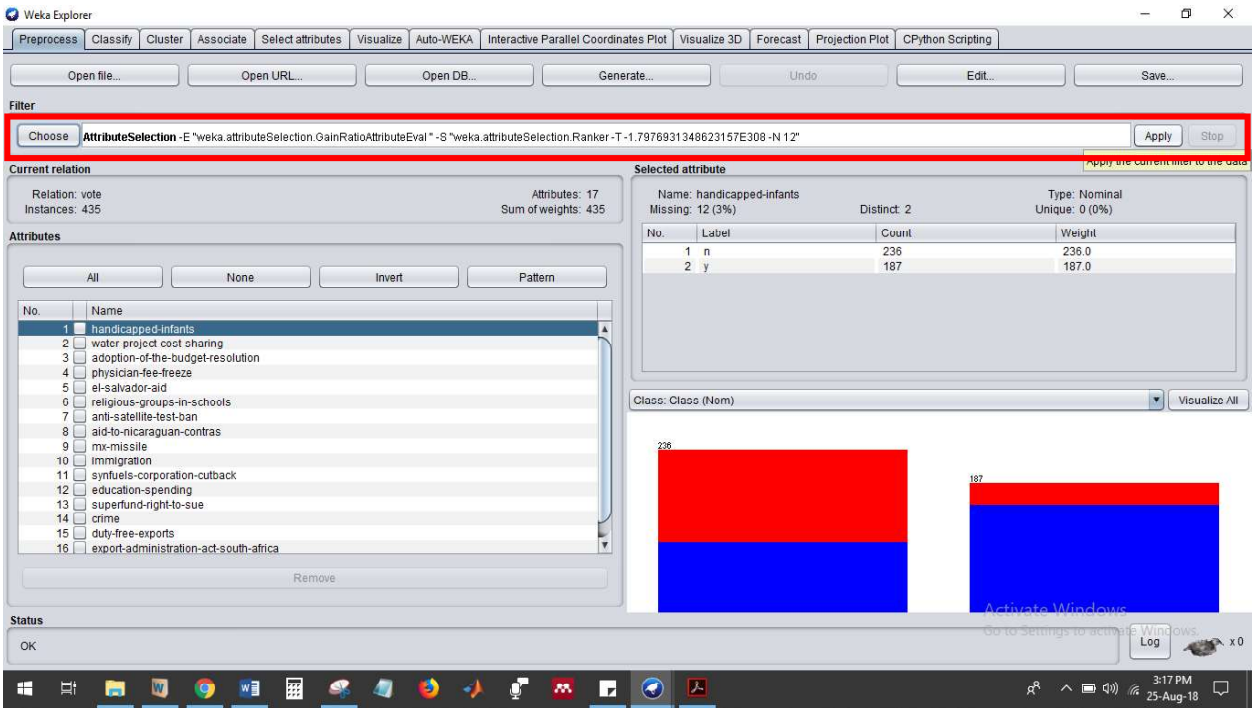
After that, "***Ranker***" has been chosen as search method.



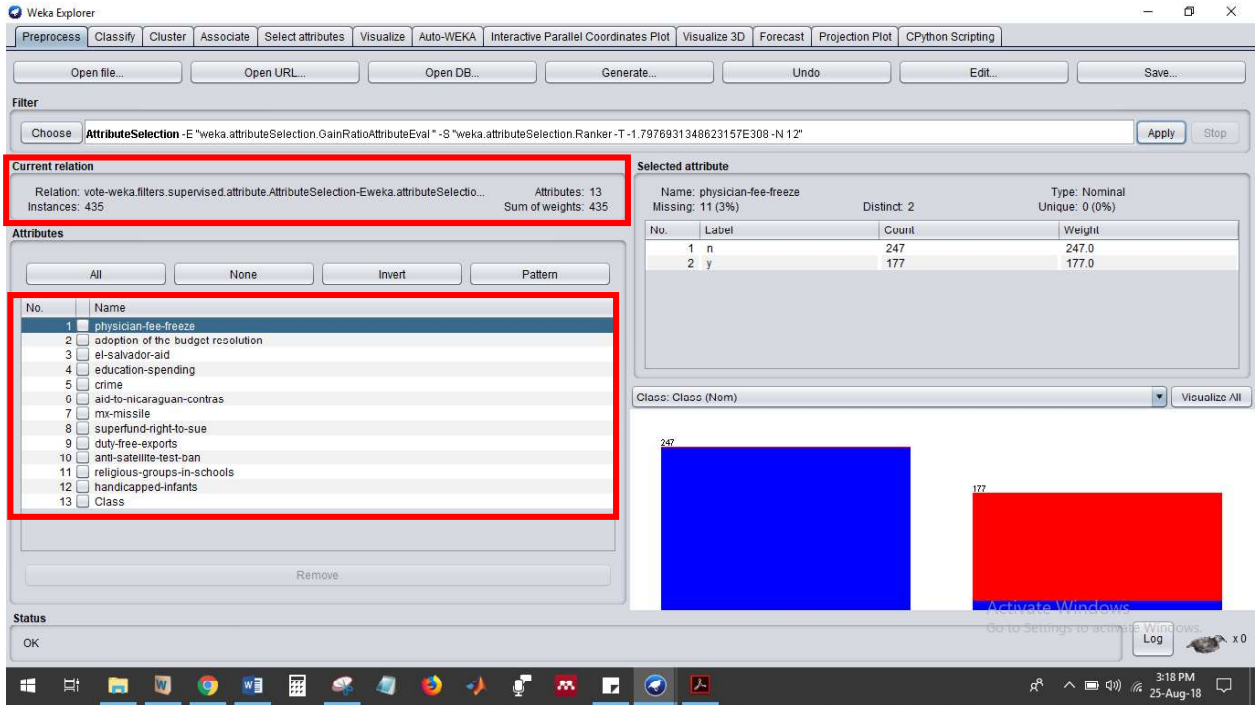
The parameter "***numToSelect***" in "***Ranker***" search has been chosen to 12, since, we have to select 12 best attributes except "***Class***".



Now, it is high time to apply this preprocessing filter on data set.



After applying the filter, now, we can see the best 12 attributes along with class attribute have been kept for next "***classify***" step.

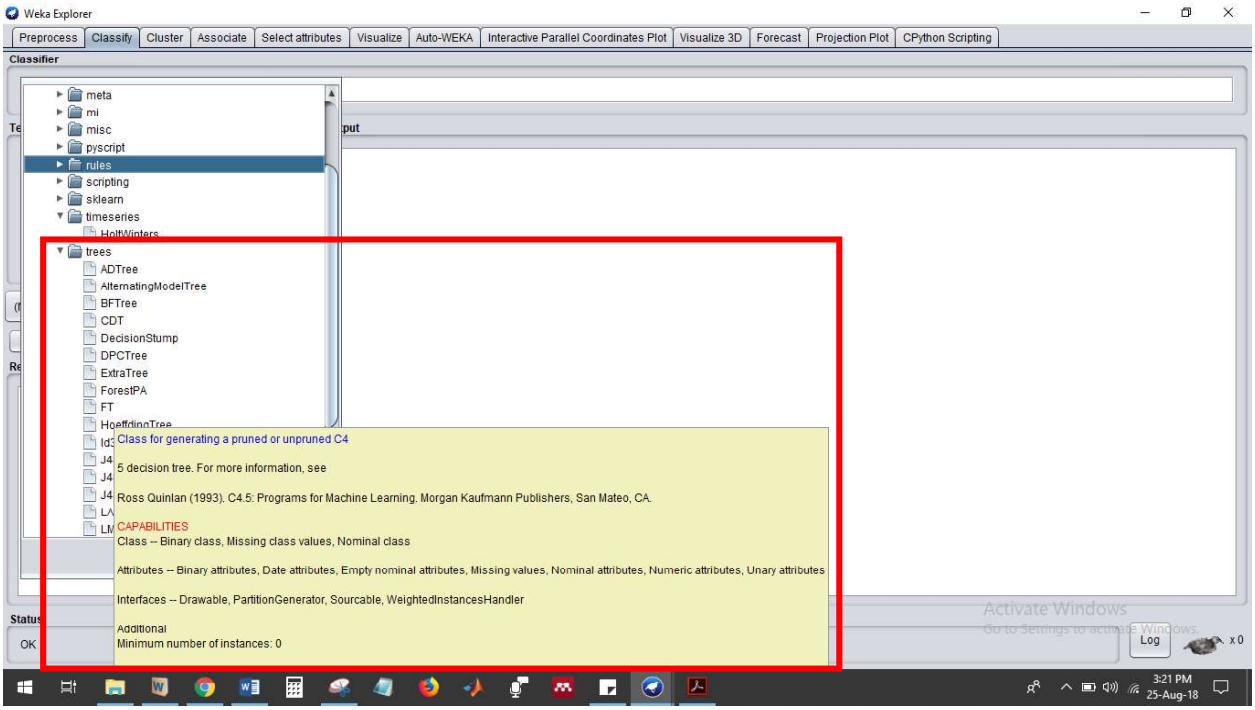


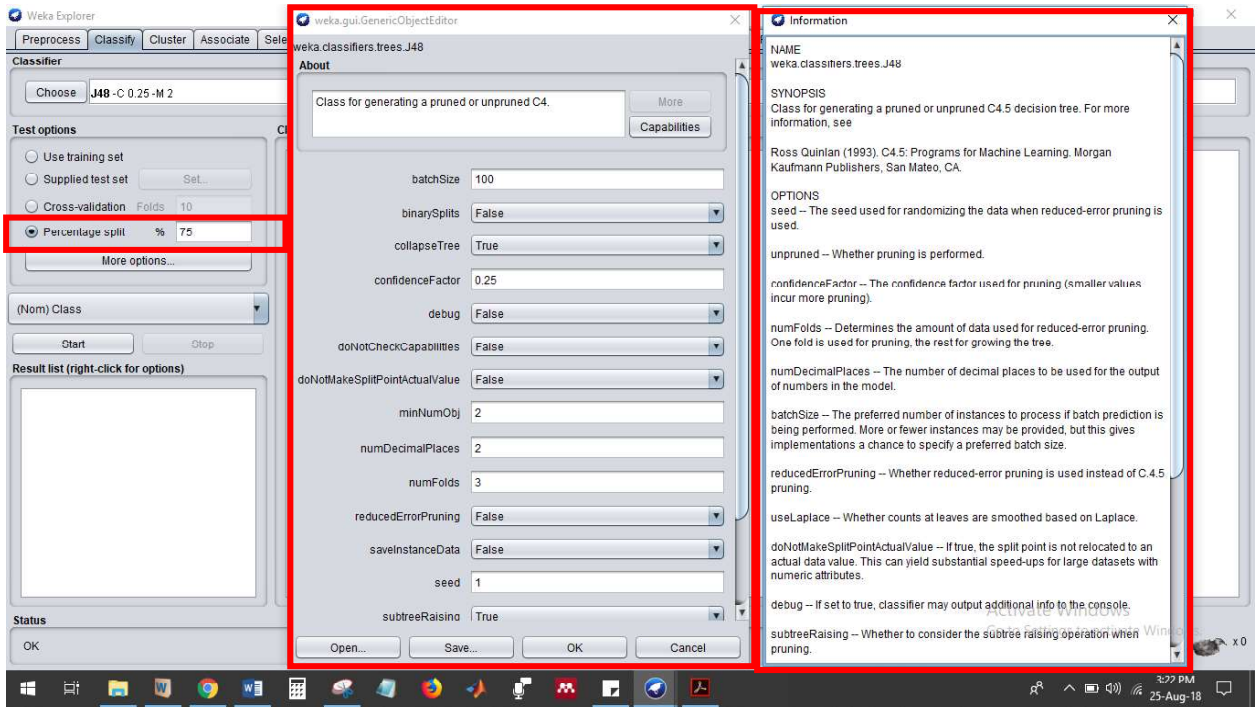
2) Step-2:

In this step, Classification algorithms will be applied on pre-processed data set.

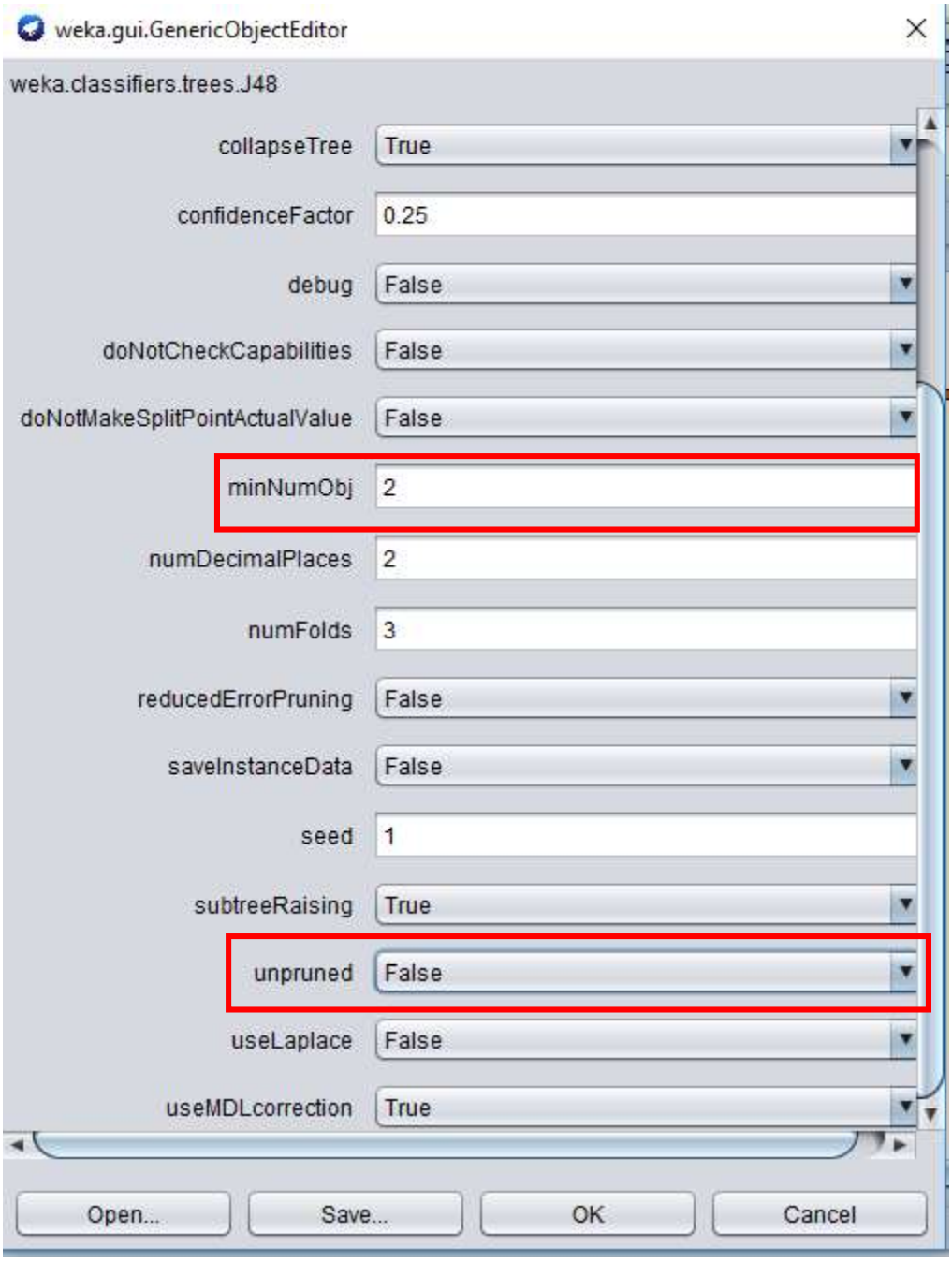
2.1 Decision tree: C4.5_Pruned tree

As classifier, Decision tree C4.5 (In Weka, it is J48) has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Different parameters setting have been given below.



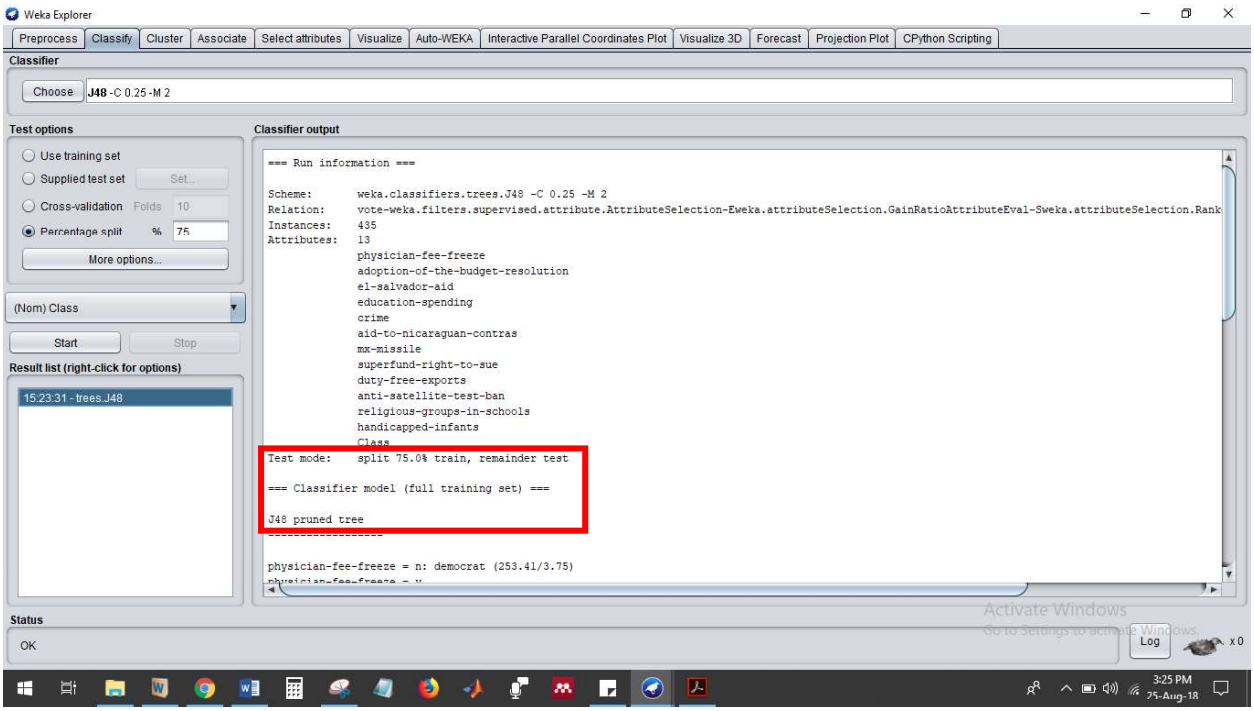


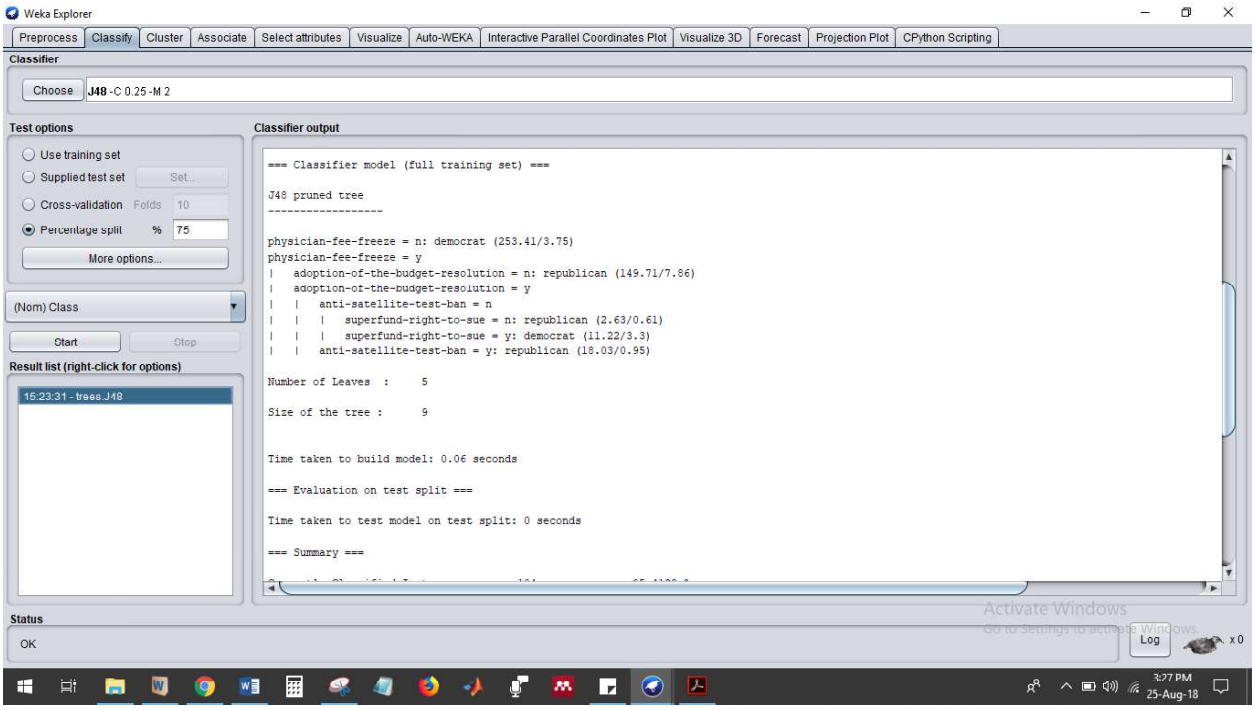
Here, "***minNumObj***" has been kept 2 & "***unpruned***" as False.

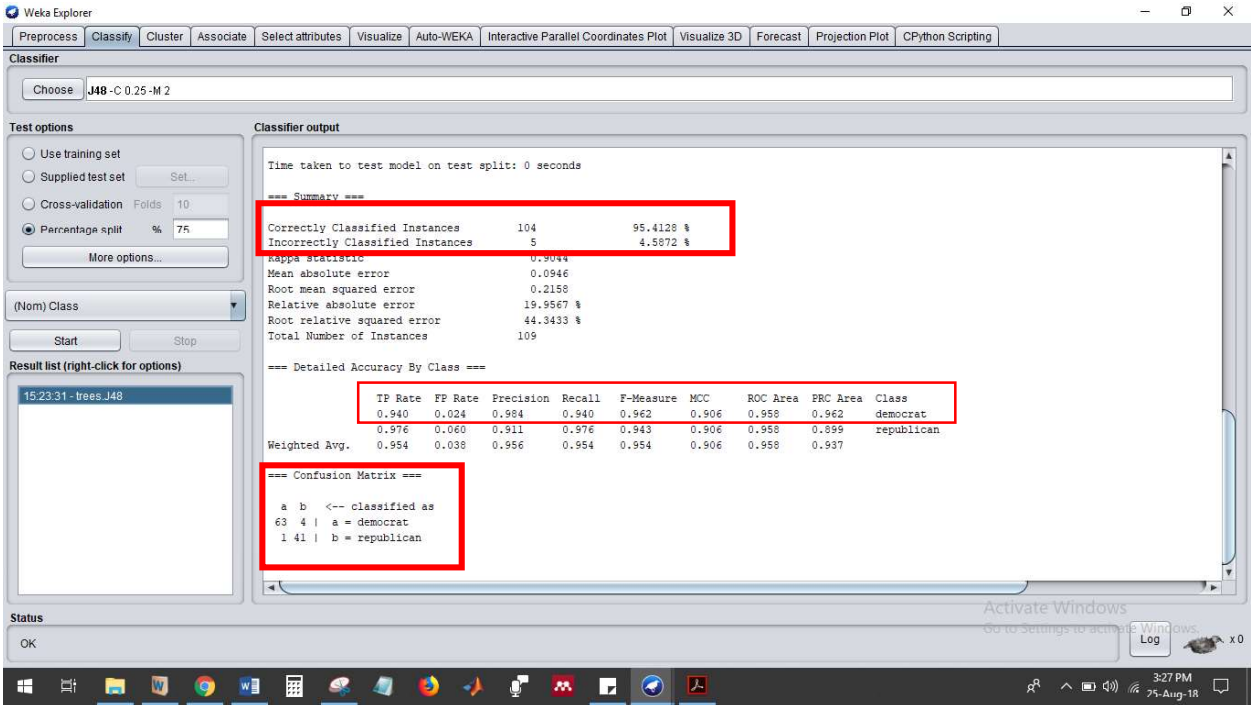


Result:

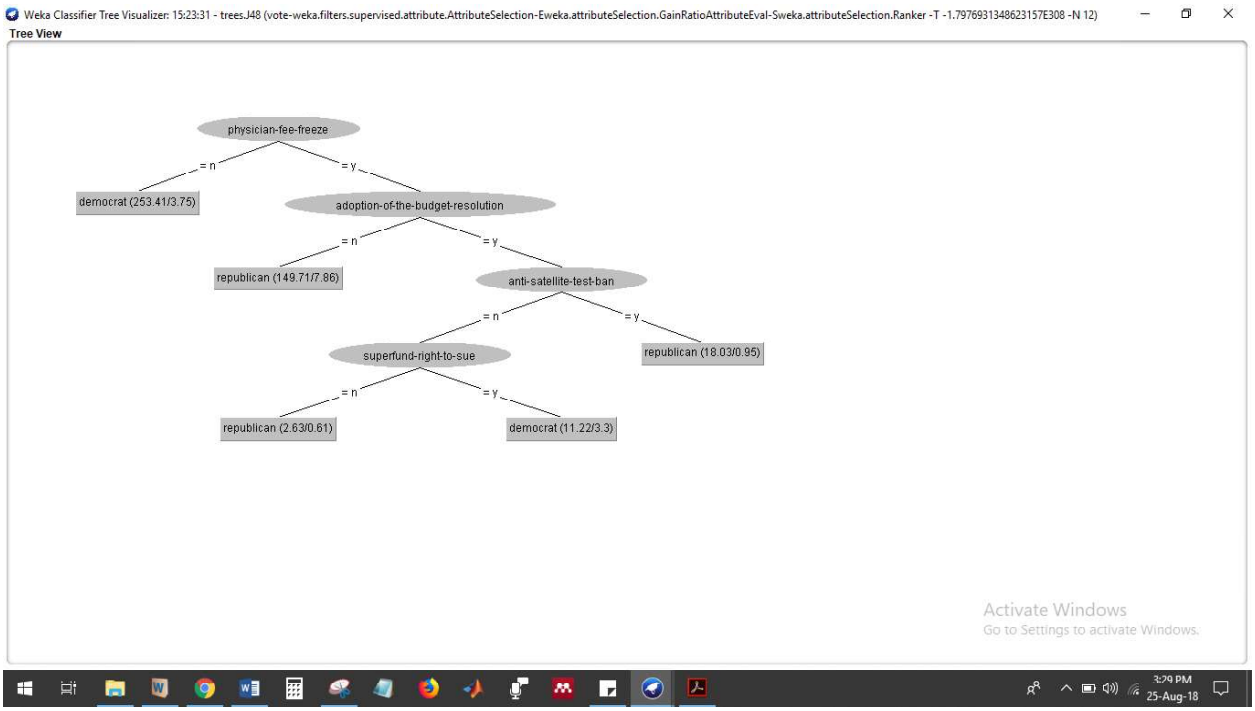
For C4.5 Pruned decision tree, the overall classification accuracy is **95.4128%**







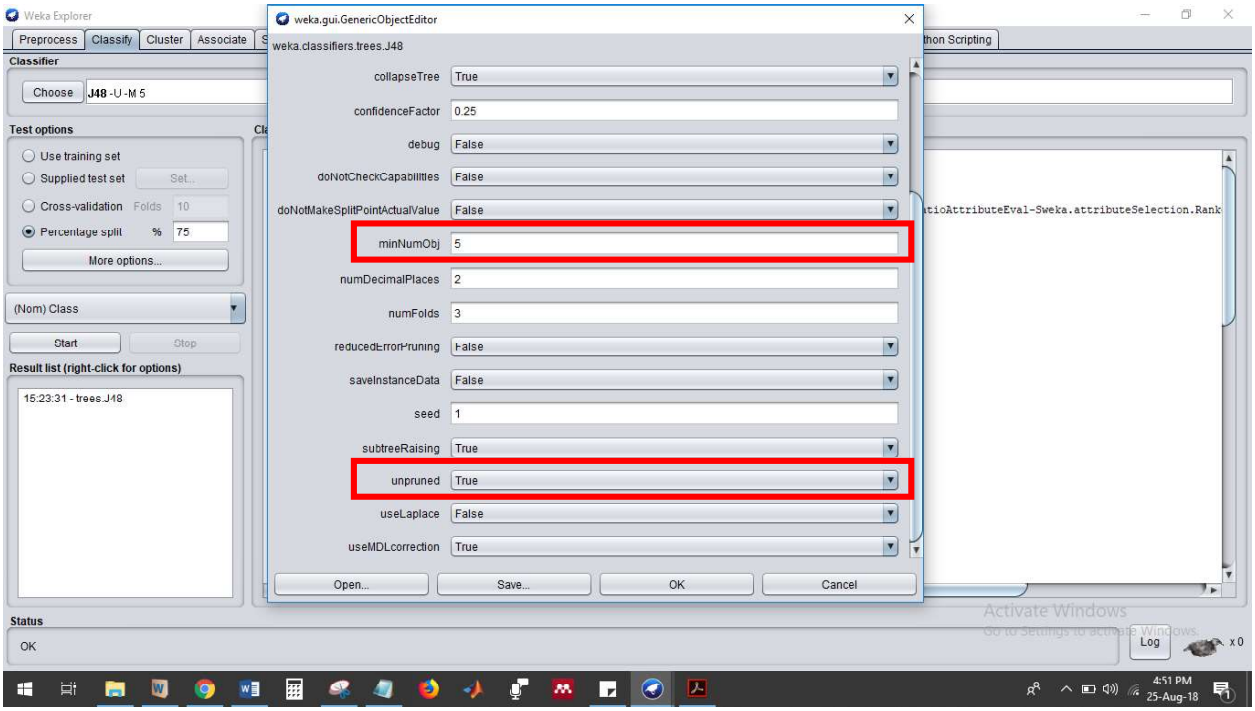
Here is the decision tree visualization.



2.2 Decision tree: C4.5_Unpruned tree

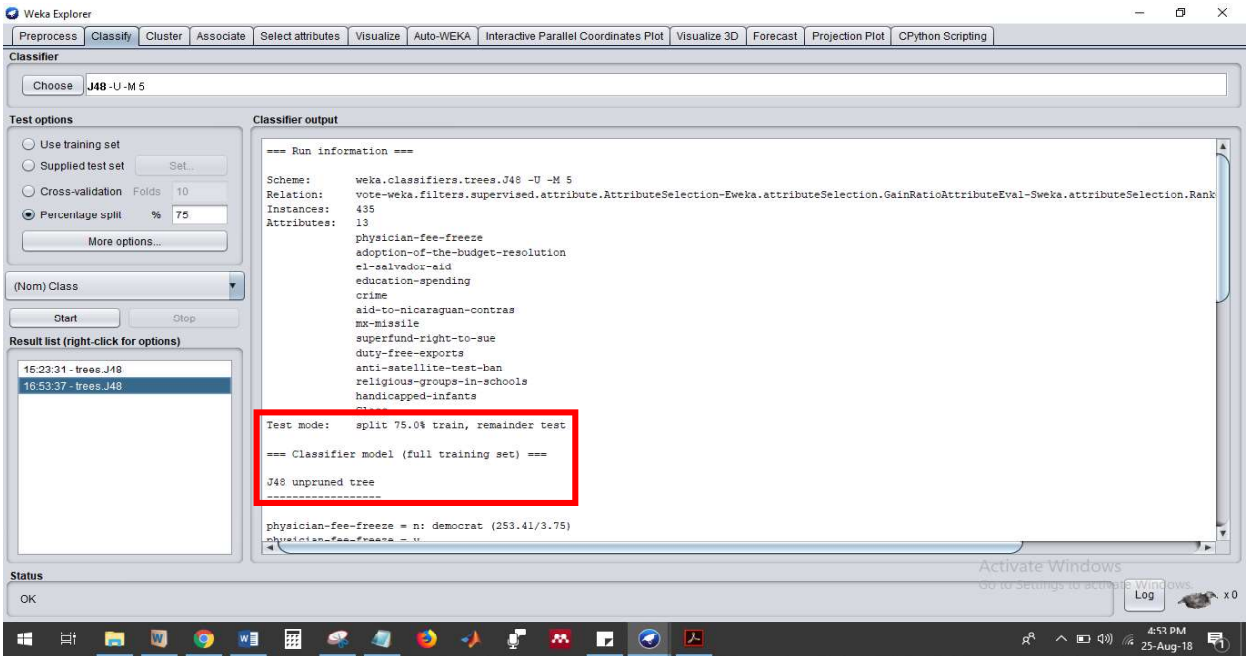
As classifier, Decision tree C4.5 (In Weka, it is J48) has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Different parameters setting have been given below.

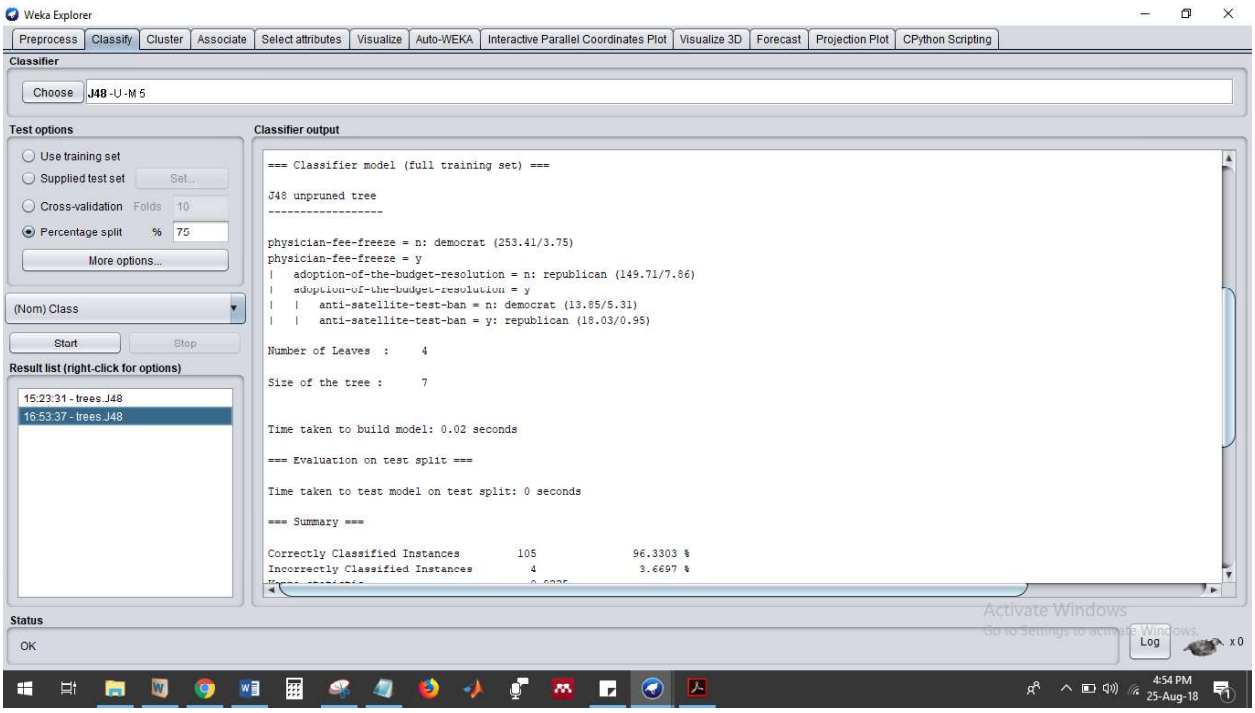
Here, "***minNumObj***" has been kept 5 & "***unpruned***" as True.

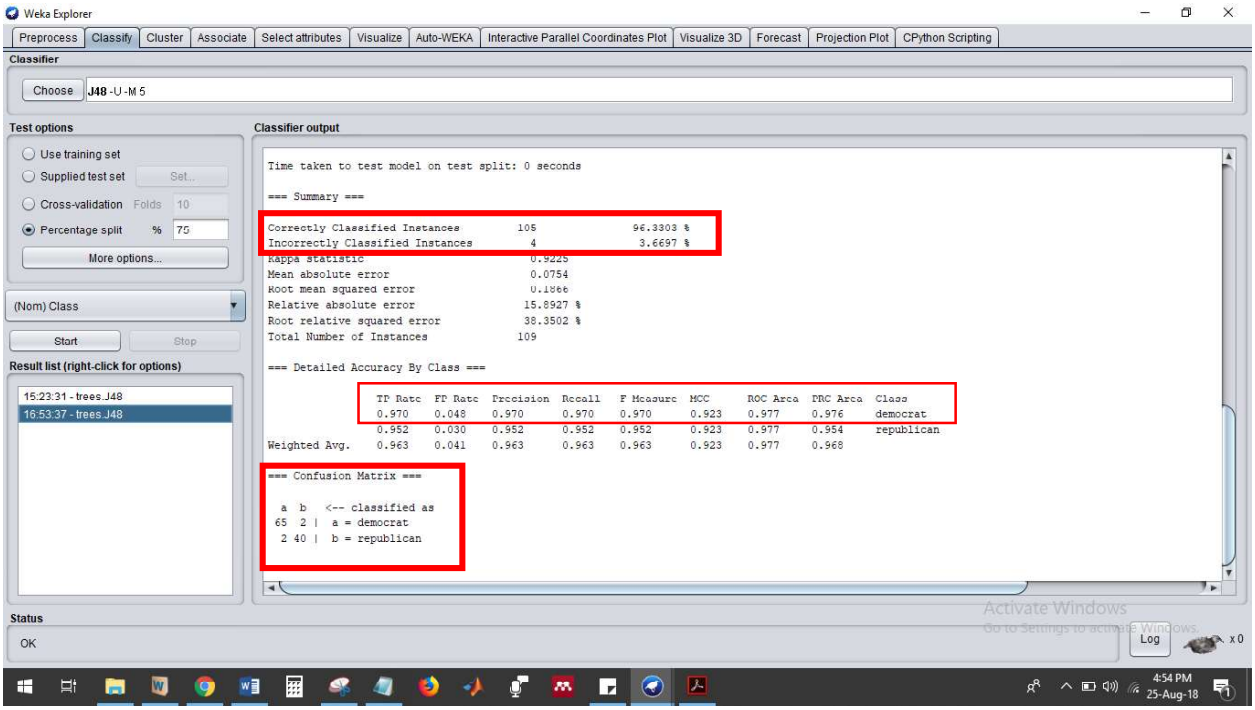


Result:

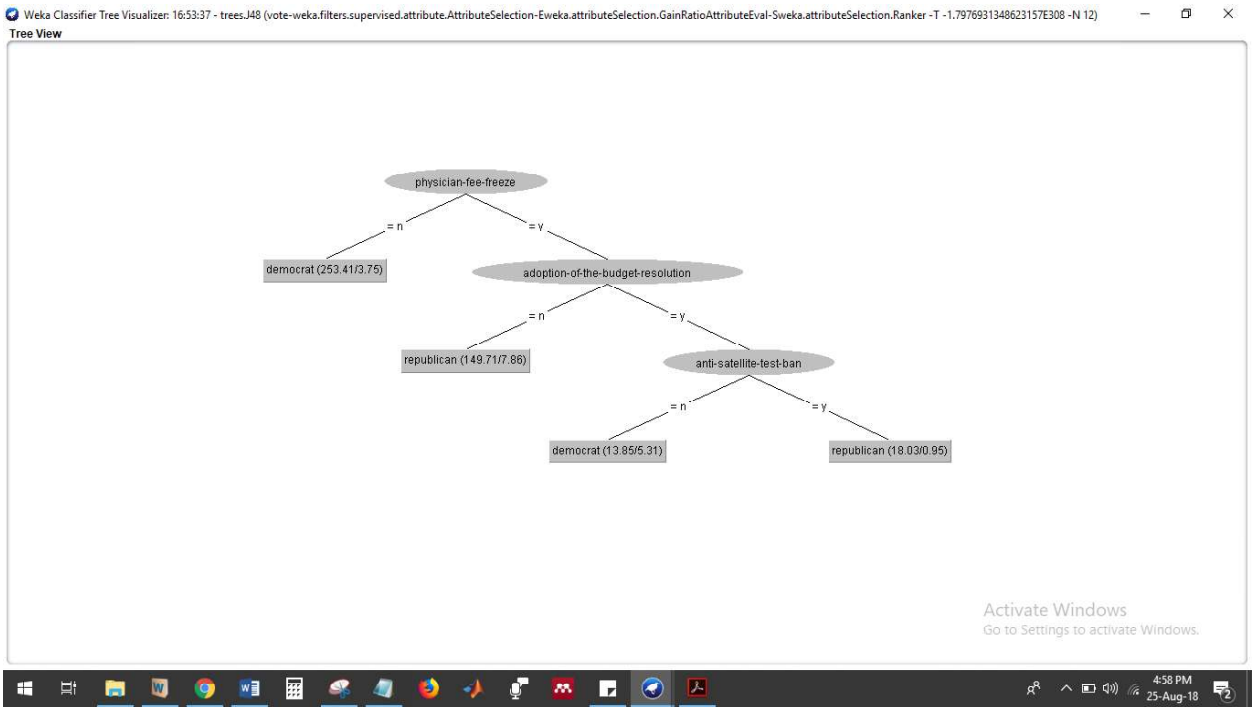
For C4.5 Unpruned decision tree, the overall classification accuracy is **96.3303%**




Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Auto-WEKA
Interactive Parallel Coordinates Plot
Visualize 3D
Forecast
Projection Plot
CPython Scripting
Classifier
Choose
J48 -U -M 5
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
75
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
15:23:31 - trees.J48
16:53:37 - trees.J48
Classifier output
=== Classifier model (full training set) ===
J48 unpruned tree
physician-fee-freeze = n: democrat (253.41/3.75)
physician-fee-freeze = y
| adoption-of-the-budget-resolution = n: republican (149.71/7.86)
| adoption-of-the-budget-resolution = y
| | anti-satellite-test-ban = n: democrat (13.85/5.31)
| | anti-satellite-test-ban = y: republican (18.03/0.95)
Number of Leaves : 4
Size of the tree : 7
Time taken to build model: 0.02 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 105 96.3303 %
Incorrectly Classified Instances 4 3.6697 %
Status
OK
Log
x 0
4:54 PM
25-Aug-18

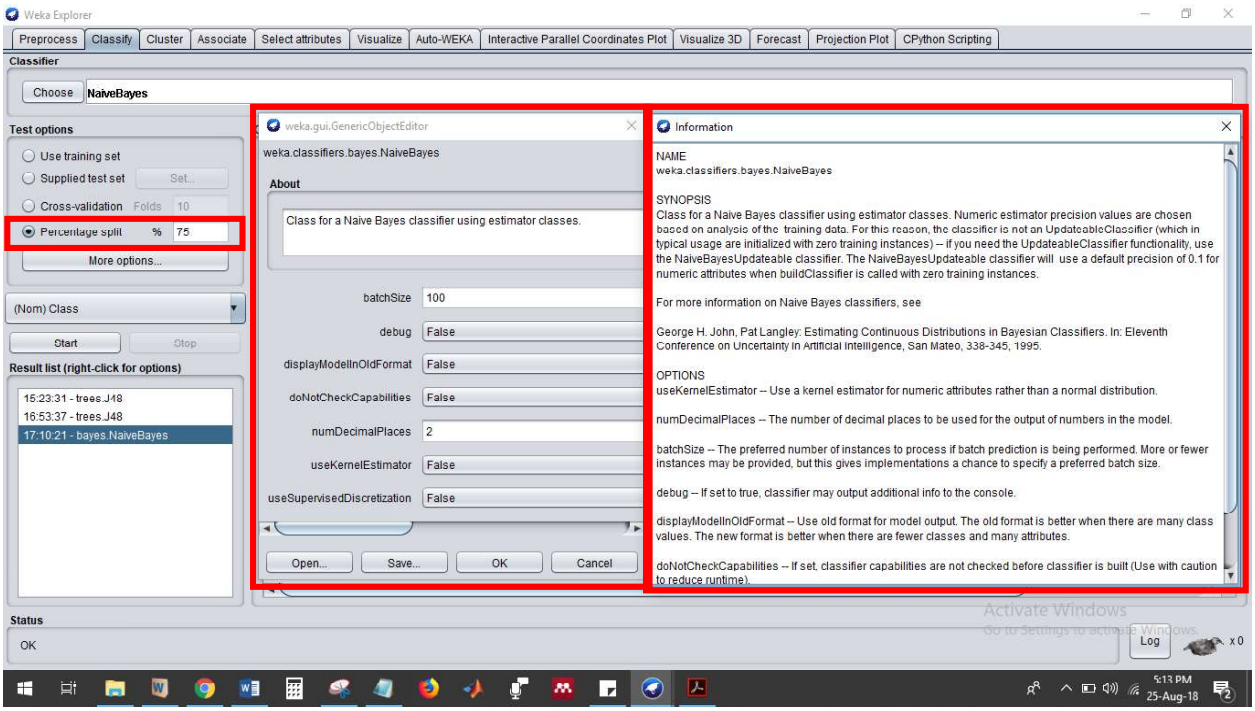

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Auto-WEKA
Interactive Parallel Coordinates Plot
Visualize 3D
Forecast
Projection Plot
CPython Scripting
Classifier
Choose
J48 -U -M 5
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
75
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
15:23:31 - trees.J48
16:53:37 - trees.J48
Classifier output
Time taken to test model on test split: 0 seconds
=== Summary ===
Correctly Classified Instances 105 96.3303 %
Incorrectly Classified Instances 4 3.6697 %
Kappa statistic 0.9225
Mean absolute error 0.0754
Root mean squared error 0.1866
Relative absolute error 15.8927 %
Root relative squared error 38.3502 %
Total Number of Instances 109
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.970 0.048 0.970 0.970 0.970 0.923 0.977 0.976 democrat
0.952 0.030 0.952 0.952 0.952 0.923 0.977 0.954 republican
Weighted Avg. 0.963 0.041 0.963 0.963 0.963 0.923 0.977 0.968
=== Confusion Matrix ===
a b <-- classified as
65 2 | a = democrat
2 40 | b = republican
Status
OK
Log
x 0
4:54 PM
25-Aug-18

Here is the decision tree visualization.



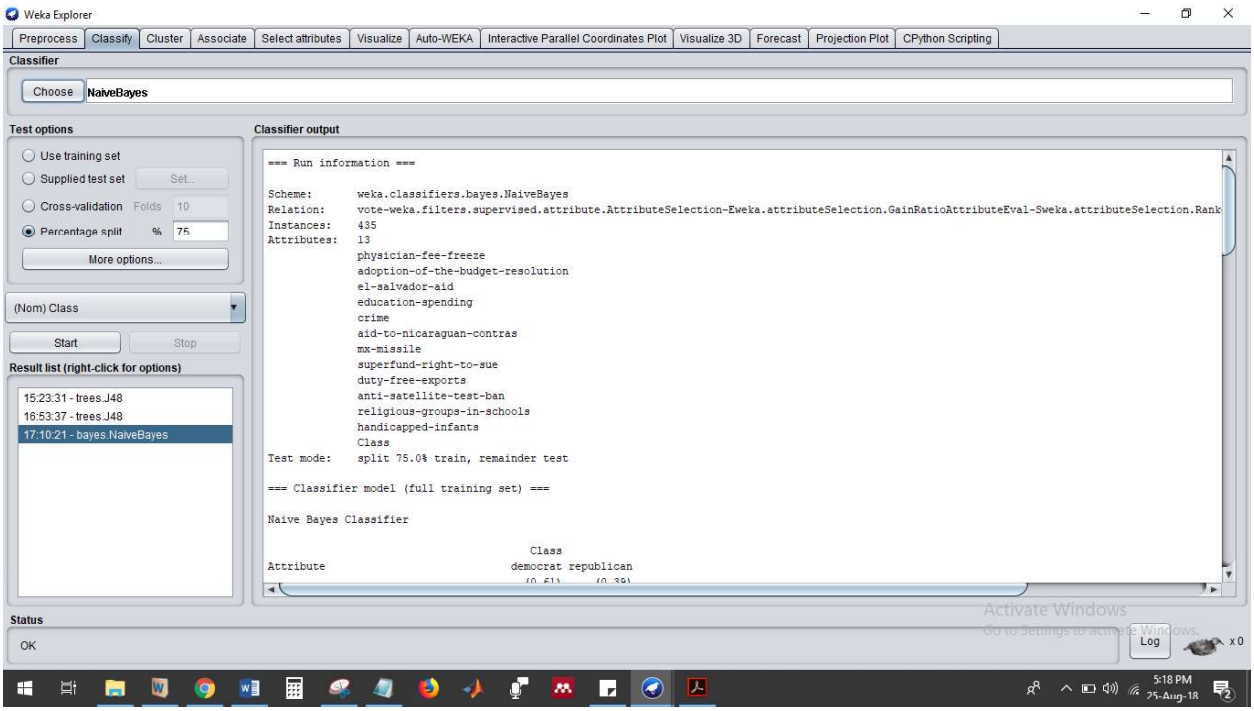
2.3 NaiveBayes Classifier:

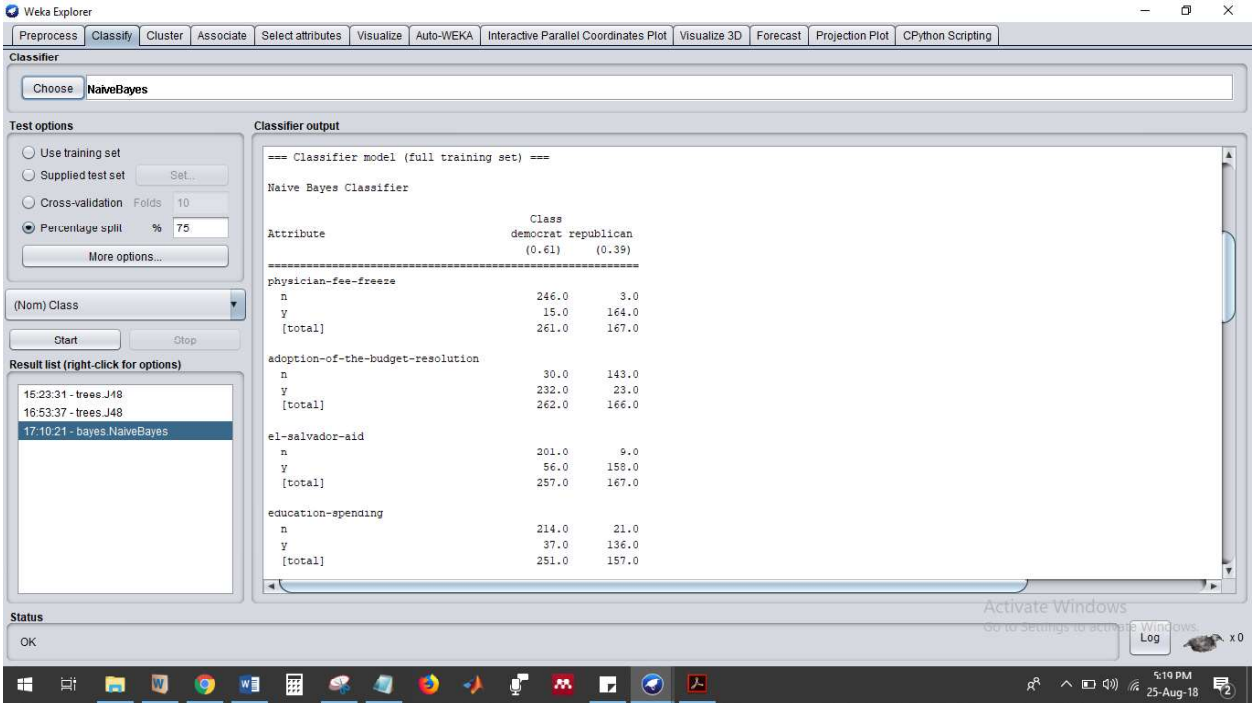
As classifier, "***NaiveBayes***" has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Different parameters setting have been given below.

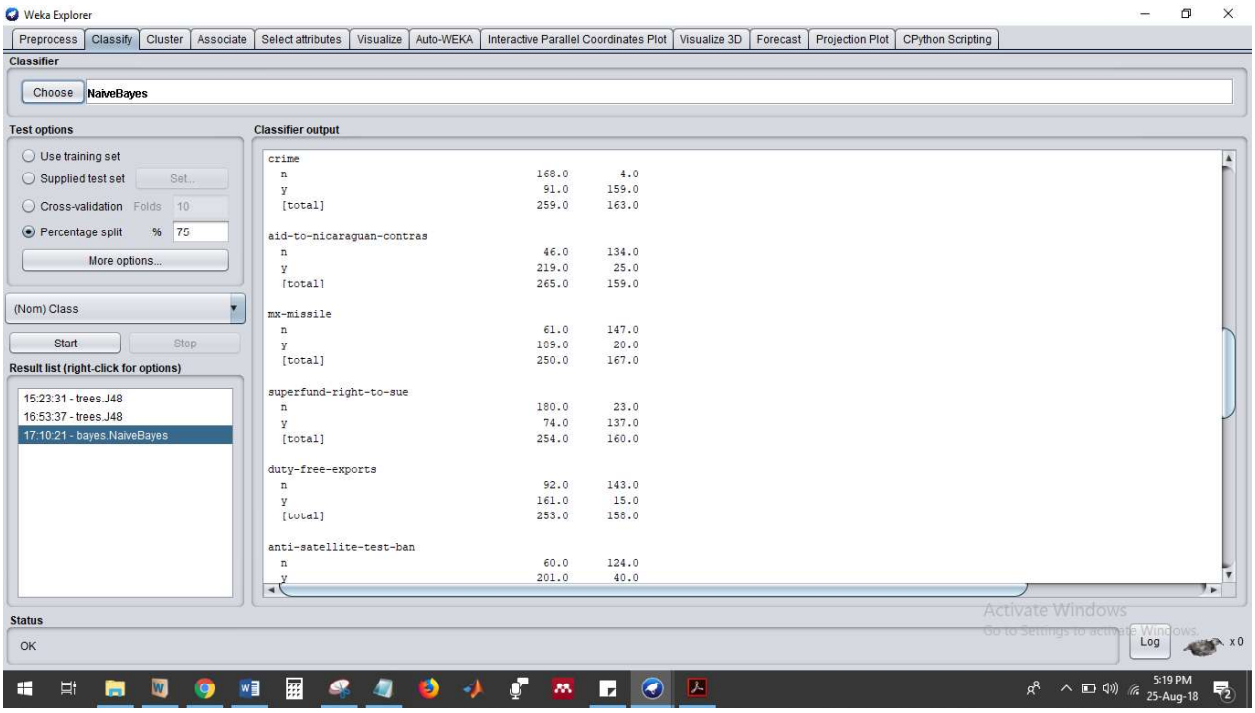


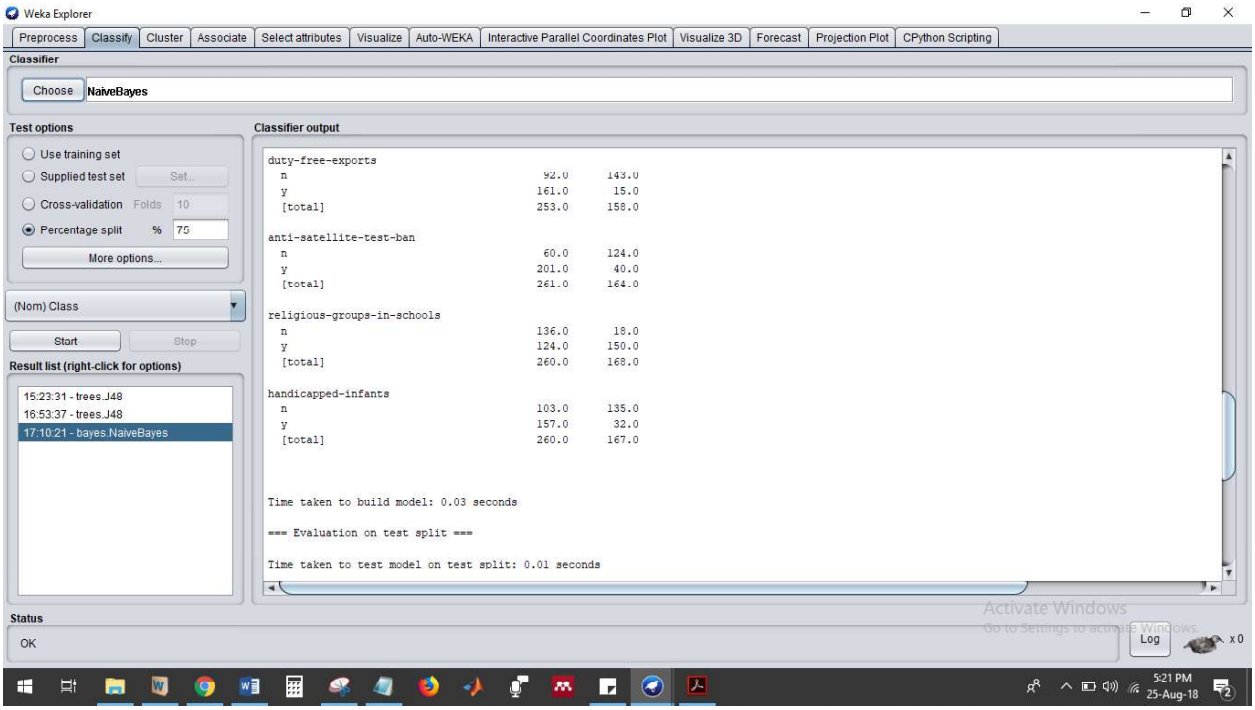
Result:

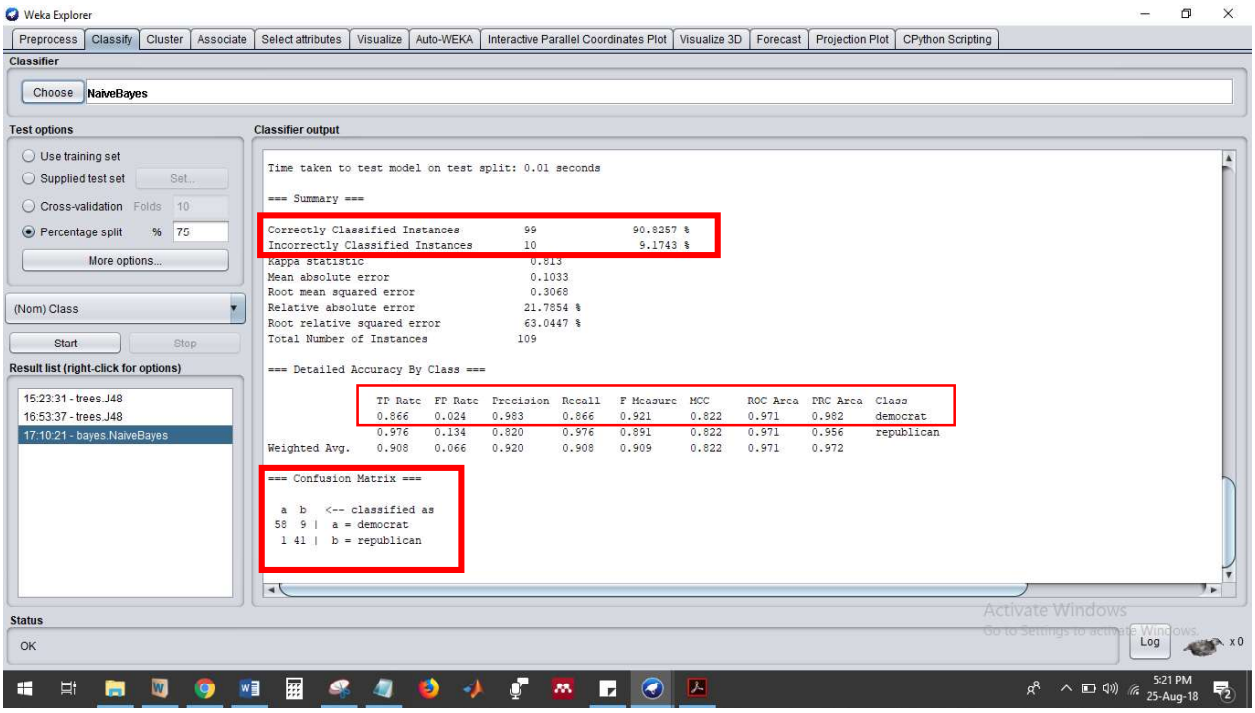
For NaiveBayes classifier, the overall classification accuracy is **90.8257%.**





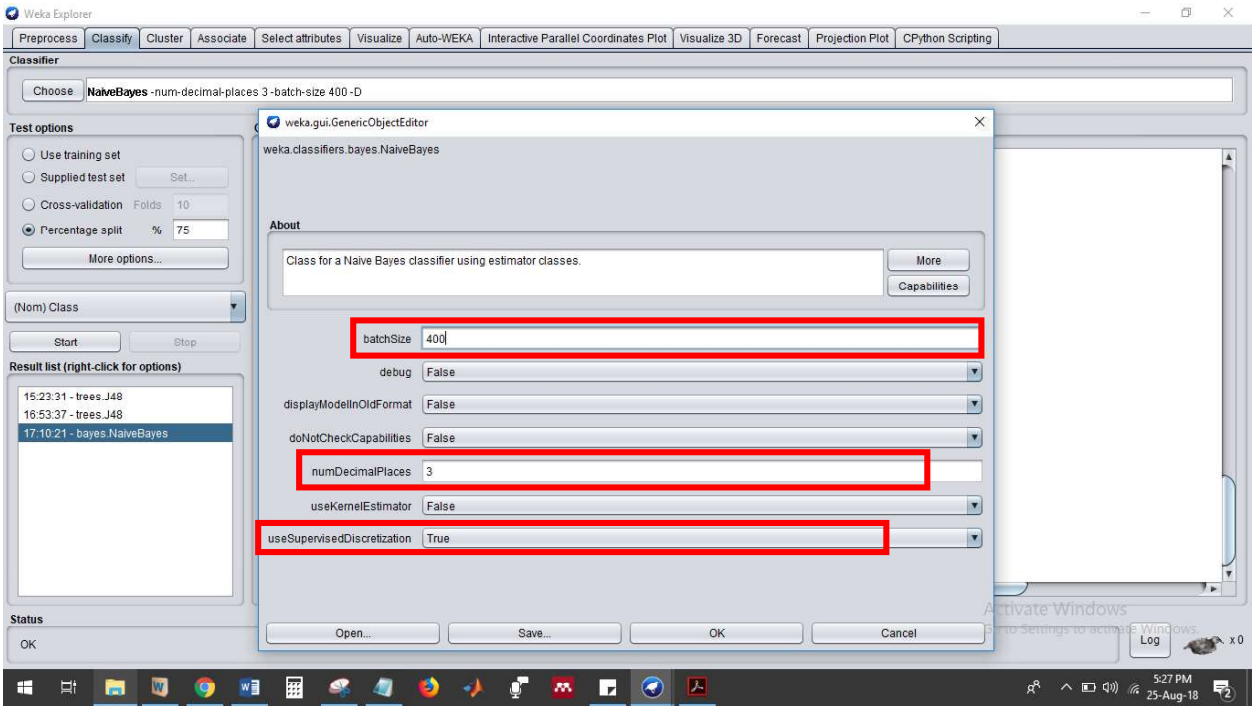

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Auto-WEKA
Interactive Parallel Coordinates Plot
Visualize 3D
Forecast
Projection Plot
CPython Scripting
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
75
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
15:23:31 - trees.J48
16:53:37 - trees.J48
17:10:21 - bayes.NaiveBayes
Classifier output
crime
n 168.0 4.0
y 91.0 159.0
[total] 259.0 163.0
aid-to-nicaraguan-contras
n 46.0 134.0
y 219.0 25.0
[total] 265.0 159.0
mx-missile
n 61.0 147.0
y 189.0 20.0
[total] 250.0 167.0
superfund-right-to-sue
n 180.0 23.0
y 74.0 137.0
[total] 254.0 160.0
duty-free-exports
n 92.0 143.0
y 161.0 15.0
[total] 253.0 158.0
anti-satellite-test-ban
n 60.0 124.0
y 201.0 40.0
Status
OK
Log
x 0
Activate Windows
Go to Settings to activate Windows.
5:19 PM
25-Aug-18


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Auto-WEKA
Interactive Parallel Coordinates Plot
Visualize 3D
Forecast
Projection Plot
CPython Scripting
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
75
More options...
(Nom) Class
Start
Stop
Result list (right-click for options)
15:23:31 - trees.J48
16:53:37 - trees.J48
17:10:21 - bayes.NaiveBayes
Classifier output
duty-free-exports
n 92.0 143.0
y 161.0 15.0
[total] 253.0 158.0
anti-satellite-test-ban
n 60.0 124.0
y 201.0 40.0
[total] 261.0 164.0
religious-groups-in-schools
n 136.0 18.0
y 124.0 150.0
[total] 260.0 168.0
handicapped-infants
n 103.0 135.0
y 157.0 32.0
[total] 260.0 167.0
Time taken to build model: 0.03 seconds
=== Evaluation on test split ===
Time taken to test model on test split: 0.01 seconds
Status
OK
Log
x 0
Activate Windows
Go to Settings to activate Windows.
5:21 PM
25-Aug-18



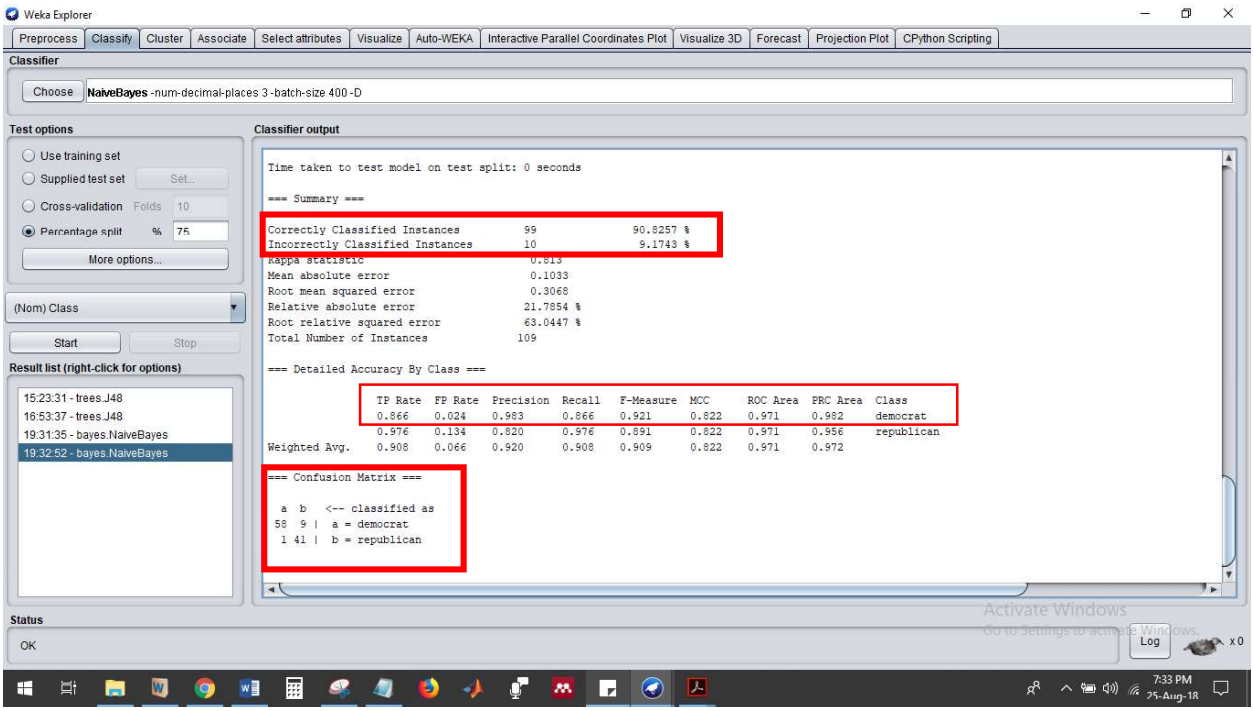
2.4 NaiveBayes Classifier with edited parameter:

As classifier, "***NaiveBayes***" has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Here, "***batchSizeDifferent***", "***numDecimalPlaces***" & "***useSuperviseDiscretization***" parameters have been set to 400, 3 & True respectively. Other parameter settings have been given below.



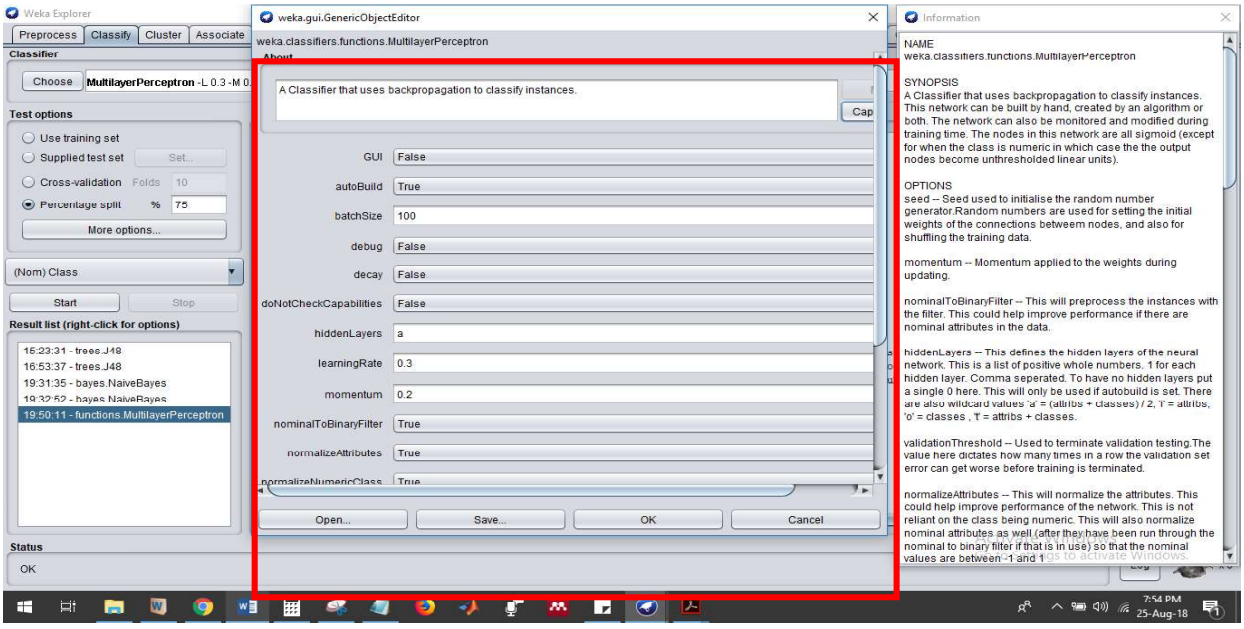
Result:

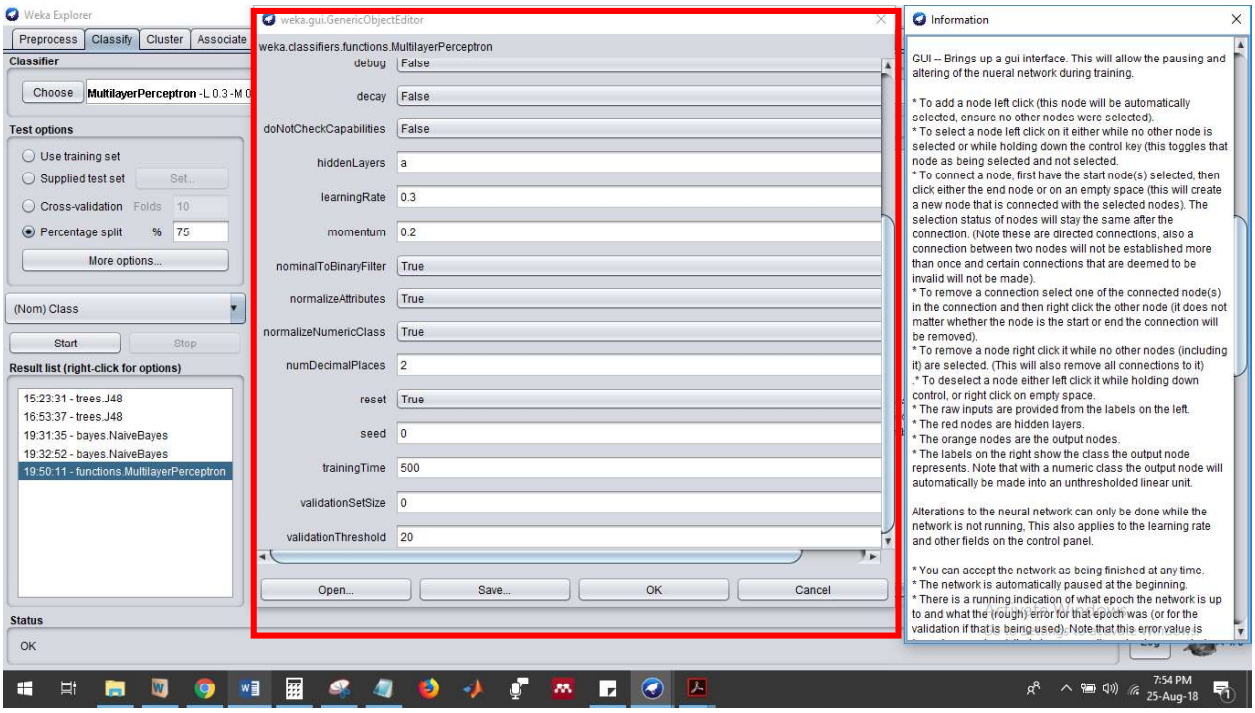
For NaiveBayes classifier with edited parameter, the overall classification accuracy is **90.8257%.**



2.5 MultilayerPerceptron function Classifier:

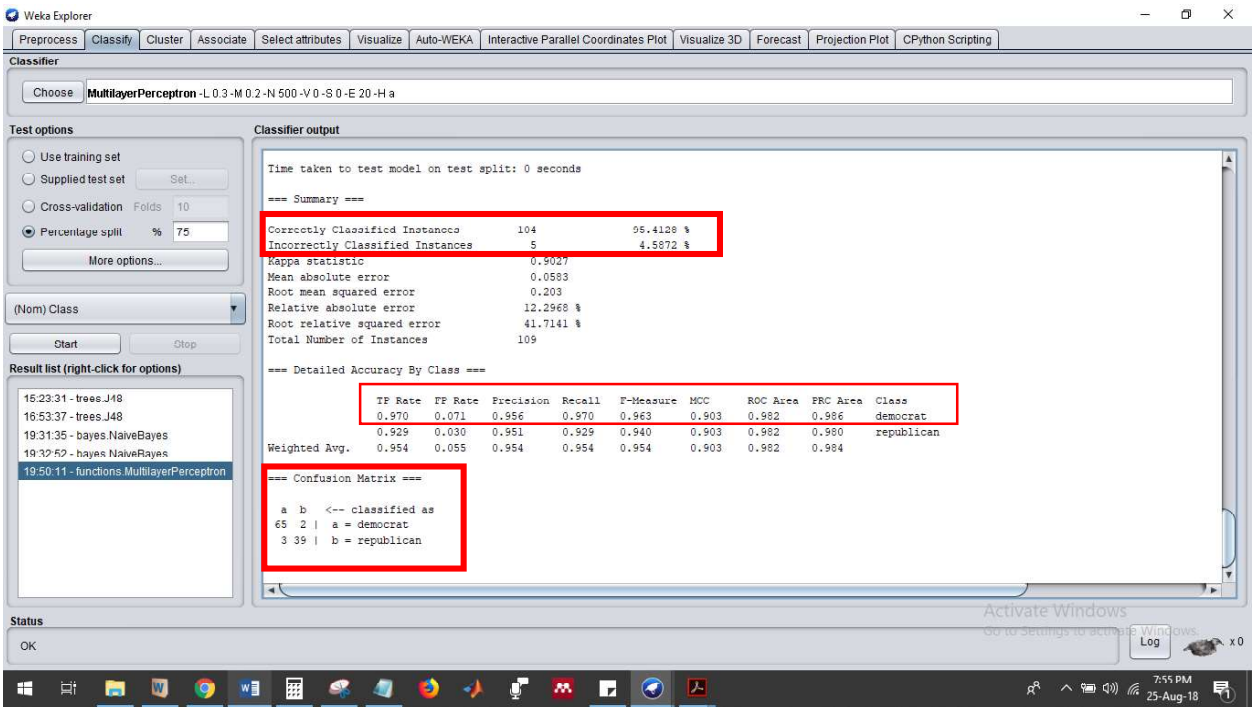
As classifier, "***MultilayerPerceptron***" has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Other parameter settings have been given below.





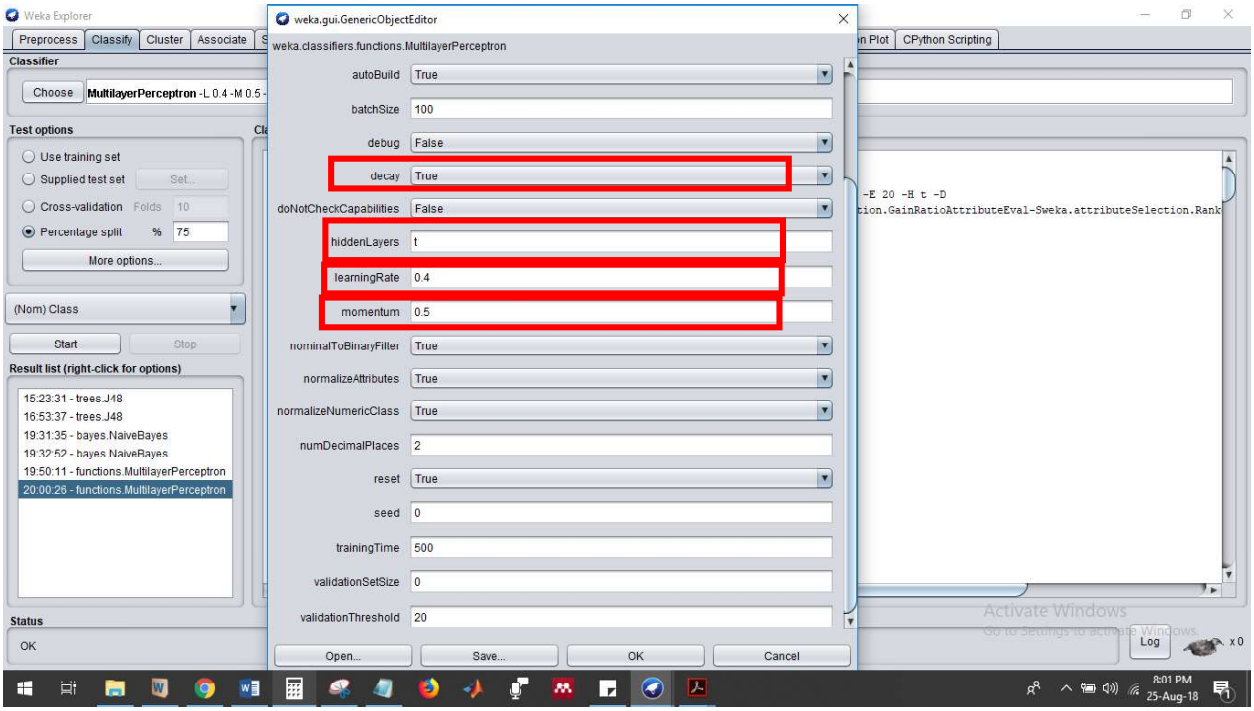
Result:

For MultilayerPerceptron classifier, the overall classification accuracy is **95.4128%.**



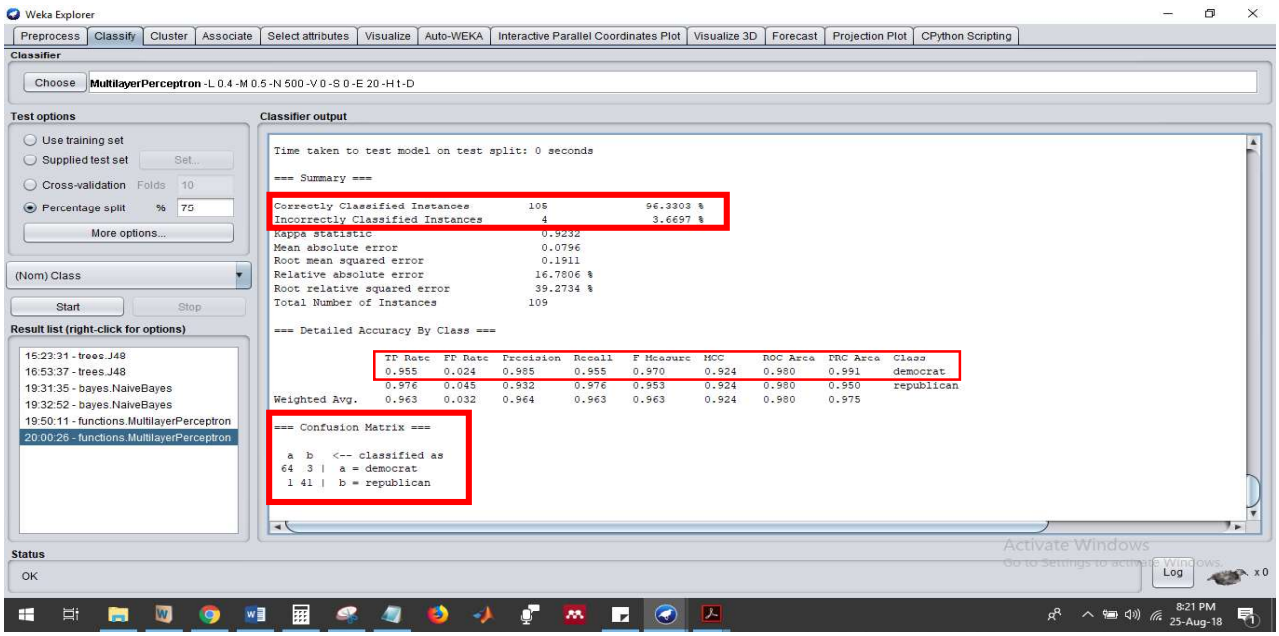
2.6 MultilayerPerceptron function Classifier with edited parameter:

As classifier, "***MultilayerPerceptron***" classifier with edited parameter has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Here, "***decay***", "***hiddenLayers***", "***learningRate***" & "***momentum***" parameters have been set to True, t, 0.4 & 0.5 respectively. Other parameter settings have been given below.



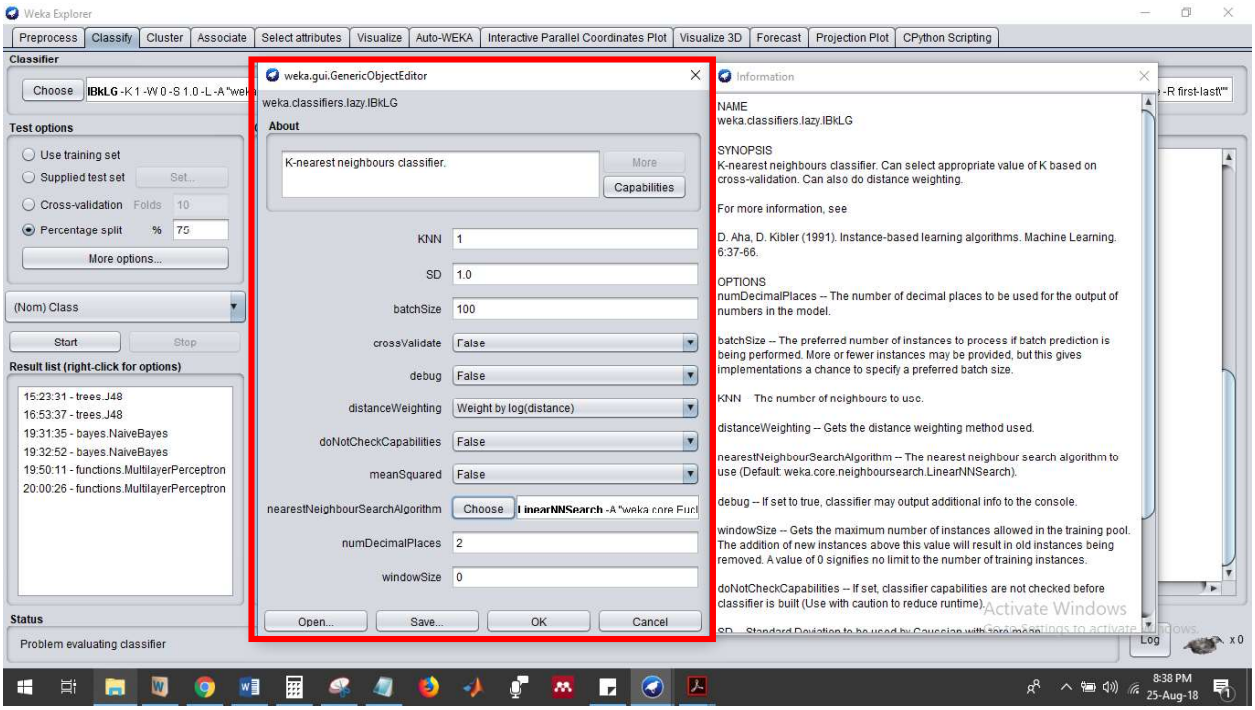
Result:

For MultilayerPerceptron classifier, the overall classification accuracy is **96.3303%.**



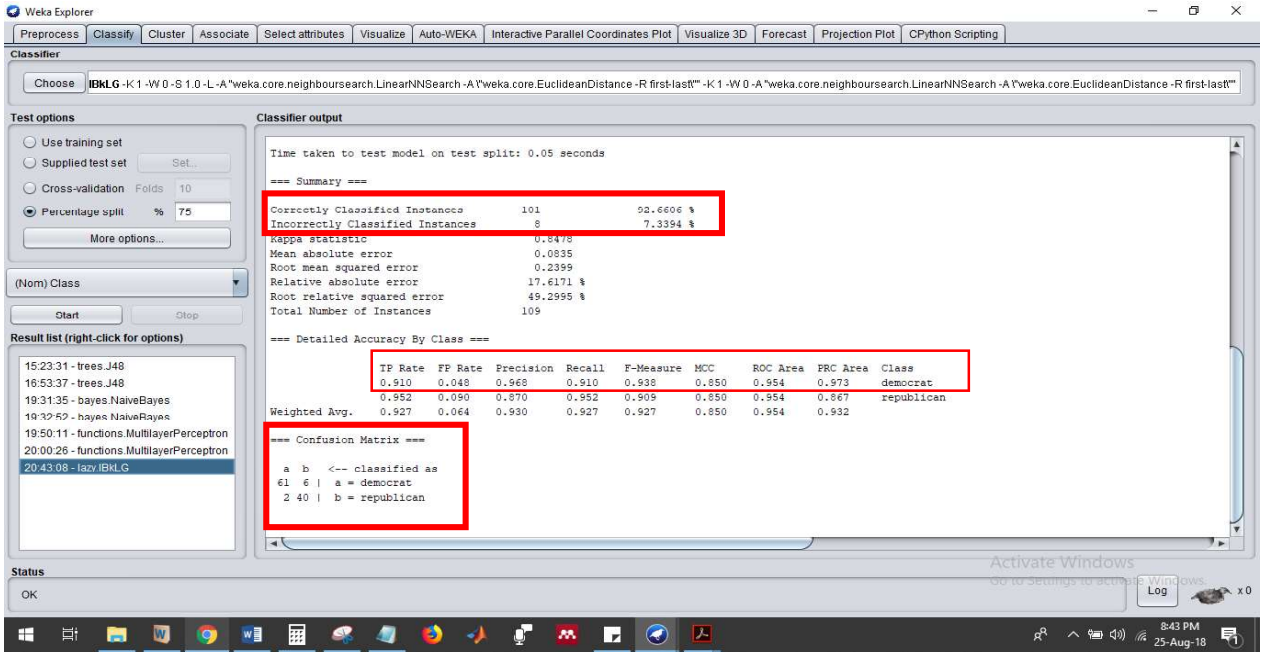
2.7 K-nearest neighbours classifier:

As classifier, "***K-nearest neighbours classifier***" has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Other parameter settings have been given below.



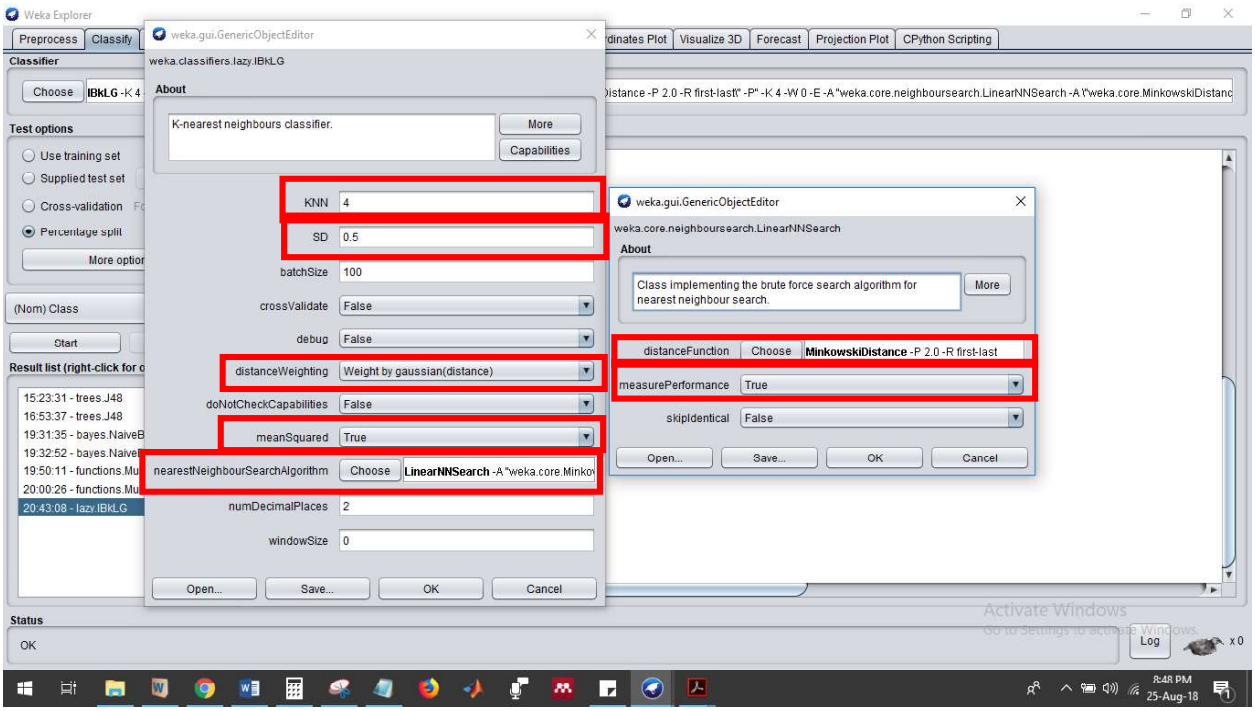
Result:

For K-nearest neighbours classifier, the overall classification accuracy is 92.6606%.



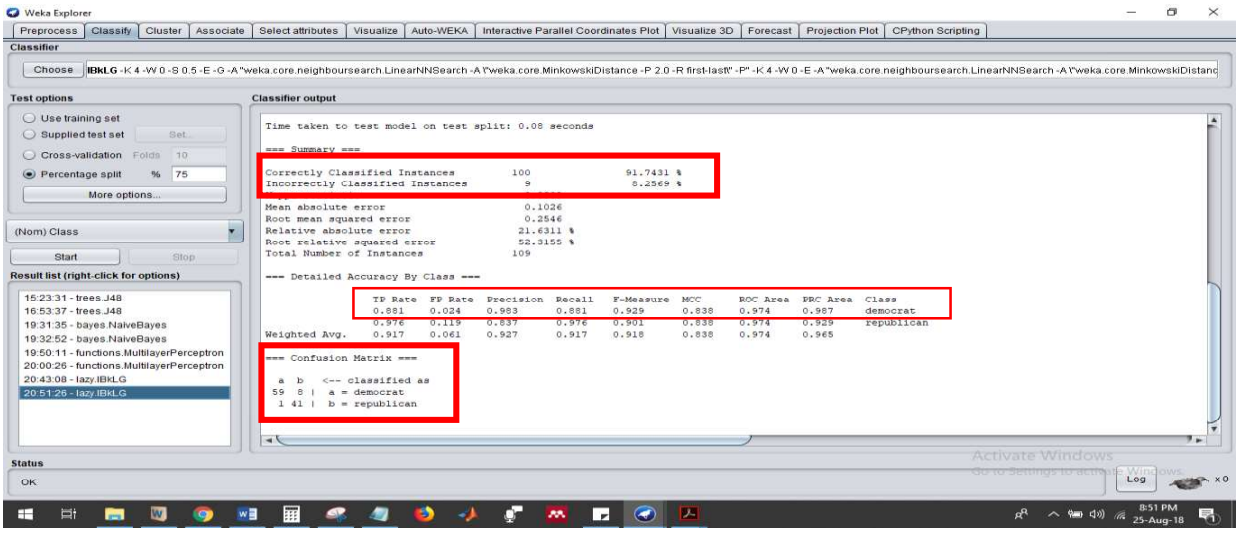
2.8 K-nearest neighbours classifier with edited parameter:

As classifier, "***K-nearest neighbours classifier***" with edited parameter has been chosen. 75% of total data set have been chosen for training purpose and rest 25% have been chosen for testing. Here, the value of parameter "***KNN***", "***SD***", "***distanceWeighting***" & "***meanSquared***" have been set to 4, 0.5, weight by guassian(distance) & True respectively. Also, in "***nearestNeighbourSearchAlgorithm***", "***MinkowskiDistance***" has been chosen as "***distanceFunction***" as well as "***measurePerformance***" has been chosen as True. Other parameter settings have been given below.



Result:

For K-nearest neighbours classifier with edited parameter, the overall classification accuracy is **91.7431%.**



3) Step-3:

For every model, below table contain accuracy, precision, recall, sensitivity and specificity for the class "***Democrat***".

		Class: Democrat				
S.I	Model name	Accuracy	Precision	Recall	Sensitivity	Specificity
2.1	Decision tree: C4.5_Pruned tree	94	98.4	94	94.02	97.61
2.2	Decision tree: C4.5_Unpruned tree	97	97	97	97.01	95.23
2.3	NaiveBayes Classifier	86.6	98.3	86.6	86.56	97.61
2.4	NaiveBayes Classifier with edited parameter	86.6	98.3	86.6	86.56	97.61
2.5	MultilayerPerceptron function Classifier	97	95.6	97	97.01	92.85
2.6	MultilayerPerceptron function Classifier with edited parameter	95.5	98.5	95.5	95.5	97.6
2.7	K-nearest neighbours classifier	91	96.8	91	91.04	95.23
2.8	K-nearest neighbours classifier with edited parameter	88.1	98.3	88.1	88.05	97.61

4) Step-4:

Here, we will choose the best, the second best & the third best model to predict future voters from step-2.

4.1 The best model (Decision tree: C4.5_Unpruned tree):

Decision tree algorithm C4.5 with unpruned data is the best model. Because—

a. Compared to other models, this model has highest accuracy rate of 96.33%.
b. This algorithm have taken only 0.02 second to build the model. Therefore, it is faster than other models.
c. Relative absolute error & Root relative squared error for this model are only 15.8927% & 38.3502% respectively, which are lowest among all the models.

4.2 The second best model (MultilayerPerceptron function Classifier with edited parameter):

Multilayer Perceptron function Classifier with edited parameter is the second best model. Because—

a. This model has accuracy rate of 96.33%, which is equal to the best model.
b. This algorithm have taken only 0.97 second to build the model. Therefore, it is very fast than other models but slower than the best.
c. Relative absolute error & Root relative squared error for this model are only 16.7806% & 39.2734% respectively, which are greater than the best but lowest among all other models.

4.3 The third best model (Decision tree: C4.5_Pruned tree):

Decision tree algorithm C4.5 with pruned data is the third best model. Because—

a. This model has the third highest accuracy rate of 95.4128%.
b. This algorithm have taken only 0.06 second to build the model. Therefore, it is faster than other models but slower than the best.

c. Relative absolute error & Root relative squared error for this model are only 19.9567% & 44.3433% respectively, which are greater than the best & second best but lowest among all remaining models.

5) Step-5:

Three characteristics of 'democrat' voter based on the decision tree built and displayed in 2.1 & 2.2 are follows-

a. Democrats do not want physician-fee-freeze.
b. Democrats do not want anti-satellite-test-ban.
c. Democrats want superfund-right-to-sue.

Production rule:

a. If someone does not support "***physician-fee-freeze"***, then he/she is a "democrat".
b. If someone supports "***adoption-of -the-budget-resolution***" and does not support "***anti-satellite-test-ban***", then he/she is a "democrat".
c. If someone supports "***adoption-of -the-budget-resolution***" and does not support "***anti-satellite-test-ban***" and supports "***superfund-right-to-sue***", then he/she is a "democrat".

Conclusion:

We have successfully implemented classification algorithms based on 1984 United States Congressional Voting Records to predict future voters' class which could be "Democrat" or "Republican". Based on accuracy, runtime and other key features, we have also ranked the best, the second best & the third best classification algorithm. These high accurate classifier algorithms will be very helpful to classify future voters to predict their preferred team to vote.