EX.No: 17
Date :

EVALUATING ACCURACY OF THE CLASSIFIERS

DESCRIPTION :

Consider the german credit dataset which can be downloaded from the UCI repository.

ANALYSIS :

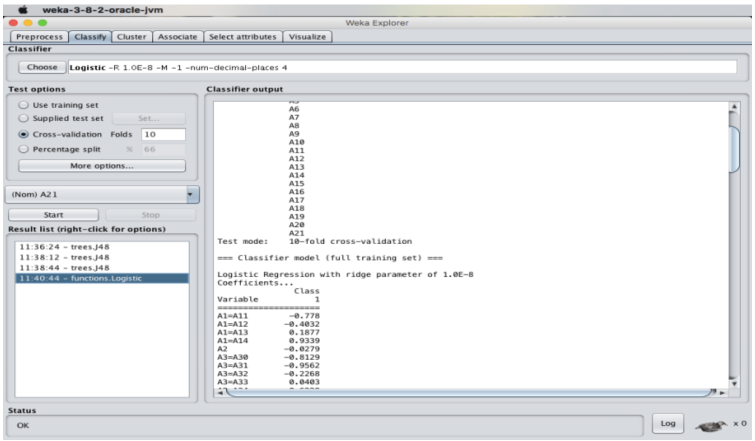
A) Logistic Regression :

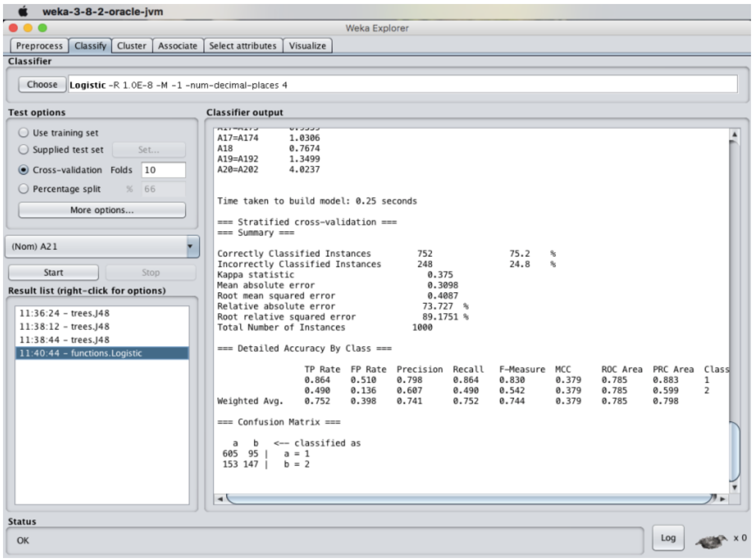
Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical).

Steps :

- Load the dataset into the weka tool and preprocess it.
- Apply the classification the logistic regression technique and execute for the result.

Output :





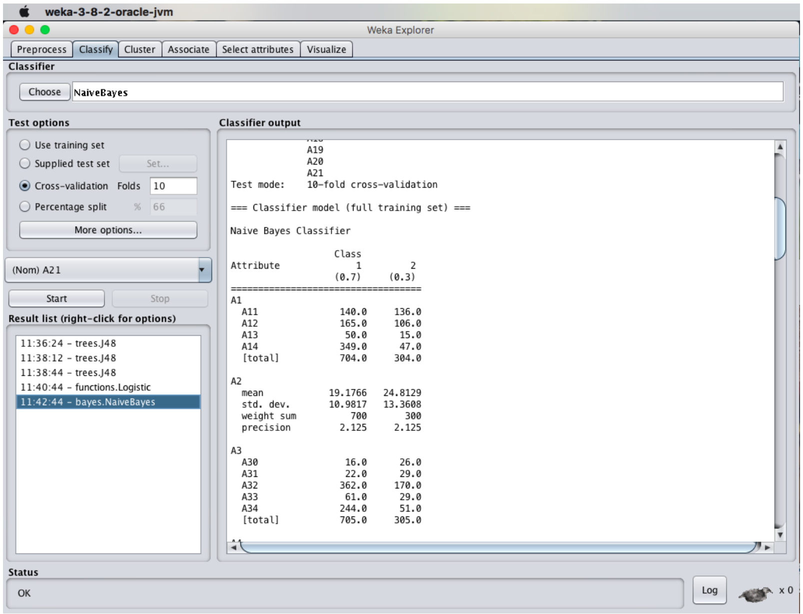
B) Naïve Bayes Algorithm :

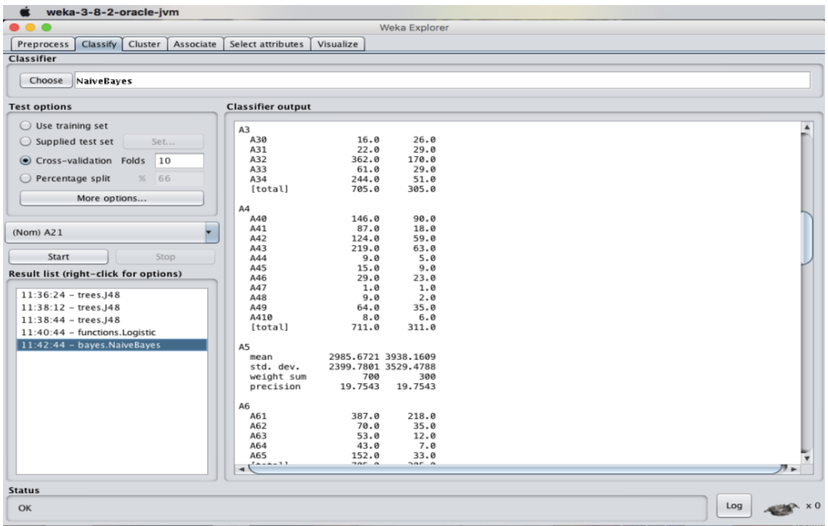
The Naive Bayesian classifier is based on Bayes' theorem with the independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

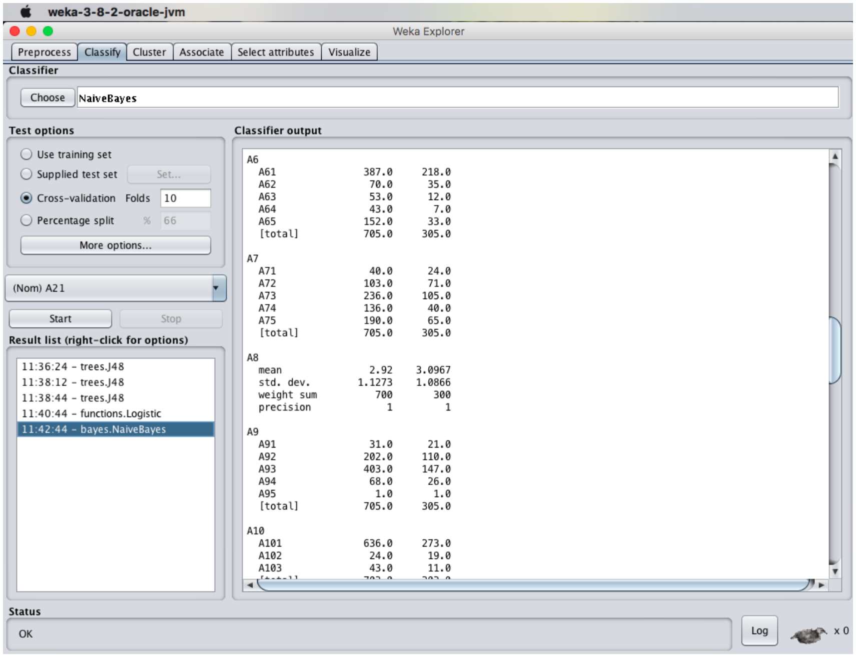
Steps :

- Load the dataset into the weka tool and preprocess it.
- Apply the classification the Naïve bayes technique and execute for the result.

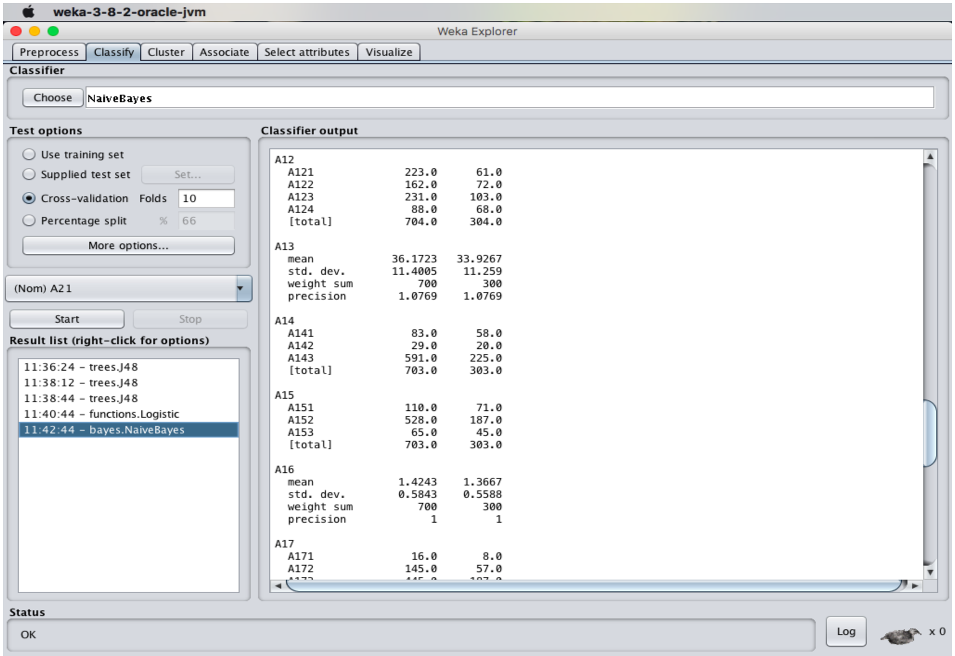
Output :



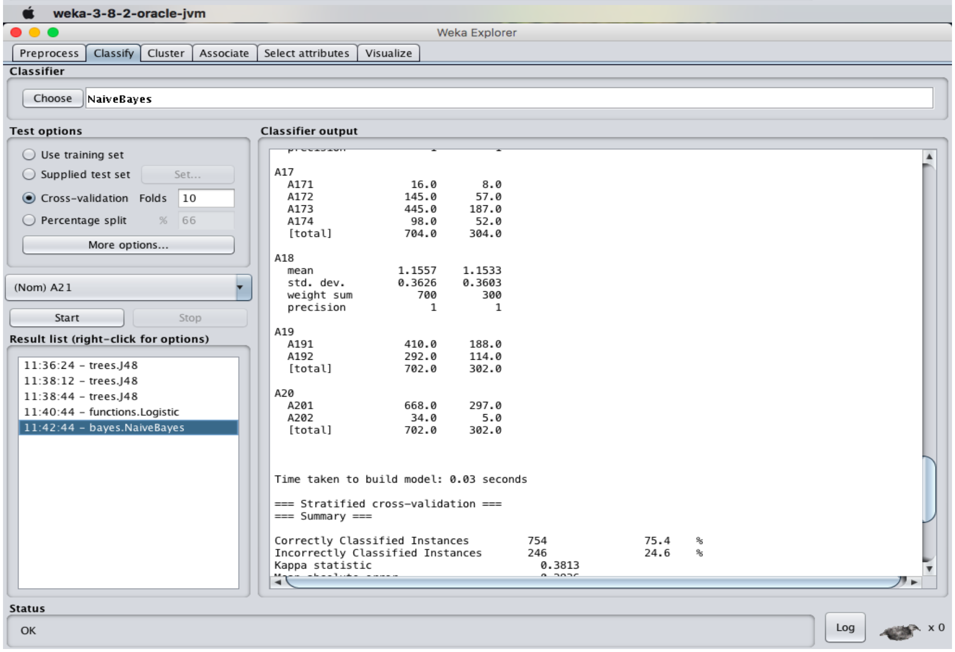




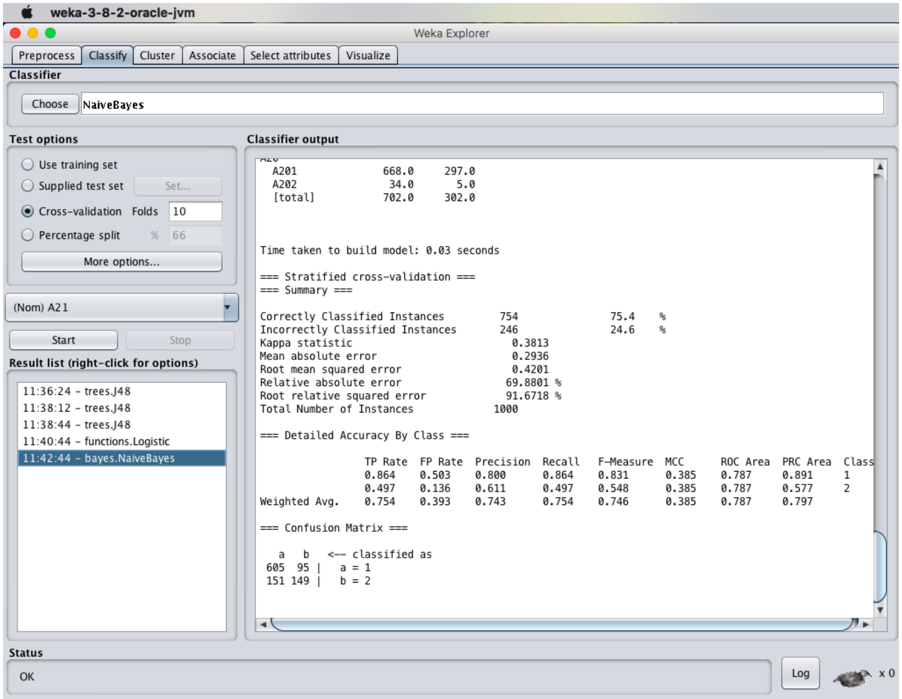
weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose NaiveBayes
Test options
Use training set
Supplied test set Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A6
A61 387.0 218.0
A62 70.0 35.0
A63 53.0 12.0
A64 43.0 7.0
A65 152.0 33.0
[total] 705.0 305.0
A7
A71 40.0 24.0
A72 103.0 71.0
A73 236.0 105.0
A74 136.0 40.0
A75 190.0 65.0
[total] 705.0 305.0
A8
mean 2.92 3.0967
std. dev. 1.1273 1.0866
weight sum 700 300
precision 1 1
A9
A91 31.0 21.0
A92 202.0 110.0
A93 403.0 147.0
A94 68.0 26.0
A95 1.0 1.0
[total] 705.0 305.0
A10
A101 636.0 273.0
A102 24.0 19.0
A103 43.0 11.0
Status
OK
Log
x 0



weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose NaiveBayes
Test options
Use training set
Supplied test set Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A12
A121 223.0 61.0
A122 162.0 72.0
A123 231.0 103.0
A124 88.0 68.0
[total] 704.0 304.0
A13
mean 36.1723 33.9267
std. dev. 11.4005 11.259
weight sum 700 300
precision 1.0769 1.0769
A14
A141 83.0 58.0
A142 29.0 20.0
A143 591.0 225.0
[total] 703.0 303.0
A15
A151 110.0 71.0
A152 528.0 187.0
A153 65.0 45.0
[total] 703.0 303.0
A16
mean 1.4243 1.3667
std. dev. 0.5843 0.5588
weight sum 700 300
precision 1 1
A17
A171 16.0 8.0
A172 145.0 57.0
Status
OK
Log
x 0



weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose NaiveBayes
Test options
Use training set
Supplied test set Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start Stop
Result list (right-click for options)
11:36:24 - trees.J48
11:38:12 - trees.J48
11:38:44 - trees.J48
11:40:44 - functions.Logistic
11:42:44 - bayes.NaiveBayes
Classifier output
A17
A171 16.0 8.0
A172 145.0 57.0
A173 445.0 187.0
A174 98.0 52.0
[total] 704.0 304.0
A18
mean 1.1557 1.1533
std. dev. 0.3626 0.3603
weight sum 700 300
precision 1 1
A19
A191 410.0 188.0
A192 292.0 114.0
[total] 702.0 302.0
A20
A201 668.0 297.0
A202 34.0 5.0
[total] 702.0 302.0
Time taken to build model: 0.03 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 754 75.4 %
Incorrectly Classified Instances 246 24.6 %
Kappa statistic 0.3813
Status
OK
Log
x 0



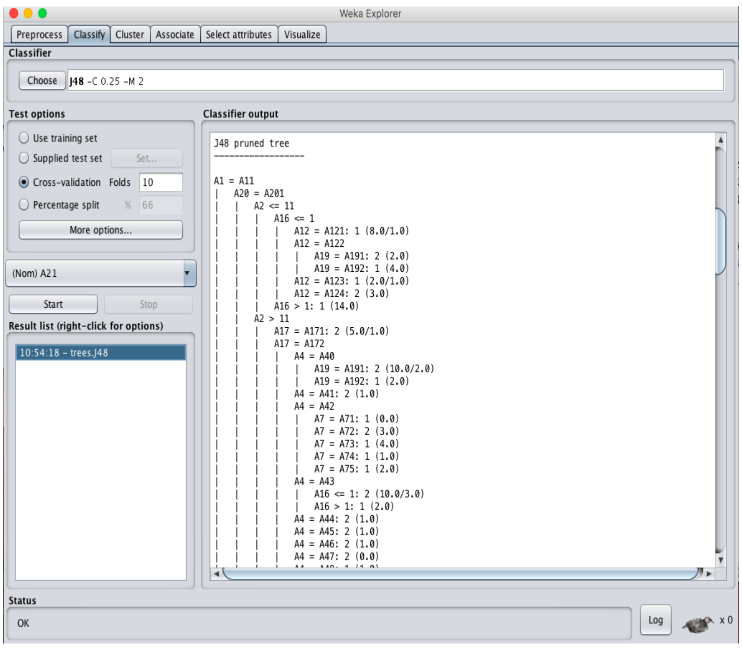
C) J48 Algorithm :

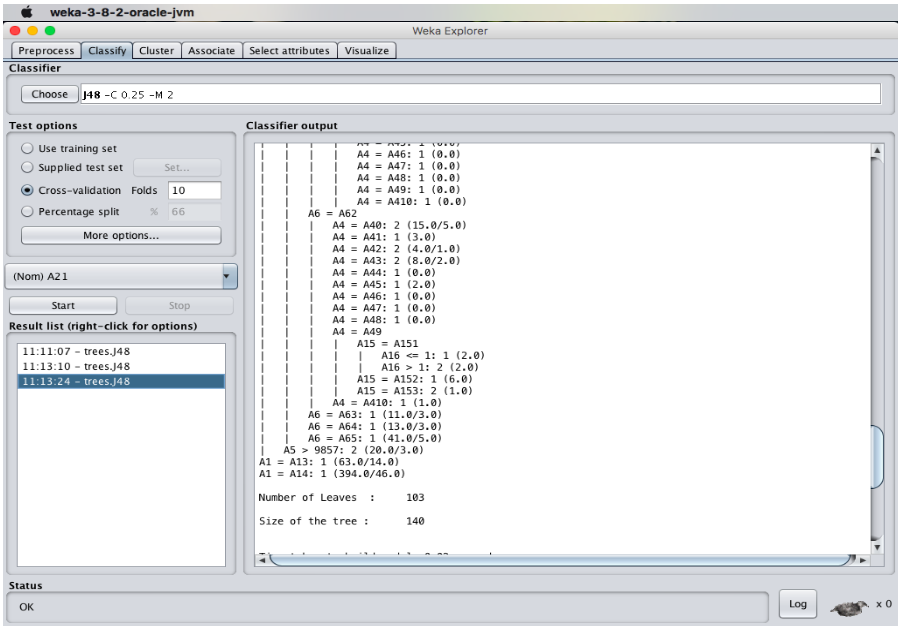
Classification is the process of building a model of classes from a set of records that contain class labels. Decision Tree Algorithm is to find out the way the attributes-vector behaves for a number of instances. Also on the bases of the training instances the classes for the newly generated instances are being found. This algorithm generates the rules for the prediction of the target variable. With the help of tree classification algorithm the critical distribution of the data is easily understandable.

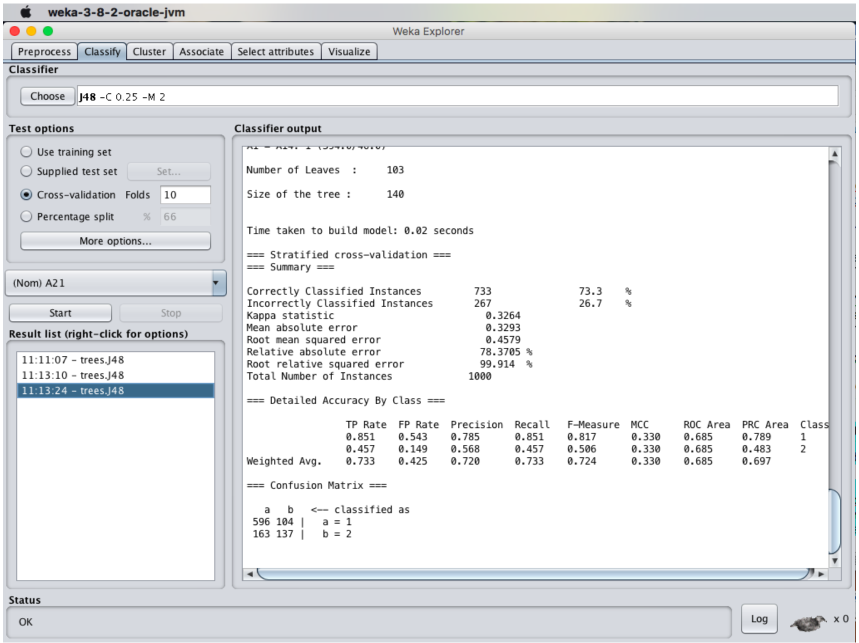
Steps :

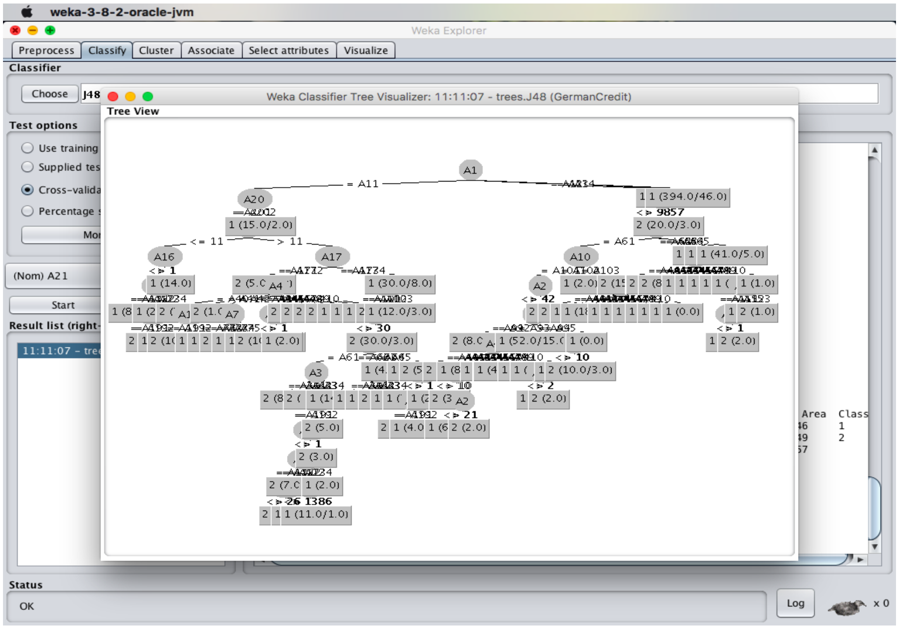
- Load the dataset into the weka tool and preprocess it.
- Apply the classification the J48 technique and execute for the result.

Output :




weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:13:24 - trees.J48
Classifier output
| | | | A4 = A46: 1 (0.0)
| | | | A4 = A47: 1 (0.0)
| | | | A4 = A48: 1 (0.0)
| | | | A4 = A49: 1 (0.0)
| | | | A4 = A410: 1 (0.0)
| | A6 = A62
| | | A4 = A40: 2 (15.0/5.0)
| | | A4 = A41: 1 (3.0)
| | | A4 = A42: 2 (4.0/1.0)
| | | A4 = A43: 2 (8.0/2.0)
| | | A4 = A44: 1 (0.0)
| | | A4 = A45: 1 (2.0)
| | | A4 = A46: 1 (0.0)
| | | A4 = A47: 1 (0.0)
| | | A4 = A48: 1 (0.0)
| | | A4 = A49
| | | | A15 = A151
| | | | | A16 <= 1: 1 (2.0)
| | | | | A16 > 1: 2 (2.0)
| | | | A15 = A152: 1 (6.0)
| | | | A15 = A153: 2 (1.0)
| | | A4 = A410: 1 (1.0)
| | A6 = A63: 1 (11.0/3.0)
| | A6 = A64: 1 (13.0/3.0)
| | A6 = A65: 1 (41.0/5.0)
| A5 > 9857: 2 (20.0/3.0)
A1 = A13: 1 (63.0/14.0)
A1 = A14: 1 (394.0/46.0)
Number of Leaves : 103
Size of the tree : 140
Status
OK
Log
x 0


weka-3-8-2-oracle-jvm
Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
J48 -C 0.25 -M 2
Test options
Use training set
Supplied test set
Set...
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) A21
Start
Stop
Result list (right-click for options)
11:11:07 - trees.J48
11:13:10 - trees.J48
11:13:24 - trees.J48
Classifier output
Number of Leaves : 103
Size of the tree : 140
Time taken to build model: 0.02 seconds
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances 733 73.3 %
Incorrectly Classified Instances 267 26.7 %
Kappa statistic 0.3264
Mean absolute error 0.3293
Root mean squared error 0.4579
Relative absolute error 78.3705 %
Root relative squared error 99.914 %
Total Number of Instances 1000
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
0.851 0.543 0.785 0.851 0.817 0.330 0.685 0.789 1
0.457 0.149 0.568 0.457 0.506 0.330 0.685 0.483 2
Weighted Avg. 0.733 0.425 0.720 0.733 0.724 0.330 0.685 0.697
=== Confusion Matrix ===
a b <-- classified as
596 104 | a = 1
163 137 | b = 2
Status
OK
Log
x 0



D) K-Nearest Neighbor :

K-Nearest Neighbors is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

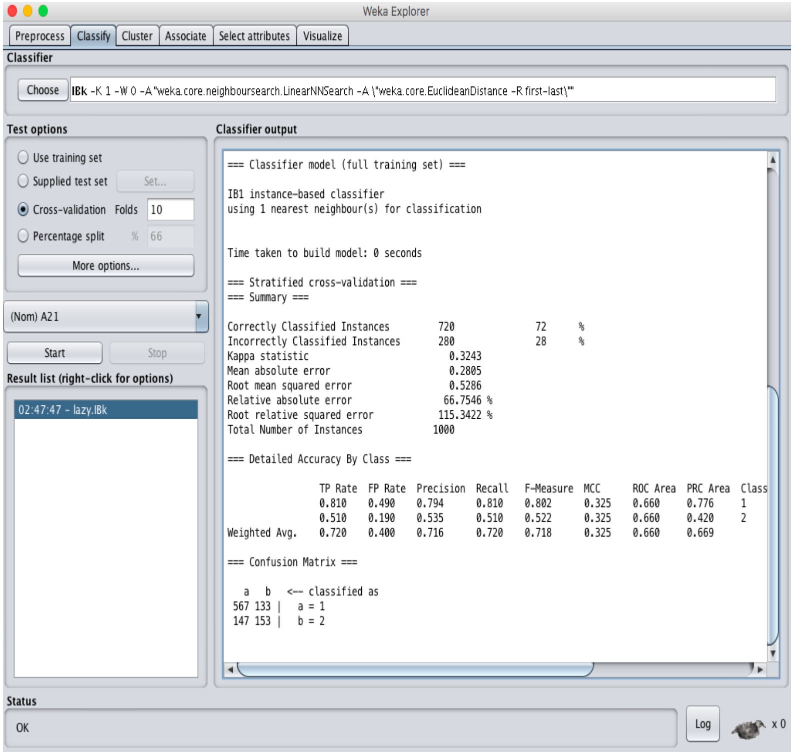
It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.

Steps :

- Load the dataset into the weka tool and preprocess it.
- Apply the classification the K- Nearest Neighbor technique and execute for the result.

Output :

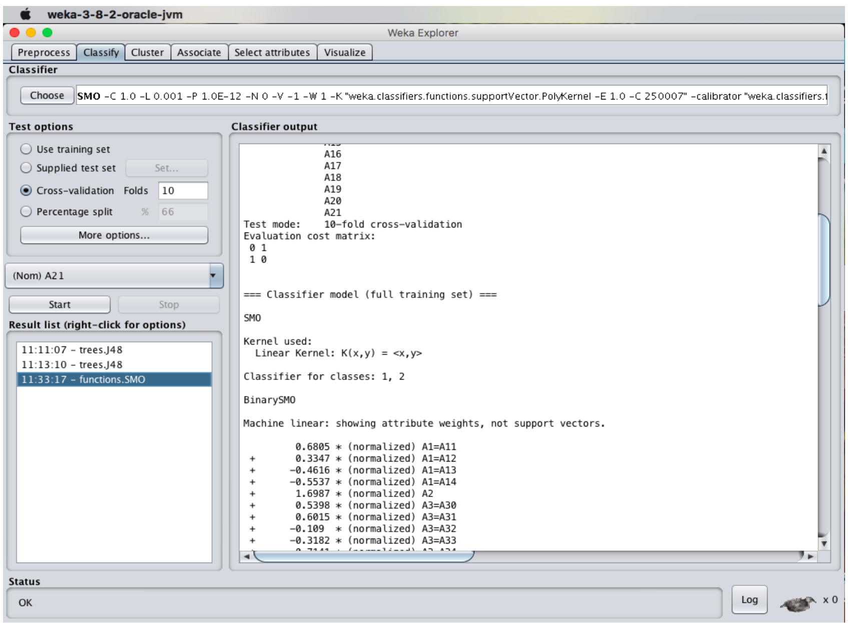


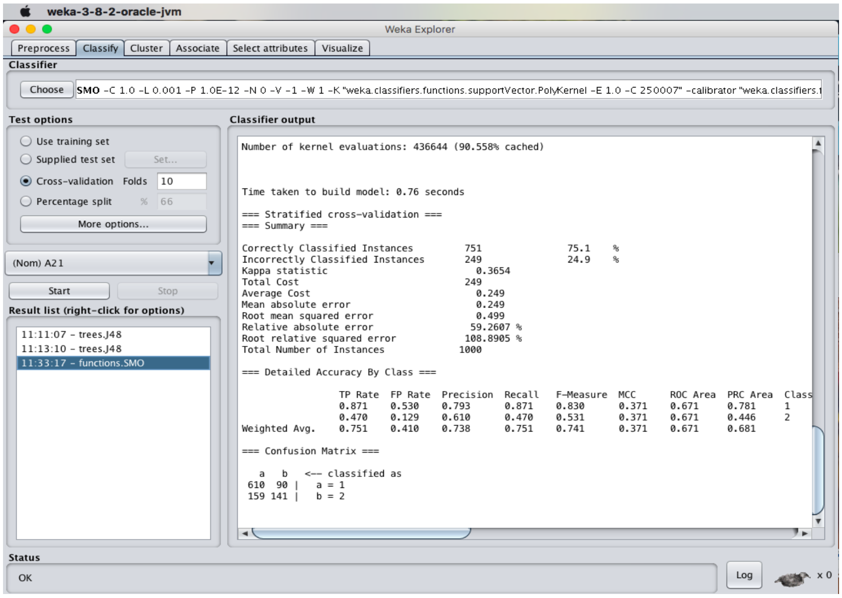
E) SMO Algorithm :

The iterative algorithm Sequential Minimal Optimization (SMO) is used for solving quadratic programming (QP) problems. One example where QP problems are relevant is during the training process of support vector machines (SVM). The SMO algorithm is used to solve in this example a constraint optimization problem. John Platt proposed this algorithm in 1998 and it was successfully used since then. We describe here the basics of the algorithm in the light of big data.

Steps :

- Load the dataset into the weka tool and preprocess it.
- Apply the classification the Sequential Minimal Optimization (SMO)technique and execute for the result.





RESULT :

Thus, the comparison of the confusion matrix for all the methods and techniques. Out of the comparing matrix with all the techniques there is a change in instances. Naïve bayes has more number of correct instances than other but when compared to time K-nearest neighbor is best. The above graphs will show the variations of values in the parameters.

EX.No: 18
Date :

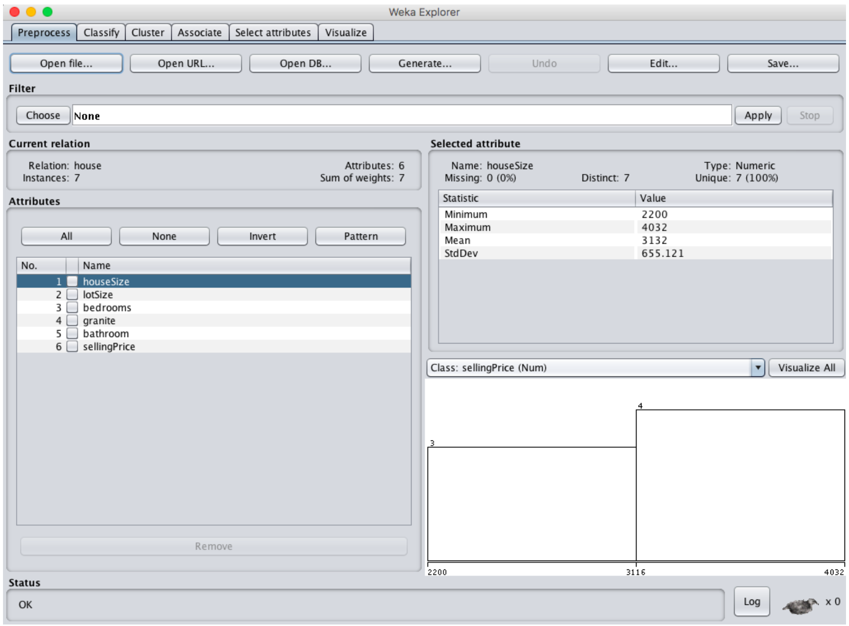
NUMERICAL PREDICTION ANALYSIS USING LINEAR REGRESSION THROUGH WEKA

DESCRIPTION :

Consider a dataset of house.arff where it contains the attributes as house size, lot size, bedrooms, granite, bathroom and the selling price.

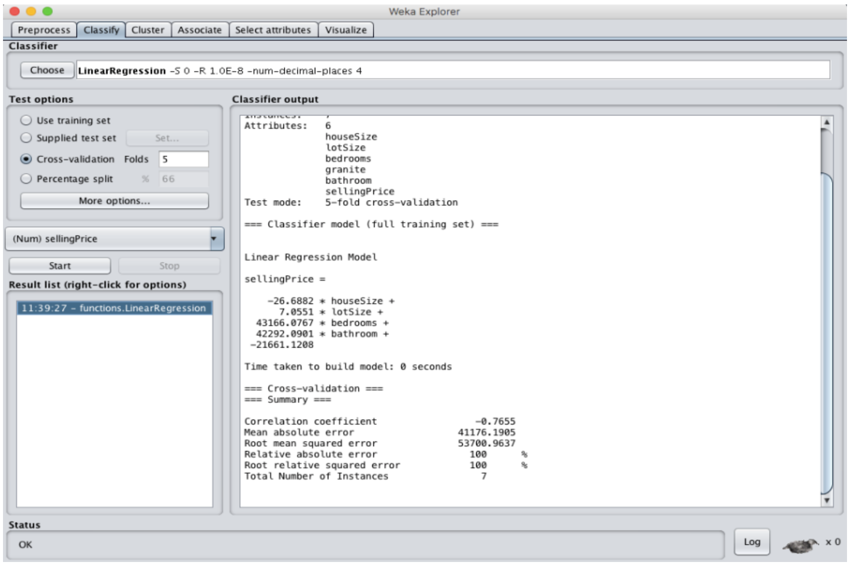
Steps :

- Load the dataset into the weka tool and check for the attributes.
- Classify the data using linear regression analysis method (or) technique.
- Check for the cross-validation folds where the value of the folds should be less than the value of the instances present in the dataset.
- Observe the cross validation summary after applying the linear regression technique for the price of the house.

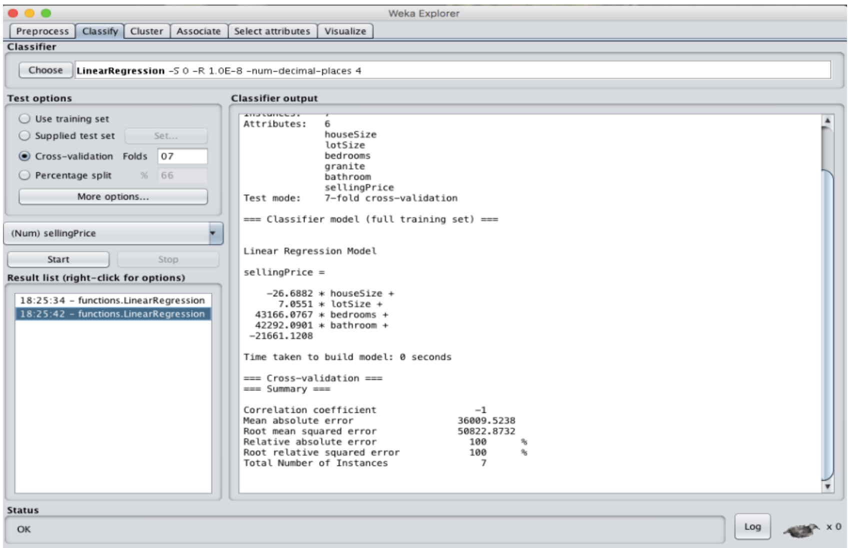


OBSERVATION :

❖ **When cross validation folds = 05 :**



- ❖ **When cross validation folds = 10 :**



RESULT :

Thus, the house selling price has been observed using linear regression model. If the value of cross validation folds decreases time for creating model will be less than when folds value high, and the mean absolute error and Root mean square error values decreases with increase in the cross validation folds value.