Data loading and analysis:-

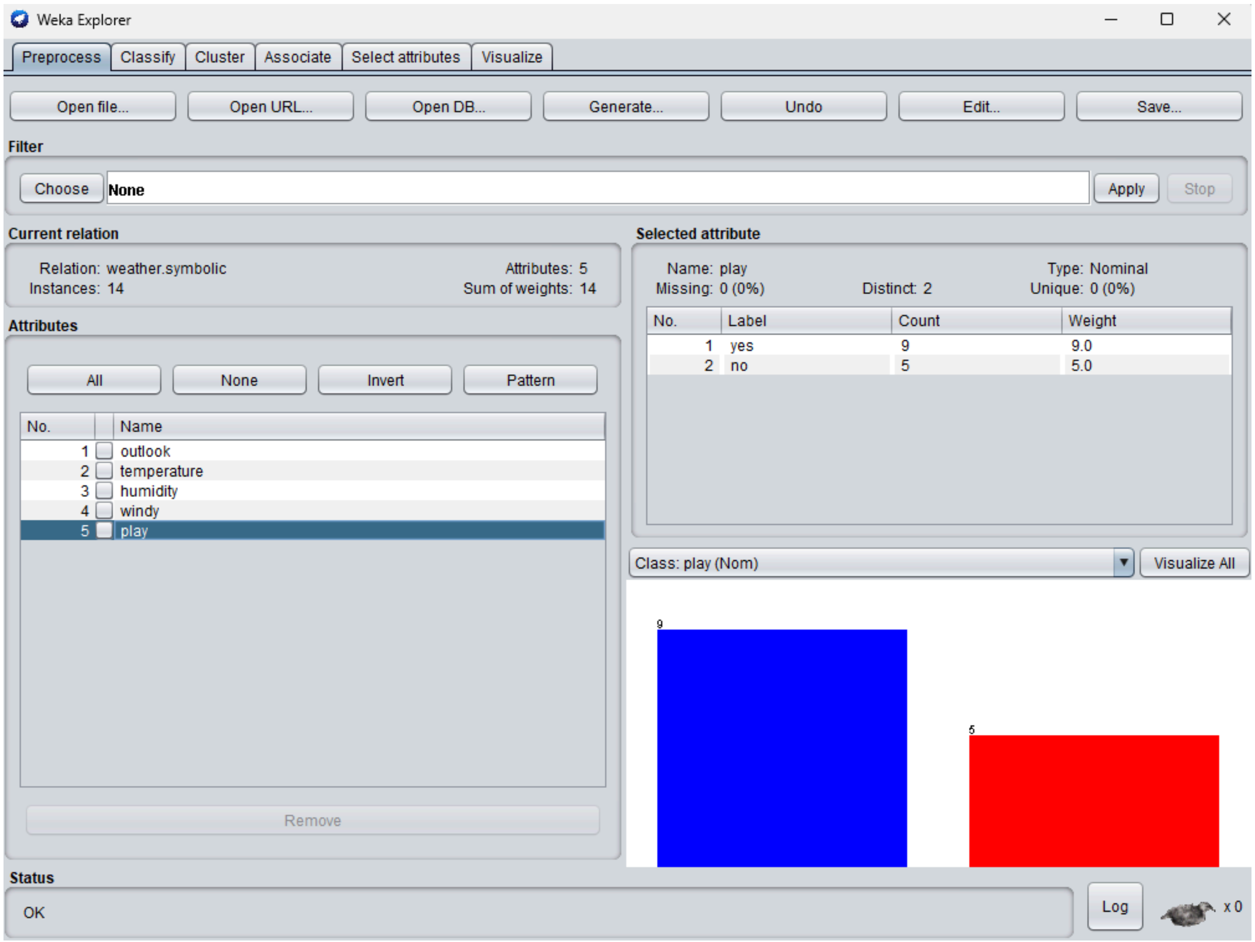
Data Set I used:

No.	1: outlook	2: temperature	3: humidity	4: windy	5: **play**
	Nominal	Nominal	Nominal	Nominal	Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

a) What is the size of the training set?
14

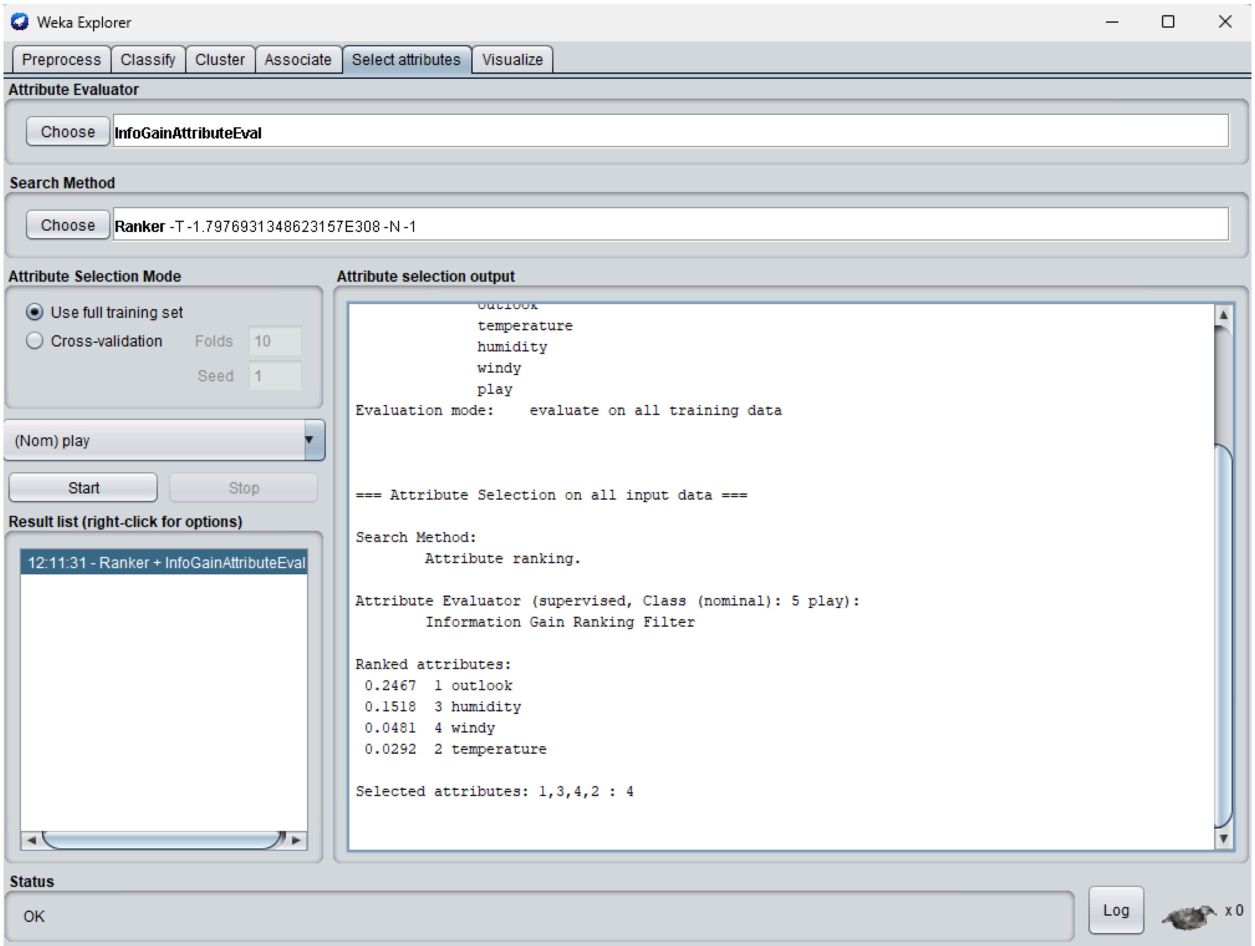
b) How many attributes exist in the training set?
5

c) How many instances are positive (Enjoy = yes) and how many negative?
Positive: 9
Negative: 5



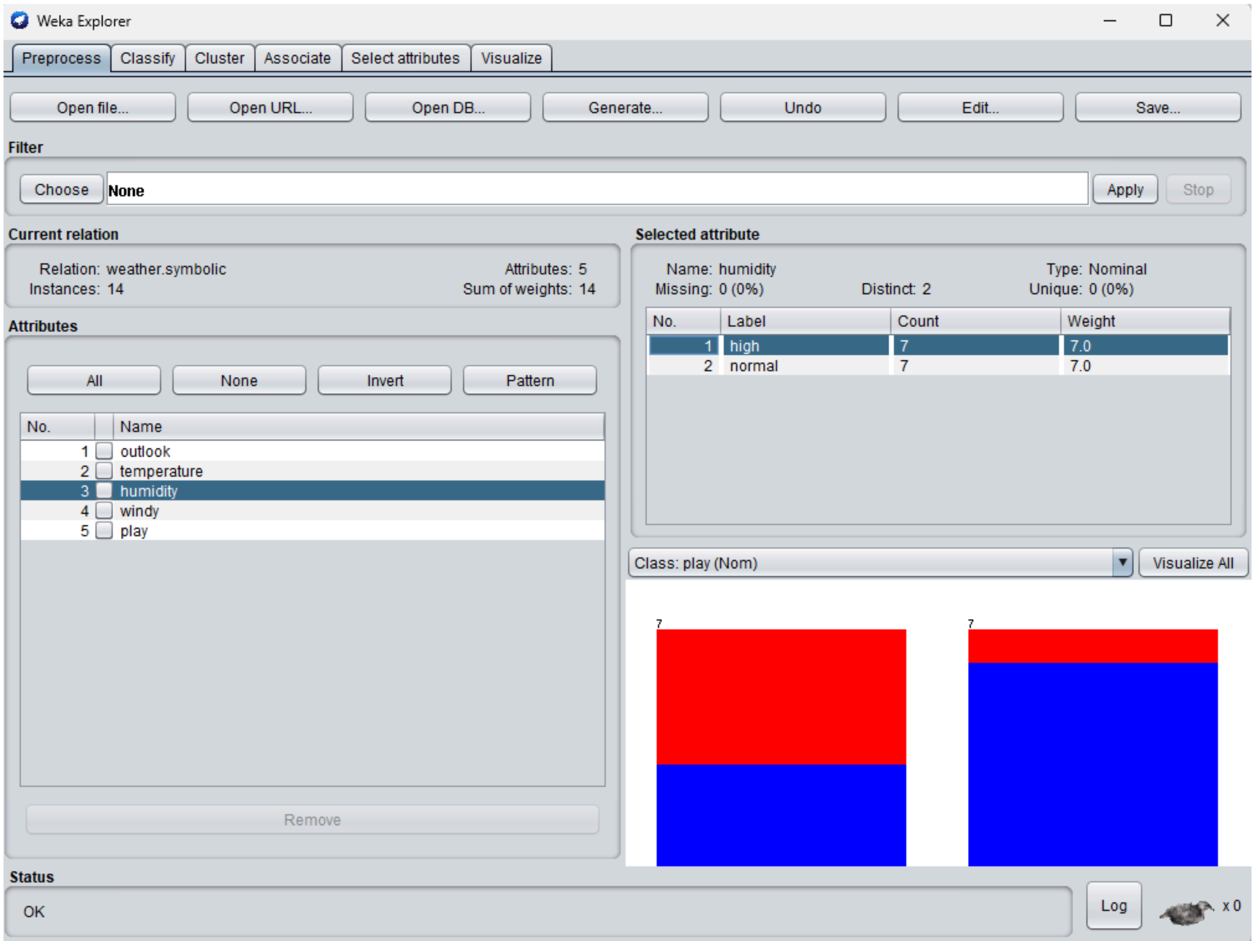
d) Which attribute best separates the data?

Outlook best seperates the data as it has the highest information gain of 0.2467.



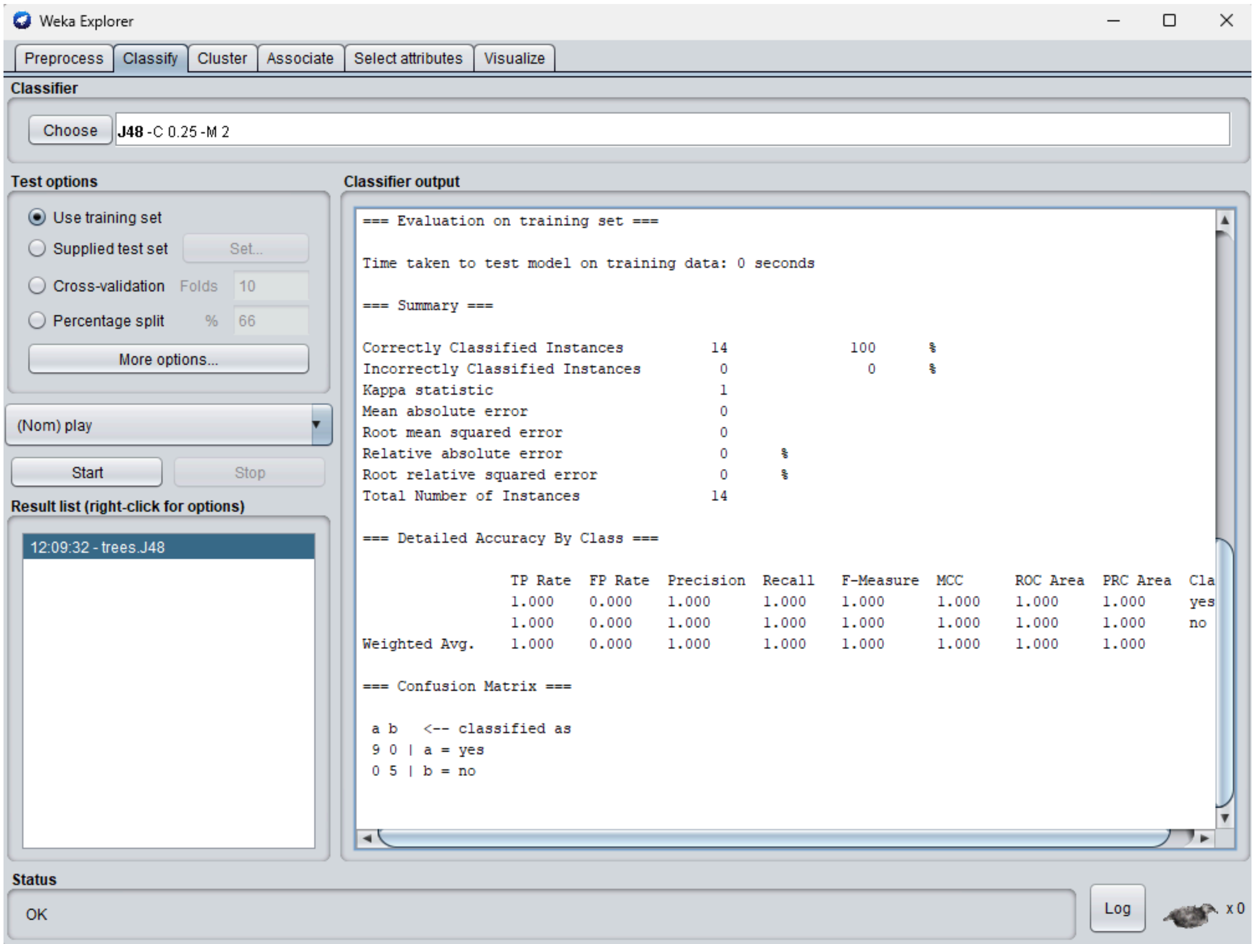
e) How many elements from the data set have the humidity attribute set as high?

3

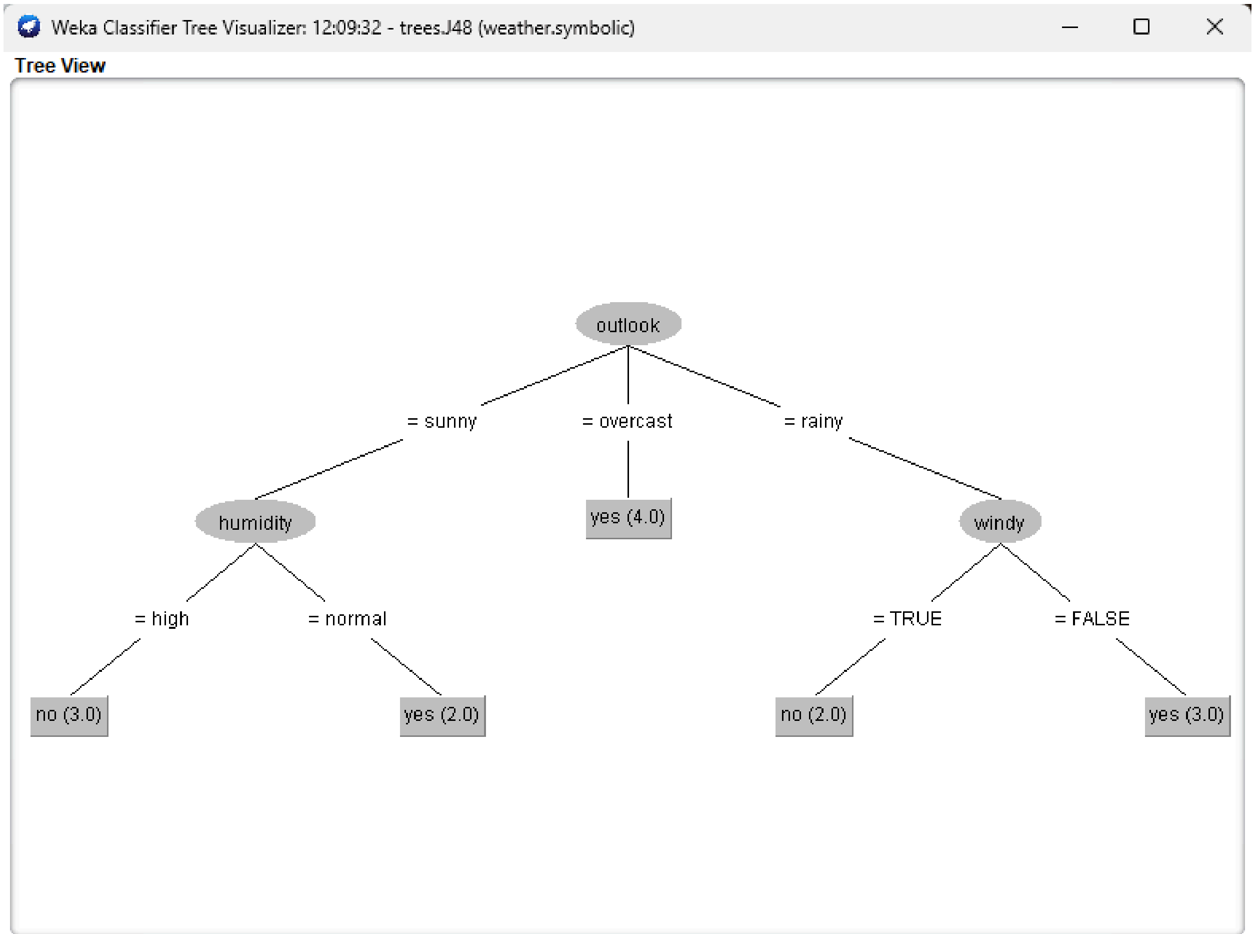


Load and Analyze data:-

The result is showing that the classifier is correctly classifying all the instances correctly and showing TN and FN = 0 in the confusion matrix. Since, we are using a training set for testing so it's quite obvious that the classifier will give 100% accuracy.



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) play
Start
Stop
Result list (right-click for options)
12:09:32 - trees.J48
Classifier output
=== Evaluation on training set ===
Time taken to test model on training data: 0 seconds
=== Summary ===
Correctly Classified Instances 14 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0
Root mean squared error 0
Relative absolute error 0 %
Root relative squared error 0 %
Total Number of Instances 14
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Cla
1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 yes
1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 no
Weighted Avg. 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000
=== Confusion Matrix ===
a b <-- classified as
9 0 | a = yes
0 5 | b = no
Status
OK
Log
x 0

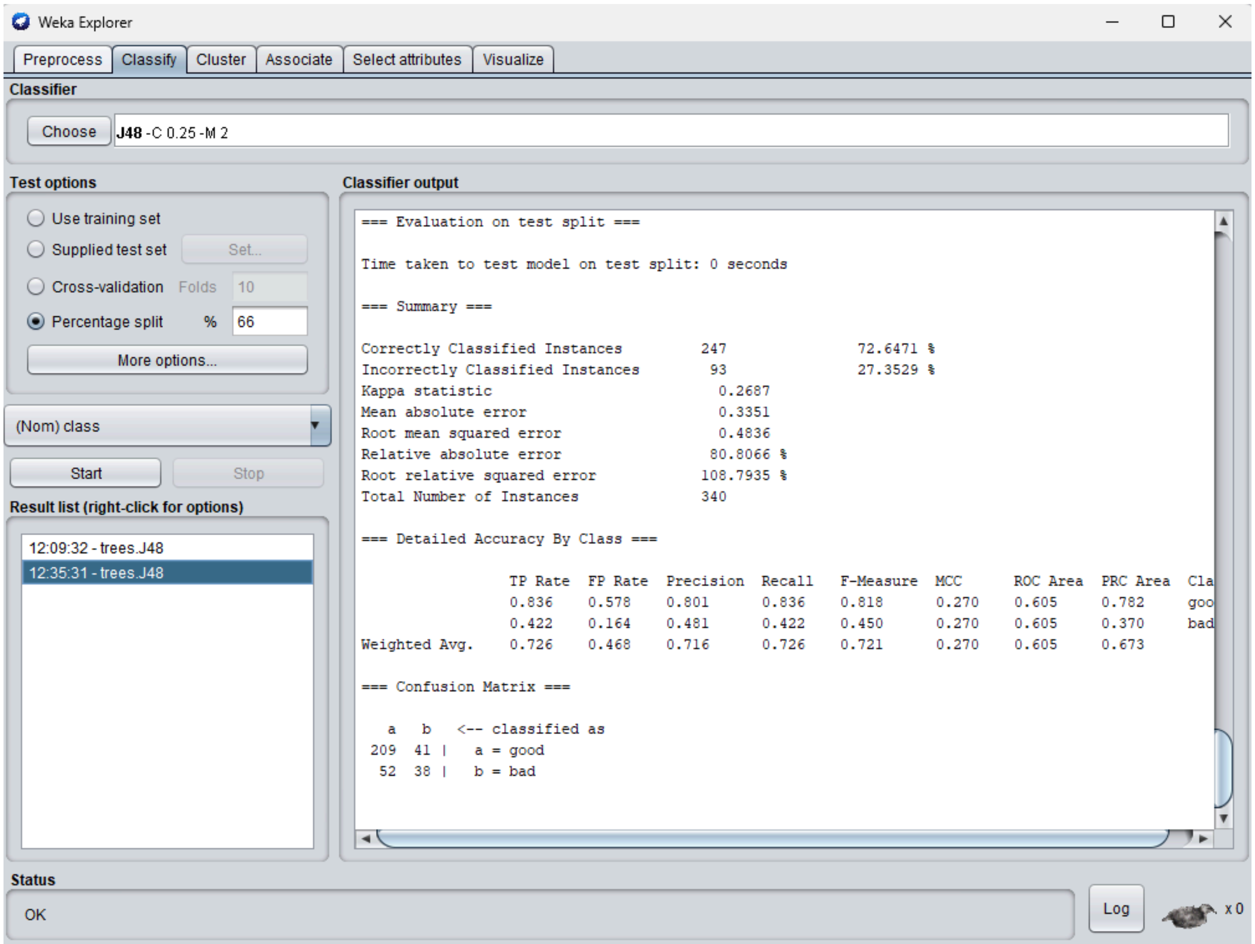


Classification accuracy:-

Result of J48 algorithm:

% of Correctly classified instances: 72.6471%
% of Incorrectly classified instances: 27.3529%

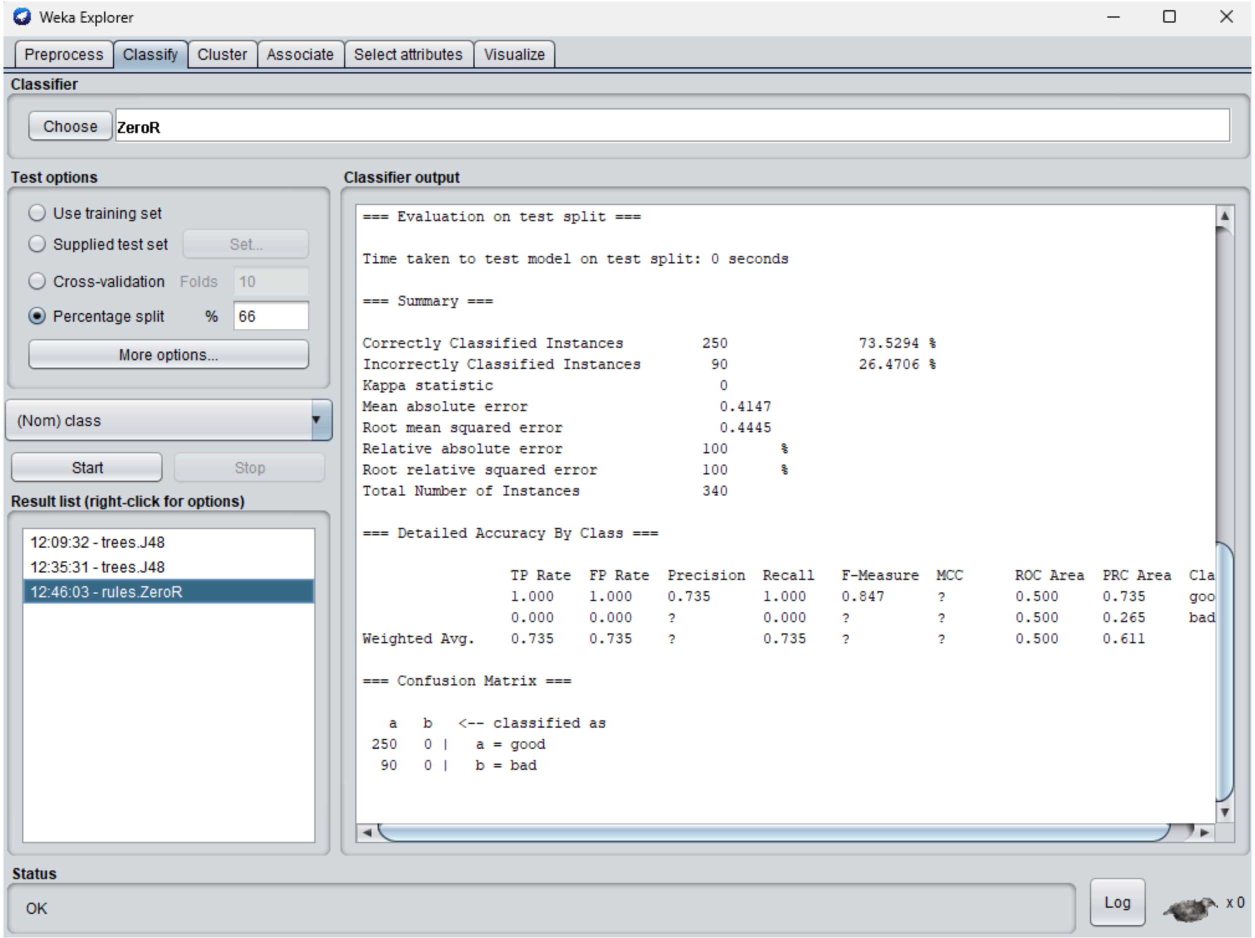
I think this is an acceptable result.



Result of ZeroR algorithm:

% of Correctly classified instances: 73.5294%
% of Incorrectly classified instances: 26.4706%

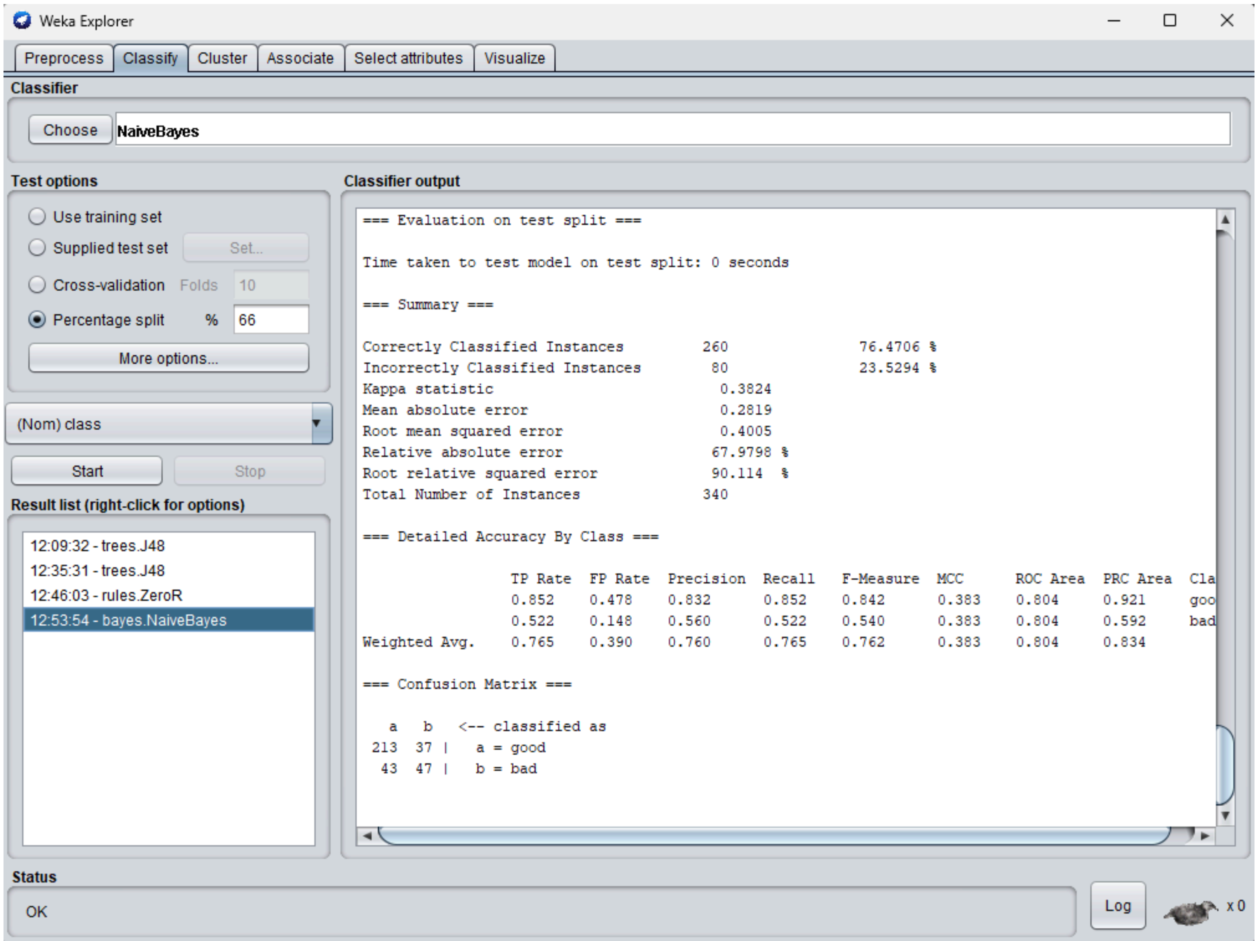
This algorithm is giving better results than the J48 algorithm.



Result of Naive Bayes algorithm:

% of Correctly classified instances: 76.4706%
% of Incorrectly classified instances: 23.5294%

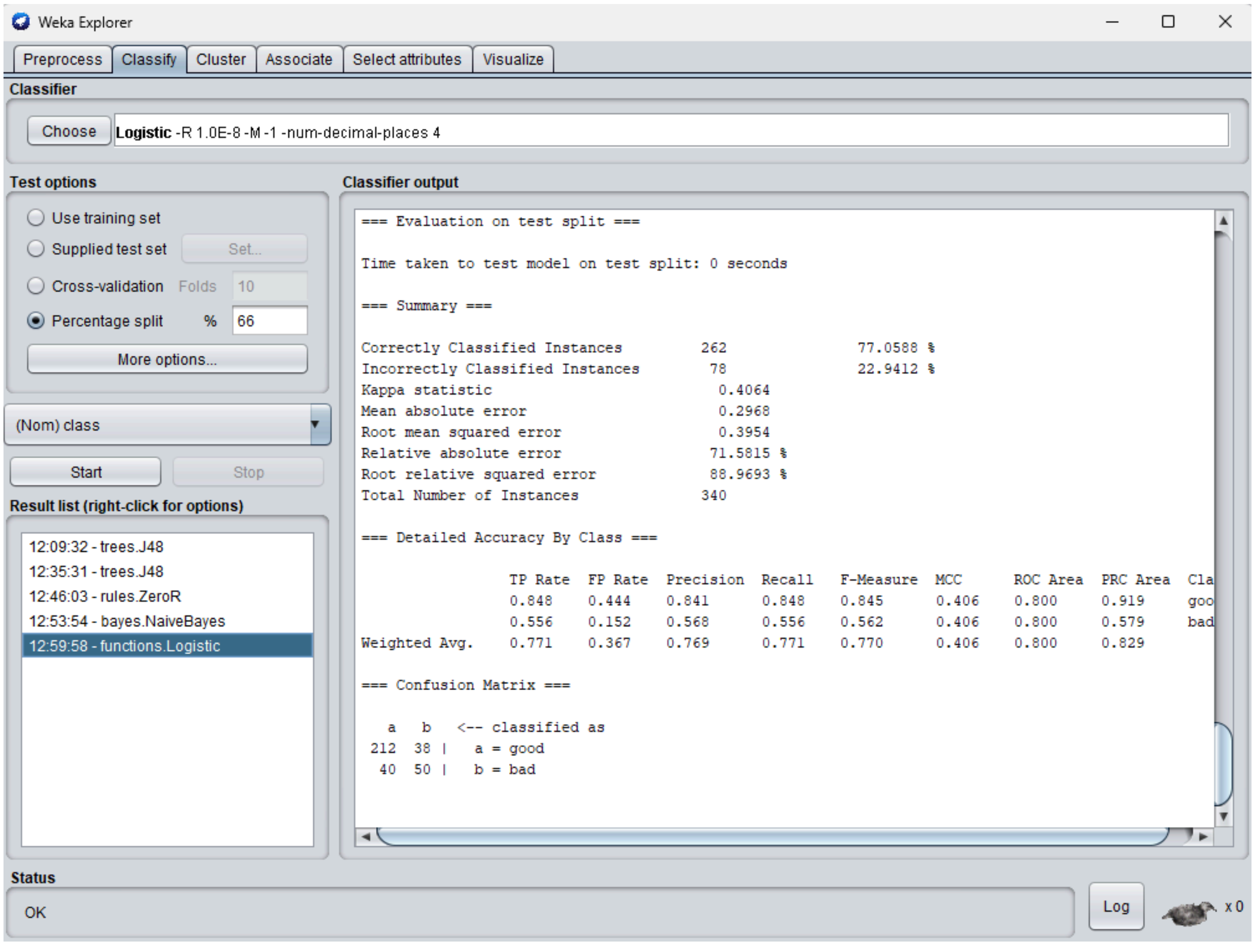
This algorithm is giving better results than both J48 algorithm and ZeroR algorithm.



Result of Logistic Regression algorithm:

% of Correctly classified instances: 77.0588%
% of Incorrectly classified instances: 22.9412%

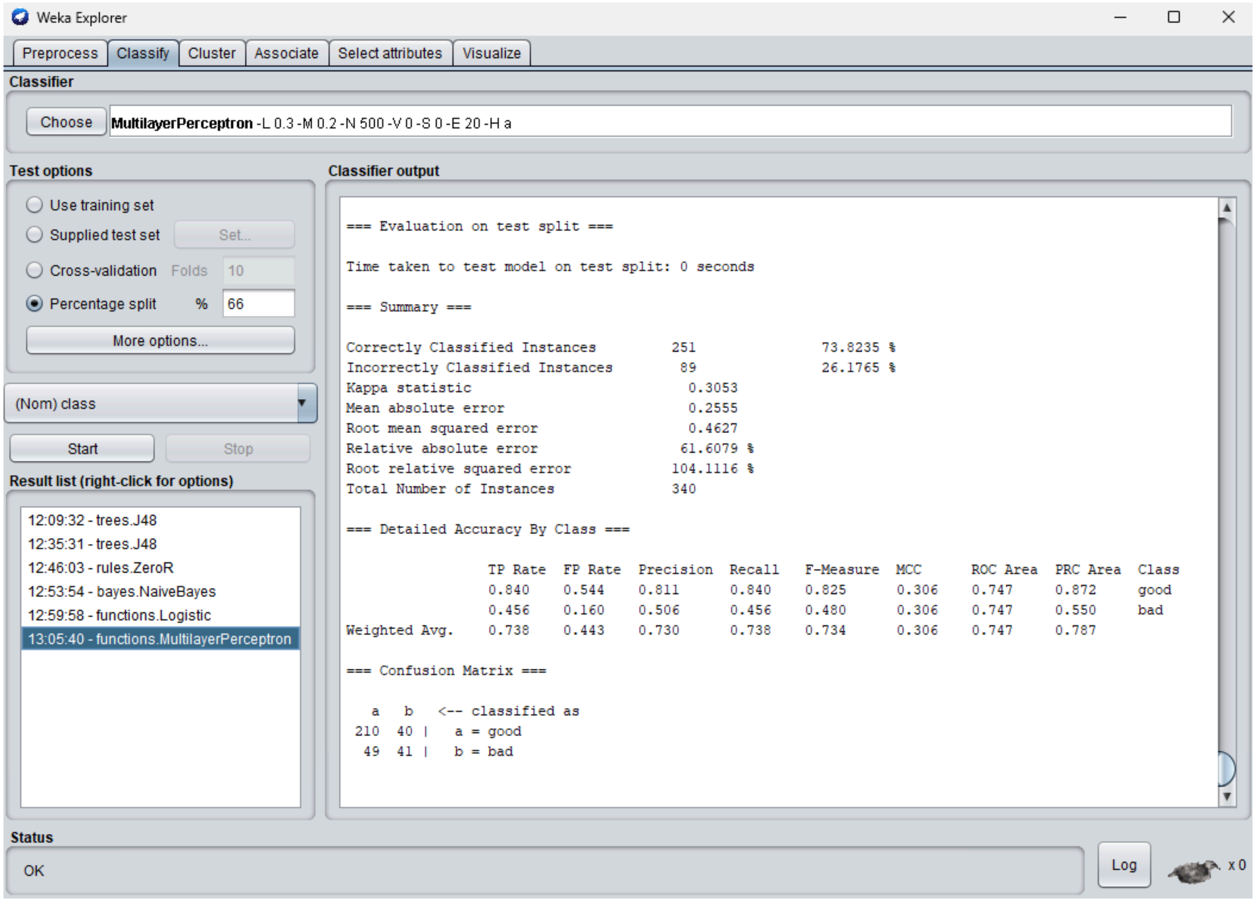
This algorithm is giving better results than all the previous algorithms.



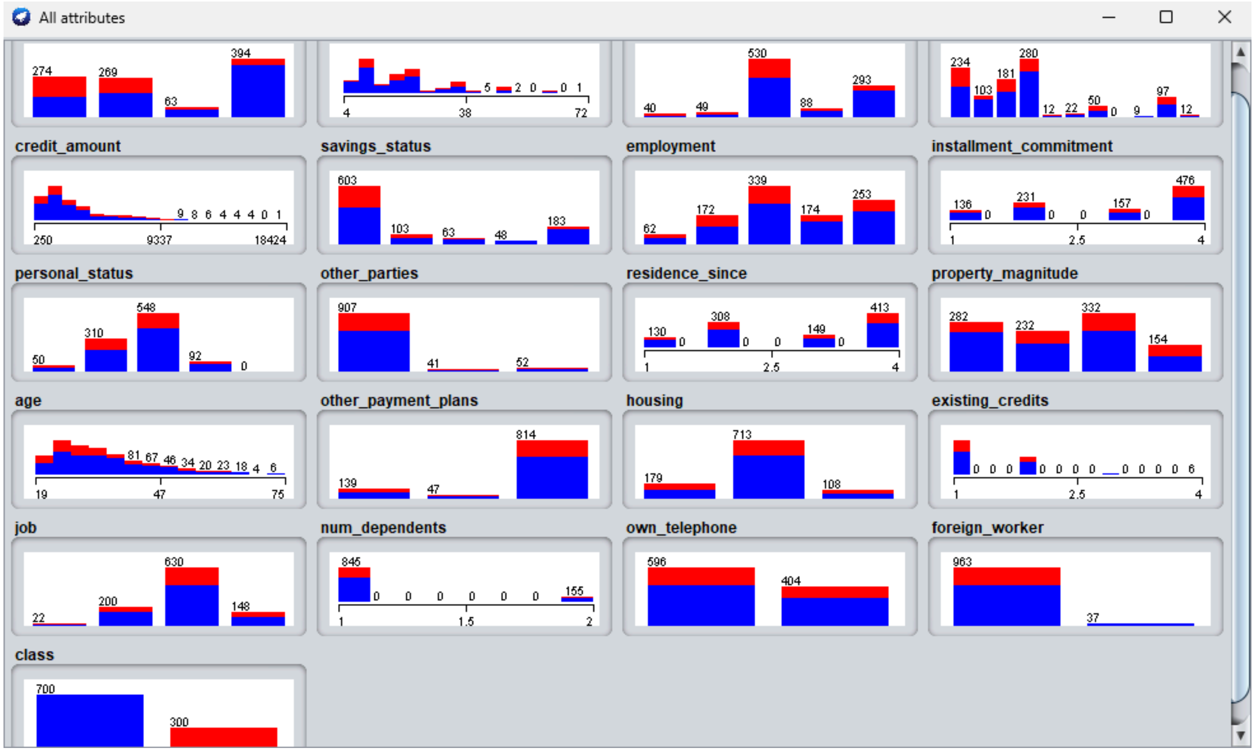
Result of Multi Layer Perceptron Classifier algorithm:

% of Correctly classified instances: 73.8235%
% of Incorrectly classified instances: 26.1765%

This algorithm is giving better results than the J48 and ZeroR algorithm. While, Naive Bayes Classifier, Logistic Regression and Multi layer perceptron are producing more robust results.



Go to the ‘Preprocess’ tab and see how the distribution of the attribute defines whether the set is good or bad. What would be the effectiveness of an algorithm that regardless of the value of attributes would "shoot" that the user is reliable or not?



In this dataset, the class attribute is imbalance so there is a higher chance of dominance of one attribute over another. This will result in false predictions. Also, some attributes like duration, credit_amount, age, e.t.c are positively skewed. If there’s too much skewness in data, then many statistical models won’t work efficiently. Because, in skewed data tail regions may act as an outlier and outlier negatively affects a model’s performance.

Why is it worth taking a look at the data before attempting a classification task?

Real world datasets are very dirty and flawed data sets will be nearly useless for any machine learning model. So, data preprocessing and analysis are the key steps to perform in order to make data suitable for machine learning models.