MD SUBMAN RAFID

18-36131-1

Mid Assignment

Here I have collect dataset about weather information. For example I want to play tennis outside so is the weather today is sunny rainy or overcast? Should I play outside or not? For that I construct a dataset about weather forecasts. There's five attribute in my dataset outlook, temperature, humidity, windy & play. There are 3 labels or sub-attribute in the outlook are sunny rainy and overcast. 3 levels in the temperature attribute are hot mild & cool. 2 levels in the humidity attribute are high & normal. 2 levels in the windy attribute are True & false. 2 levels in the play attribute are yes or no. I'm going to import ta dataset in the WEKA tool. I'm going to apply the classification algorithms on the dataset which are decision tree, KNN & Naïve Bayes. & see which algorithm's correctness is better then the other two.
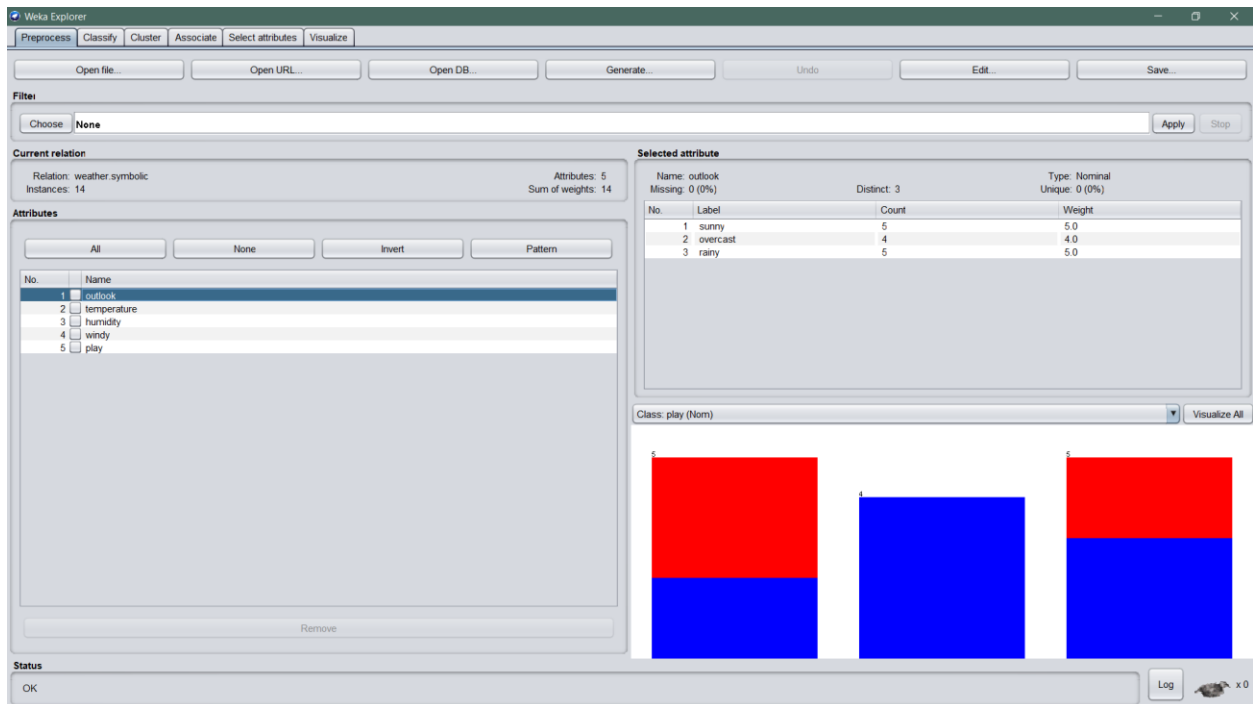
Decision Tree

A decision tree is an supervised machine learning algorithm used for predicting outcomes based on certain rules and is done by partitioning the data into subsets. Here we applied the decision tree algorithm on the dataset. From the classification output we can see the correctly classified is 50% and incorrect is also 50%. Relative absolute error is 87.5%.

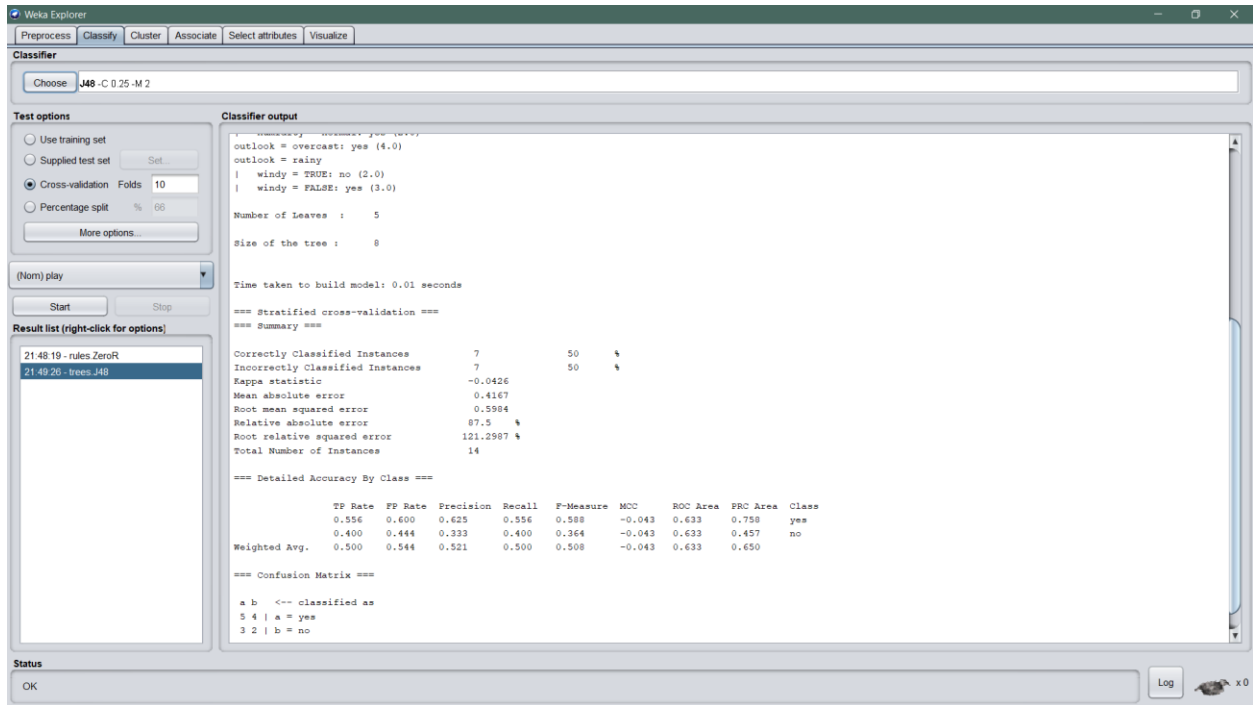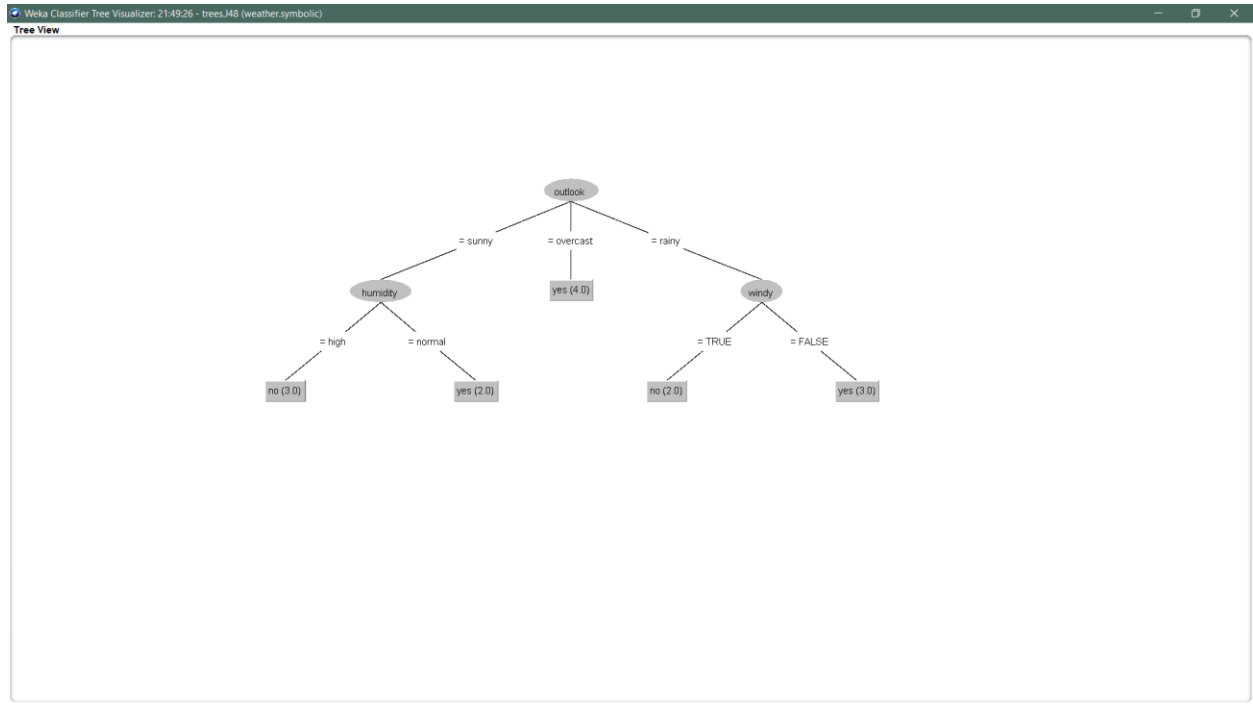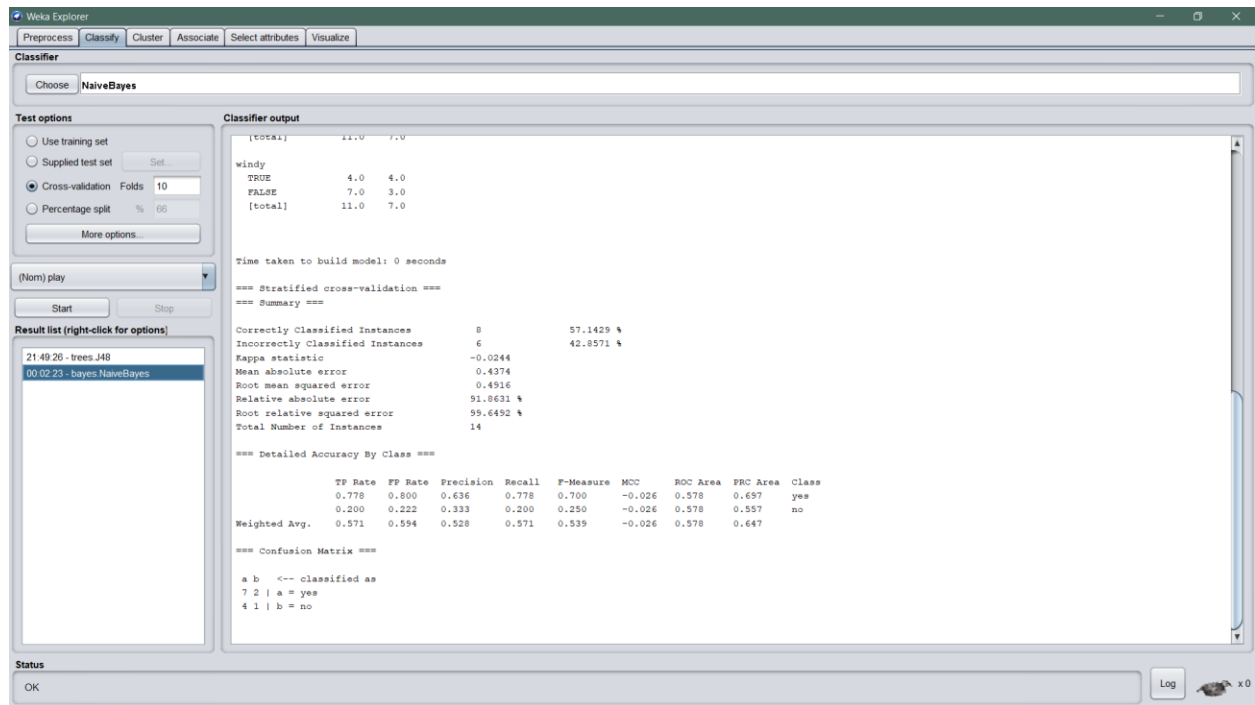Fig: Classifier output

Fig: visualization tree

Naïve Bayes

Naïve Bayes is one of the fast & easy machine learning algorithms to predict a class of datasets. Here we used the naïve bayes algorithm on the dataset. From the classification output we can see that correctly classified is 57.142% and incorrectly classified is 42.85%. Relative absolute error is 91.86%.

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) play

Start   Stop

Result list (right-click for options)

21:49:26 - trees.J48
00:02:23 - bayes NaiveBayes

Classifier output

```
[total]      11.0   7.0

windy
   TRUE       4.0   4.0
   FALSE      7.0   3.0
   [total]   11.0   7.0


Time taken to build model: 0 seconds

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

Correctly Classified Instances         8               57.1429 %
Incorrectly Classified Instances       6               42.8571 %
Kappa statistic                       -0.0244
Mean absolute error                    0.4374
Root mean squared error                0.4916
Relative absolute error               91.8631 %
Root relative squared error           99.6492 %
Total Number of Instances             14

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  MCC     ROC Area  PRC Area  Class
               0.778    0.800    0.636      0.778   0.700      -0.026  0.578     0.697     yes
               0.200    0.222    0.333      0.200   0.250      -0.026  0.578     0.557     no
Weighted Avg.  0.571    0.594    0.528      0.571   0.539      -0.026  0.578     0.647

=== Confusion Matrix ===

 a b   <-- classified as
 7 2 | a = yes
 4 1 | b = no
```

Status

OK   Log   x0

KNN

K-nearest neighbors algorithm is a type of supervised machine learning algorithm which can be used for both classification as well as regression. Here we applied the KNN algorithm on the dataset. From the classification output we can see that the correctly classified is 57.142% and incorrectly classified is 42.85%. Relative absolute error is 103.137%
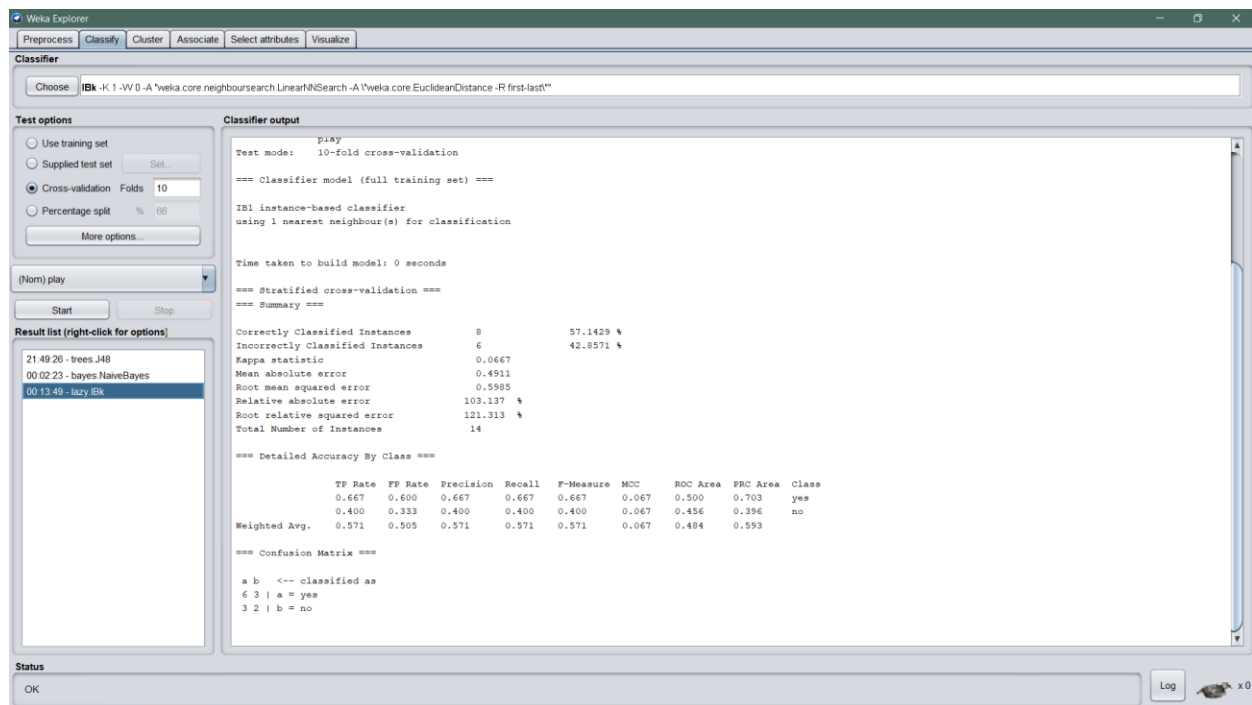
From the above results I choose the naïve bayes technique for the dataset. Because its correctness is higher than decision tree algorithm but same as the KNN algorithm. But in KNN algorithm the relative absolute error is higher than the naïve bayes's error. I choose naïve bayes because it is very good for the larger datasets where KNN is a lazy learning algorithm because it dies not have a specialized training phase and uses all the data for training while classification. KNN is mainly used for classification predictive problems in industry. Naïve bayes is much faster than KNN. Naïve bayes tree uses decision tree as the general structure and deploys naïve Bayesian classifiers at leaves. The intuition is that naïve bayes classifiers works better than decision tree when the sample dataset is small. I choose naïve bayes because it will give faster and correct output results than other two algorithms. Naïve bayes performs well in multi-class predictions as compared to the other algorithms. So by using the naïve bayes algorithm we can solve the above dataset by 3 steps- converting the given dataset into frequency tables, generate likelihood table by finding the probabilities of given features and use bayes theorem to calculate the posterior probability.