**NCTU Pattern Recognition, Homework 3**

A102548 蔡佩璇

**Part. 1, Coding (60%):**

The result of.

1. **Gini Index or Entropy is often used for measuring the “best” splitting of the data. Please compute the Entropy and Gini Index of this array *np.array([1,2,1,1,1,1,2,2,1,1,2])*.**
2. **Implement the Decision Tree algorithm (CART, Classification and Regression Trees) and train the model by the given arguments, and print the accuracy score on the test data. You should implement two arguments for the Decision Tree algorithm,**
3. **Criterion: The function to measure the quality of a split. Your model should support “gini” for the Gini impurity and “entropy” for the information gain.**
4. **Max\_depth: The maximum depth of the tree. If Max\_depth=None, then nodes are expanded until all leaves are pure. Max\_depth=1 equals split data once.**

**2.1. Using Criterion=‘gini’, showing the accuracy score of test data by Max\_depth=3 and Max\_depth=10, respectively.**

**2.2. Using Max\_depth=3, showing the accuracy score of test data by Criterion=‘gini’ and Criterion=’entropy’, respectively.**

1. **Plot the feature importance of your Decision Tree model. You can use the model from Question 2.1, max\_depth=10. (You can use simply counting to get the feature importance instead of the formula in the reference, more details on the sample code.**
2. **Implement the AdaBoost algorithm by using the CART you just implemented from question 2. You should implement one argument for the AdaBoost.**
3. **N\_estimators: The number of trees in the forest.**

**4.1. Showing the accuracy score of test data by n\_estimators=10 and n\_estimators=100, respectively.**

1. **the Random Forest algorithm by using the CART you just implemented from question 2. You should implement three arguments for the Random Forest.**
2. **N\_estimators: The number of trees in the forest.**
3. **Max\_features: The number of features to consider when looking for the best split.**
4. **Bootstrap: Whether bootstrap samples are used when building trees.**

**5.1. Using Criterion=‘gini’, Max\_depth=None, Max\_features=sqrt(n\_features), Bootstrap=True, showing the accuracy score of test data by n\_estimators=10 and n\_estimators=100, respectively.**

**5.2. Using Criterion=‘gini’, Max\_depth=None, N\_estimators=10, Bootstrap=True, showing the accuracy score of test data by Max\_features=sqrt(n\_features) and Max\_features=n\_features, respectively.**

1. **Tune the hyperparameter, perform feature engineering or implement more powerful ensemble methods to get a higher accuracy score. Screenshot your tests score on the report. Please note that only the ensemble method can be used. The neural network method is not allowed.**

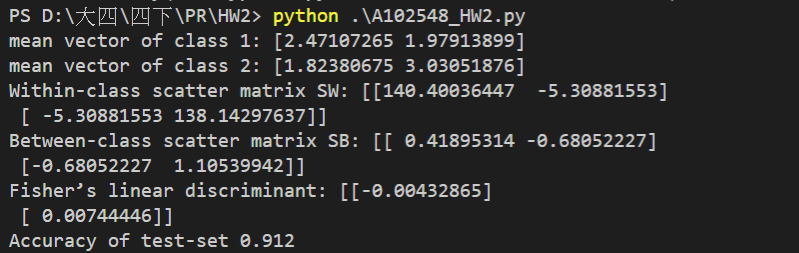


Figure.1 Screen shot of the result

**Part. 2, Questions (40%):**

1. **Consider a data set comprising 400 data points from class C1 and 400 data points from class C2. Suppose that a tree model A splits these into (300, 100) at the first leaf node and (100, 300) at the second leaf node, where (n, m) denotes that n points are assigned to C1 and m points are assigned to C2. Similarly, suppose that a second tree model B splits them into (200, 400) and (200, 0). Evaluate the misclassification rates for the two trees and hence show that they are equal. Similarly, evaluate the cross-entropy and Gini index for the two trees and show that they are both lower for tree B than for tree A. Define pk to be the proportion of data points in region R assigned to class k, where k = 1, . . . , K.**
2. Misclassification rate:

For tree A:

For tree B:

Therefore, misclassification rate for the two tree are equal.

1. Cross-entropy and Gini index:

For tree A:

,

,

For tree B:

,

,

We computed that the entropy A > the entropy B, and Gini A > Gini B. Therefore, both cross-entropy and Gini index are lower for tree B than for tree A.

1. **By making a variational minimization of the expected exponential error function given by (1) with respect to all possible functions y(x), show that the minimizing function is given by (2). Define t is target variable ∈ {−1, 1}, x is input vector.**

**(1)**

**(2)**

Set the partial derivative of the expected exponential error of y(x) to 0 to make minimization:

Because don’t have y(x), we can ignore it. Also, we know that

Therefore, we can get

Then we take to get y(x)

**#**