NYCU Pattern Recognition, Homework 3

**311553010, 陳姿羽**

**Part. 1, Coding (70%)**:

1. (5%) Compute the Entropy and Gini index of the array provided in the sample code, using the formulas on page 6 of the HW3 slide.

['+' '+' '+' '+' '+' '-']: entropy = 0.6500224216483541

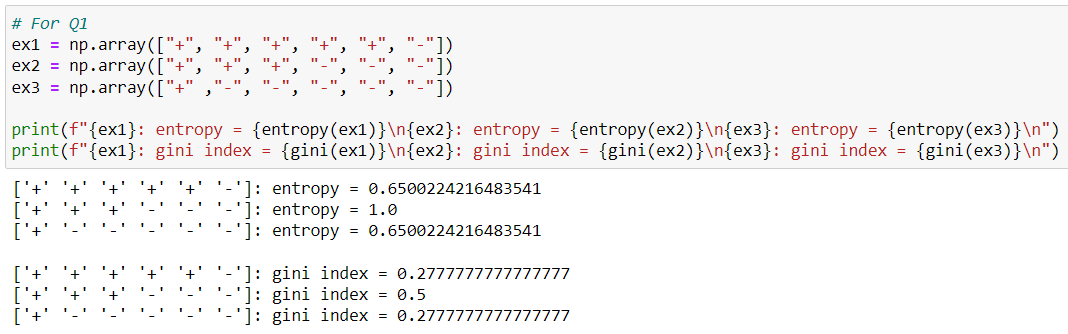
['+' '+' '+' '-' '-' '-']: entropy = 1.0

['+' '-' '-' '-' '-' '-']: entropy = 0.6500224216483541

['+' '+' '+' '+' '+' '-']: gini index = 0.2777777777777777

['+' '+' '+' '-' '-' '-']: gini index = 0.5

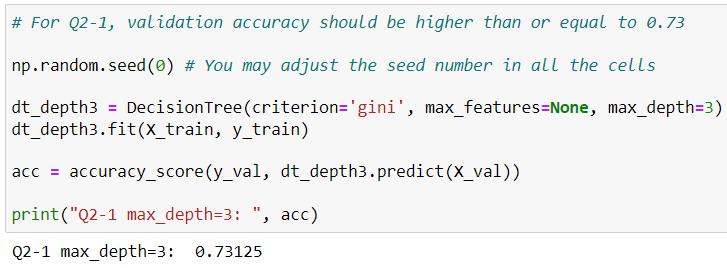
['+' '-' '-' '-' '-' '-']: gini index = 0.2777777777777777

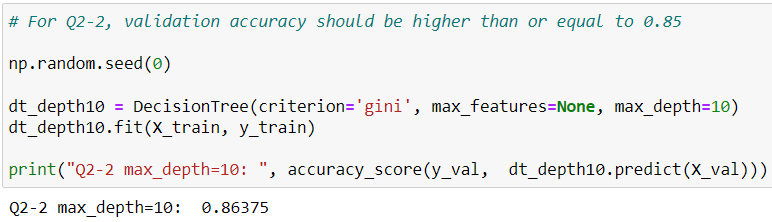


1. (10%) Show the accuracy score of the validation data using criterion='gini' and max\_features=None for max\_depth=3 and max\_depth=10, respectively.

Q2-1 max\_depth=3: 0.73125

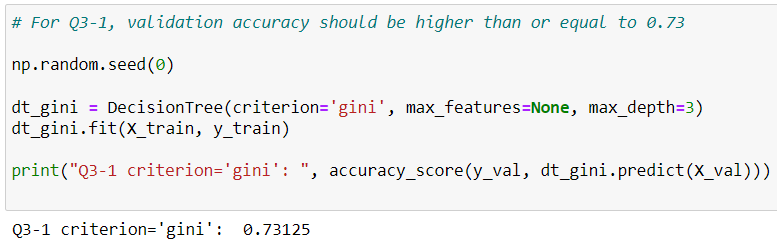
Q2-2 max\_depth=10: 0.86375

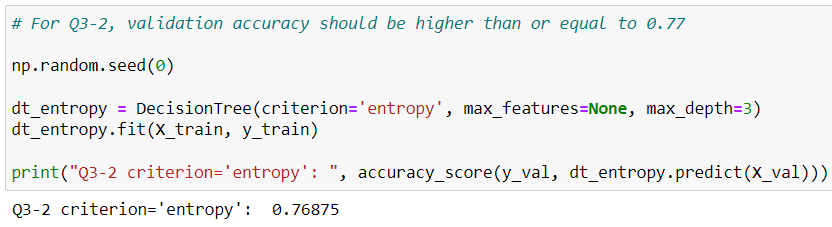


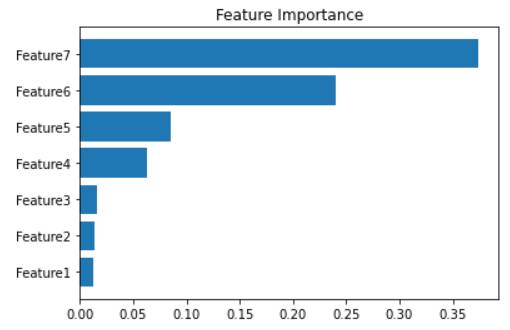


1. (10%) Show the accuracy score of the validation data using max\_depth=3 and max\_features=None, for criterion='gini' and criterion='entropy', respectively.  
   Q3-1 criterion='gini': 0.73125

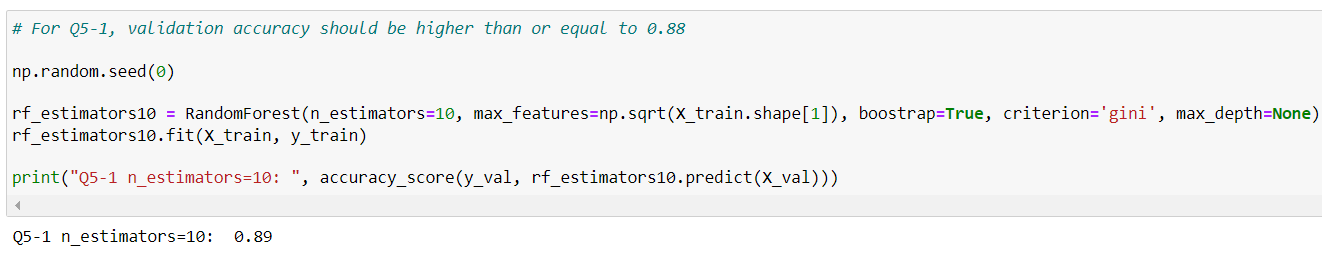
Q3-2 criterion='entropy': 0.76875

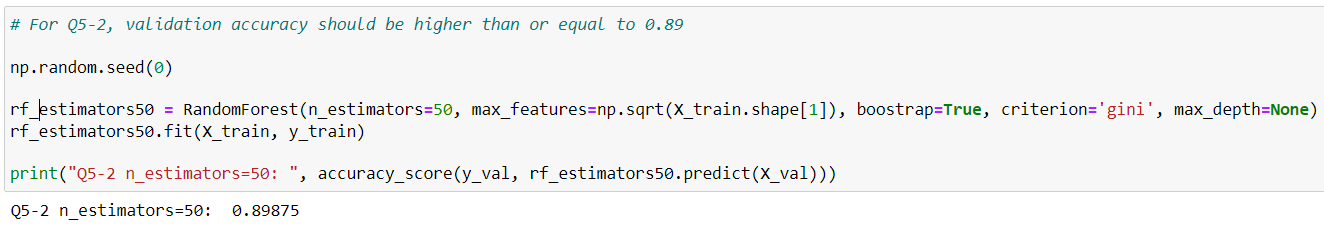




1. (5%) Train your model using criterion='gini', max\_depth=10 and max\_features=None. Plot the feature importance of your decision tree model by simply counting the number of times each feature is used to split the data.  
   
2. (10%) Show the accuracy score of the validation data using criterion='gini', max\_depth=None, max\_features=sqrt(n\_features), and bootstrap=True, for n\_estimators=10 and n\_estimators=50, respectively.  
   Q5-1 n\_estimators=10: 0.89

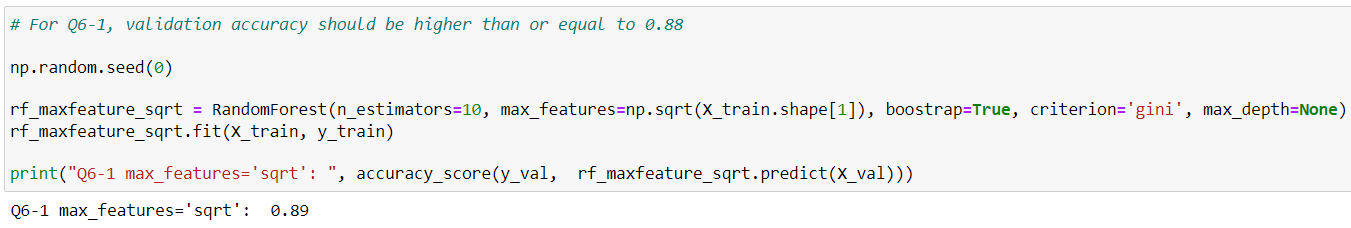
Q5-2 n\_estimators=50: 0.89875

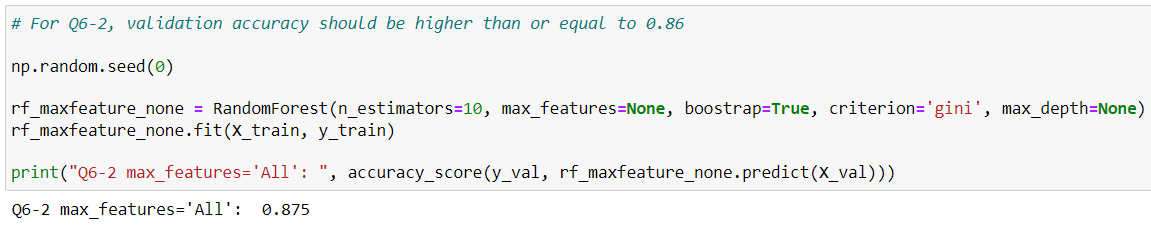




1. (10%) Show the accuracy score of the validation data using criterion='gini', max\_depth=None, n\_estimators=10, and bootstrap=True, for max\_features=sqrt(n\_features) and max\_features=n\_features, respectively.  
   Q6-1 max\_features='sqrt': 0.89

Q6-2 max\_features='All': 0.875





1. (20%) Explain how you chose/design your model and what feature processing you have done in detail. Otherwise, no points will be given.   
   random seed=220

n\_estimators=50

max\_features=np.sqrt(X\_train.shape[1])

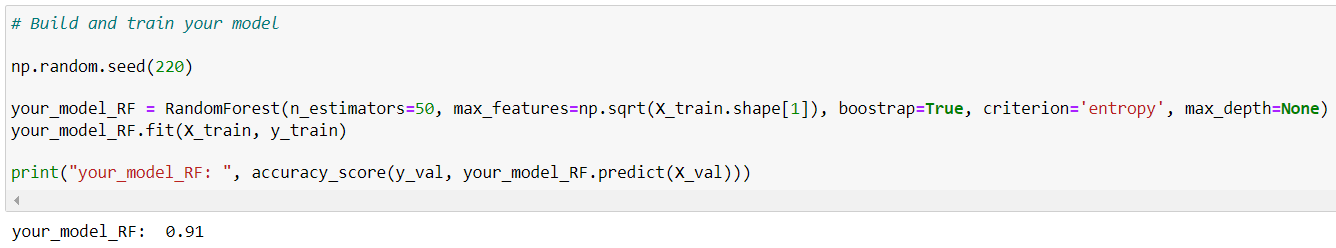
boostrap=True

criterion='entropy'

max\_depth=None  
your\_model\_RF accuracy\_score: 0.91

First, I used the models of decision tree and random forest respectively to choose the appropriate model. However, I found that it was difficult for the decision tree to achieve an accuracy higher than 0.9, so I chose random forest model. Next, I began adjusting the parameters, referencing the parameters used in Q5-2. I found that the accuracy score was higher when using Entropy as the criterion rather than Gini index, and that the accuracy score did not change significantly when n\_estimators was greater than 50. At this point, the accuracy score had reached 0.905. Finally, I adjusted the random seed based on the discussions in the forum, and the accuracy score reached 0.91 when using a random seed of 220.

And I also found that the accuracy score using for loop to compute impurity is higher than using np.sum().



**Part. 2, Questions (30%):**

1. Answer the following questions in detail:
2. Why does a decision tree tend to overfit the training set?

When the deeper and more complex the tree, the greater the likelihood of overfitting. Because it captures too much noise data and creates the rules that only apply to the training set.

1. Is it possible for a decision tree to achieve 100% accuracy on the training set?

Yes. If we keep add branches until each instance has its own leaf, it is possible to achieve 100% accuracy on the training set. But the result would be severe overfitting and poor generalization to new data.

1. List and describe at least three strategies we can use to reduce the risk of overfitting in a decision tree.
2. Setting a maximum depth: Setting a maximum depth to avoid learning complex and creating the rules that only apply to the training set.
3. Ensemble methods: Combine multiple decision trees to make predictions, such as Random Forest. By combining the predictions of multiple trees, to reduce overfitting and improve the generalization ability of the model.
4. Pruning: Removing some nodes to avoid the tree growing to its full depth. It can improve the ability of the validation set tree.
5. For each statement, answer True or False and provide a detailed explanation:
6. In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.

False. The misclassified examples will go up, but not by the same multiplicative factor.

1. In AdaBoost, weighted training error of the weak classifier on training data with weights tends to increase as a function of *t*.

False. The weighted training error will decrease as a function of *t*.

1. AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

False. We cannot ensure that AdaBoost will eventually give zero training error. If the weak classifier isn't fit the training data or isn't strong, AdaBoost will not give zero training error in any iterations.

1. Consider a data set comprising 400 data points from class and 400 data points from class . Suppose that a tree model A splits these into (200, 400) at the first leaf node and (200, 0) at the second leaf node, where (n, m) denotes that n points are assigned to and m points are assigned to . Similarly, suppose that a second tree model B splits them into (300, 100) and (100, 300). **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**. Define to be the proportion of data points in region R assigned to class k, where k = 1, . . . , K.

|  |
| --- |
| Misclassification rates |
|  |
| Cross-entropy |
|  |
| Gini index |
|  |