NYCU Introduction to Machine Learning, Homework 3

Deadline: Nov. 15, 23:59

Part. 1, Coding (80%):

1. (5%) 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])` by the formula below.

```
[4]  print("Gini of data is ", gini(data))

     Gini of data is  0.4628099173553719

[5]  print("Entropy of data is ", entropy(data))

     Entropy of data is  0.9456603046006401
```

2. (10%) 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,
   1) **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.
   2) **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

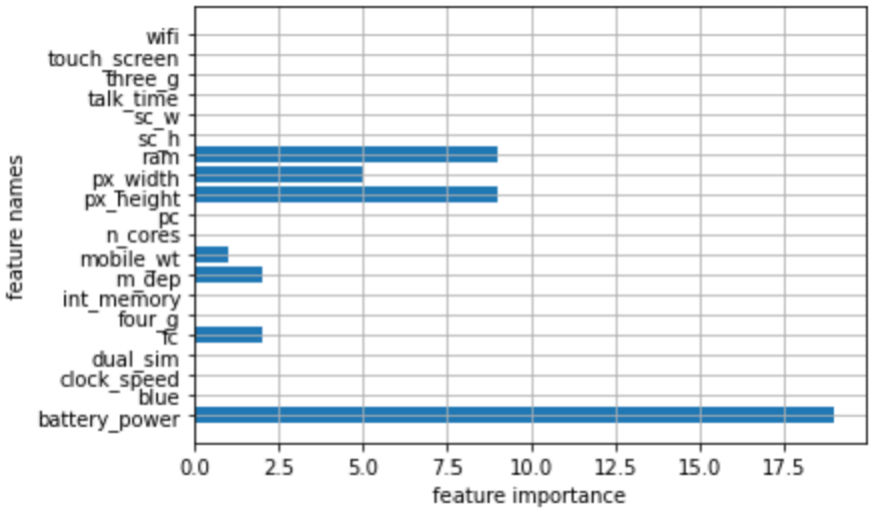
   2.1. Using Criterion='gini', showing the accuracy score of test data by Max_depth=3 and Max_depth=10, respectively.

```
DecisionTree(criterion='gini', max_depth=3), Accuracy Score: 0.9166666666666666
DecisionTree(criterion='gini', max_depth=10), Accuracy Score: 0.9366666666666666
```

   2.2. Using Max_depth=3, showing the accuracy score of test data by Criterion='gini' and Criterion='entropy', respectively.

```
DecisionTree(criterion='gini', max_depth=3), Accuracy Score: 0.9166666666666666
DecisionTree(criterion='entropy', max_depth=3), Accuracy Score: 0.93
```

3. (5%) Plot the feature importance of your Decision Tree model. You can use the model from Question 2.1, max_depth=10.



4. (15%) Implement the AdaBoost algorithm by using the CART you just implemented from question 2. You should implement **one argument** for the AdaBoost.
   1) **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.

```
AdaBoost(n_estimators=10), Accuracy Score: 0.8933333333333333
AdaBoost(n_estimators=100), Accuracy Score: 0.8933333333333333
```

5. (15%) Implement the Random Forest algorithm by using the CART you just implemented from question 2. You should implement **three arguments** for the Random Forest.
   1) **N_estimators**: The number of trees in the forest.
   2) **Max_features**: The number of features to consider when looking for the best split
   3) **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.

```
RandomForest(criterion='gini', n_estimators=10, max_features=np.sqrt(n_features), boostrap=True), Accuracy Score: 0.74
RandomForest(criterion='gini', n_estimators=100, max_features=np.sqrt(n_features), boostrap=True), Accuracy Score: 0.90333
```

   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.

```
RandomForest(criterion='gini', n_estimators=10, max_features=np.sqrt(n_features), boostrap=True), Accuracy Score: 0.68
RandomForest(criterion='gini', n_estimators=10, max_features=n_features, boostrap=True), Accuracy Score: 0.95
```

6. (20%) Tune the hyperparameter, perform feature engineering or implement more powerful ensemble methods to get a higher accuracy score. Please note that only the ensemble method can be used. The neural network method is not allowed.

```
DecisionTree(criterion='gini', max_depth=1), Accuracy Score: 0.8933333333333333
DecisionTree(criterion='gini', max_depth=3), Accuracy Score: 0.9166666666666666
DecisionTree(criterion='gini', max_depth=5), Accuracy Score: 0.94
RandomForest(criterion='gini', n_estimators=3, max_features=19, boostrap=True), Accuracy Score: 0.9566666666666667
RandomForest(criterion='gini', n_estimators=9, max_features=14, boostrap=True), Accuracy Score: 0.96
RandomForest(criterion='gini', n_estimators=9, max_features=20, boostrap=True), Accuracy Score: 0.9666666666666667

Test-set accuarcy score:  0.9666666666666667
```

Part. 2, Questions (30%):

1.

1.1. Why does a decision tree have a tendency to overfit to the training set?

Decision tree(DT) have a tendency to overfit because it is very data sensitive, it examines data in many ways. Even with very small number of variables, there can still have a lot to be examined. The more levels in categorical variable, the more ways of splitting data can be, which eventually causes overfit.

1.2. Is it possible for a decision tree to reach a 100% accuracy in the training set? please explain.

If there is no limits set on DT, then it might occur 100% accuracy. Since at the worst case it will end up making 1 leaf for each observation. However, if the limit is set to DT, then it is impossible to reach 100% accuracy.

1.3. List and describe at least 3 strategies we can use to reduce the risk of overfitting of a decision tree.

We can reduce the risk of overfitting of a decision tree by:

i. Pre-Pruning:

Set maximum depth to prevent DT growing to its full depth.

ii. Post-Pruning:

Differ from Pre-Pruning, Post-Pruning allows DT to grow to its full depth, then removes the tree branches to prevent overfitting.

iii. Random Forest:

Random Forest follows bootstrap sampling and aggregation techniques to prevent overfitting.

2. This part consists of three True/False questions. Answer True/False for each question and briefly explain your answer.
    a. In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.

       True, follows from the update equation

    b. In AdaBoost, weighted training error $\boldsymbol{\varepsilon}_t$ of the t_{th} weak classifier on training data with weights D_t tends to increase as a function of t.

       True. The weights will increase for data that are repeatedly misclassified by the weak classifiers. The weighted training error of the weak classifier on the training data therefore tends to increase.

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

       False, AdaBoost can only achieve zero training error If the data is separable by a linear combination of the weak classifiers.

3. Consider a data set comprising 400 data points from class C_1 and 400 data points from class C_2. 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 C_1 and m points are assigned to C_2. 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** $Entropy = -\sum_{k=1}^{K} p_k \log_2 p_k$ and **Gini index** $Gini = 1 - \sum_{k=1}^{K} p_k^2$ **for the two trees**. Define p_k to be the proportion of data points in region R assigned to class k, where k = 1, . . . , K.