NYCU Introduction to Machine Learning, Homework 3

109550018 郭昀

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])`.

```
Gini of data is  0.4628099173553719
```

```
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

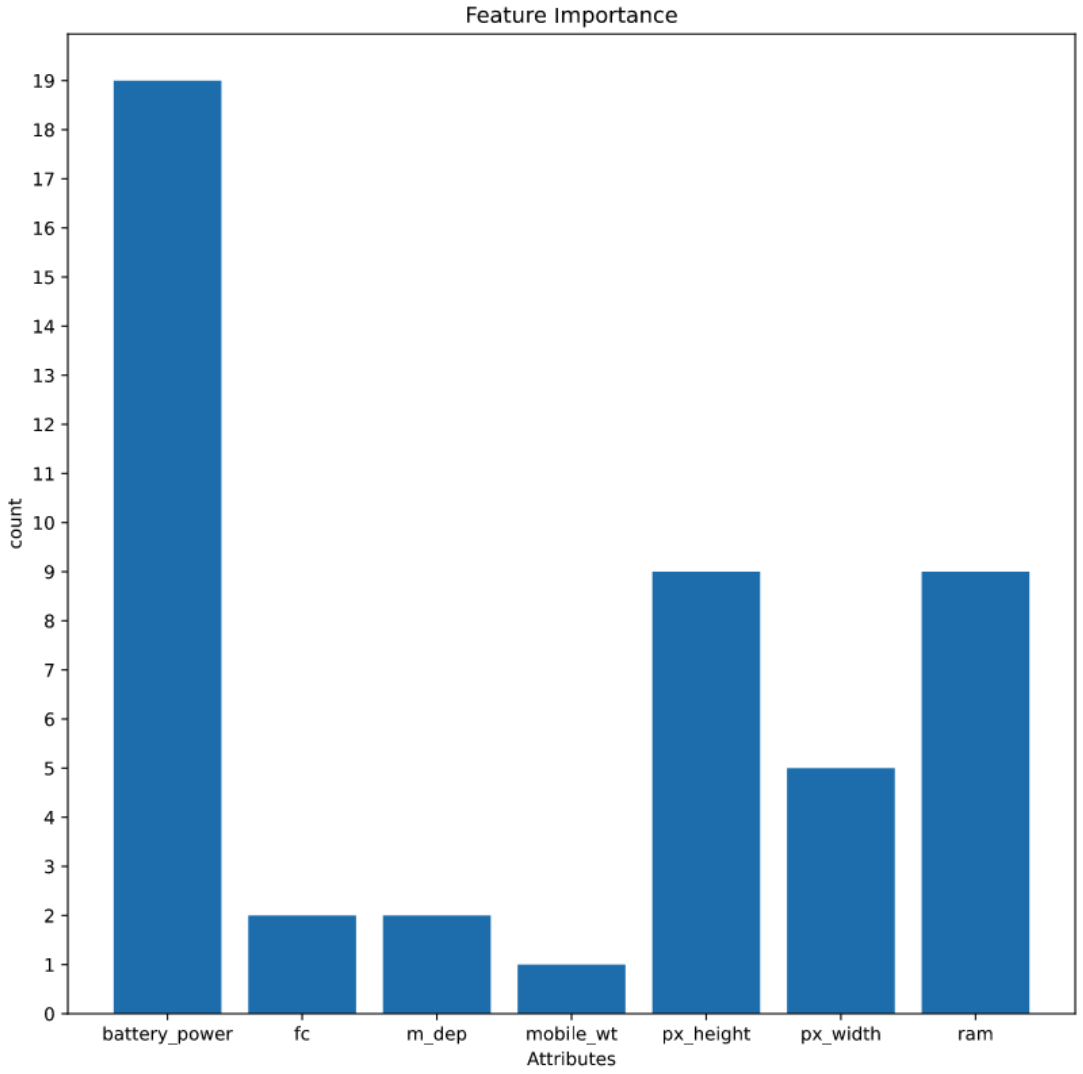
   2.1. Using Criterion='gini', showing the accuracy score of test data by Max_depth=3 and Max_depth=10, respectively.

```
max_depth = 3:  Accuracy of validation data = 0.9166666666666666
max_depth = 10: Accuracy of validation data = 0.9366666666666666
```

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

```
criterion=gini:    Accuracy of validation data = 0.9166666666666666
criterion=entropy: Accuracy of validation data = 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. (You can use simply counting to get the feature importance instead of the formula in the reference, more details on the sample code. **Matplotlib** is allowed to be used)



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.

```
n_estimators = 10:  Accuracy of validation data = 0.9366666666666666
n_estimators = 100:  Accuracy of validation data = 0.9766666666666667
```

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.

```
n_estimators = 10:  Accuracy of validation data = 0.9266666666666666
n_estimators = 100:  Accuracy of validation data = 0.9466666666666667
```

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.

```
random feature: Accuracy of validation data = 0.94
all feature:    Accuracy of validation data = 0.9533333333333334
```

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.

Accuracy	Your scores
acc > 0.975	20 points
0.95 < acc <= 0.975	15 points
0.9 < acc <= 0.95	10 points
acc < 0.9	0 points

Part. 2, Questions (30%):

1. **Why does a decision tree have a tendency to overfit to the training set? Is it possible for a decision tree to reach a 100% accuracy in the training set? please explain. List and describe at least 3 strategies we can use to reduce the risk of overfitting of a decision tree.**

   A decision tree tends to overfit to the training set because we can always find out a way to split a dataset on a node which may result in depending only on train data when building the tree node and become overfitting. This also tells the reason why a decision tree can reach 100% accuracy in the training set. There are several strategies to reduce the risk of overfitting a decision tree. First, define the maximum depth of a tree. Second, define the minimum length of a dataset that we can split. Third, check whether all data in the dataset are in the same class, if yes, we do not split it anymore. All the strategies above can prevent a decision tree from splitting the train set too many time which result in 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. It can be found out from the weight updating formula in the slides.

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

      True. The goal of each AdaBoost iteration is to classify data that is hard to classify. The weights of each data will increase if it is misclassified by the weak classifiers. Therefore, the weighted training error ε_t of the t_{th} weak classifier on training data with weights D_t tends to increase as a function of t

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

      False. It may be influenced by the train data. If the train data cannot be separated by a linear combination of the weak classifiers we are using, training error will never reach 0 no matter how many iterations are performed.

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.**

Misclassification rate of tree A = 200/800 + 0/800 = 1/4

Misclassification rate of tree B = 100/800 + 100/800 = 1/4

→ misclassification rate of tree A = misclassification rate of tree B

Cross entropy of tree A

= - (1/3 * log2(1/3) + 2/3 * log2(2/3)) * (3/4) - (1 * log2(1)) * (1/4)

= 0.6887218755408672

Cross entropy of tree B

= - (3/4 * log2(3/4) + 1/4 * log2(1/4)) * (1/2) - (1/4 * log2(1/4) + 3/4 * log2(3/4)) * (1/2)

= 0.8112781244591328

Gini of tree A

= (1 - (1/3)^2 - (2/3)^2) * (3/4) + (1 - 1^2) * (1/4) = 1/3

Gini of tree B

= (1 - (3/4)^2 - (1/4)^2) * (1/2) + (1 - (1/4)^2 - (3/4)^2) * (1/2) = 3/8