Report of Intro- to Machine Learning, Homework 3

109550027 紀竺均

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

```
print("Gini of data is ", gini(data))
```
✓ 0.6s Python

```
Gini of data is  0.4628099173553719
```

```
print("Entropy of data is ", entropy(data))
```
✓ 0.6s Python

```
Entropy of data is  0.9456603046006401
```

2. (10%) Implement the Decision Tree algorithmand train the model by the given arguments, and print the accuracy score on the test data.

(1) Using Criterion=‘gini’, showing the accuracy score of test data by Max_depth=3 and Max_depth=10, respectively.

```
clf_depth3 = DecisionTree(criterion='gini', max_depth=3)
clf_depth3.fit(x_train, y_train)
y_pred1 = clf_depth3.predict(valx_test)
acc1 = accuracy_score(valy_test, y_pred1)
clf_depth10 = DecisionTree(criterion='gini', max_depth=10)
clf_depth10.fit(x_train, y_train)
y_pred2 = clf_depth10.predict(valx_test)
acc2 = accuracy_score(valy_test, y_pred2)

print(acc1, acc2)
```
Python

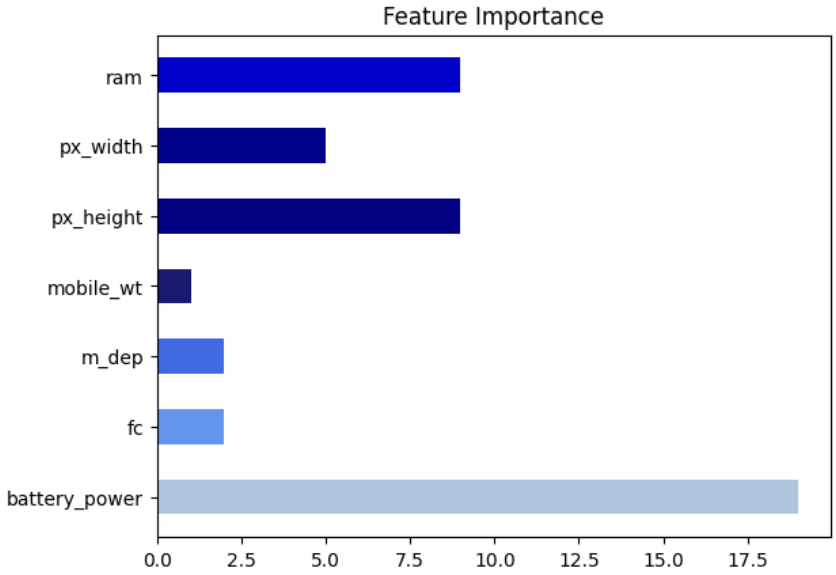
```
0.9166666666666666 0.9366666666666666
```

(2) Using Max_depth=3, showing the accuracy score of test data by Criterion=‘gini’ and Criterion=’entropy’, respectively.

```
clf_gini = DecisionTree(criterion='gini', max_depth=3)
clf_gini.fit(x_train, y_train)
y_pred1 = clf_gini.predict(valx_test)
acc1 = accuracy_score(valy_test, y_pred1)
clf_entropy = DecisionTree(criterion='entropy', max_depth=3)
clf_entropy.fit(x_train, y_train)
y_pred2 = clf_entropy.predict(valx_test)
acc2 = accuracy_score(valy_test, y_pred2)
print(acc1, acc2)
```
Python

```
0.9166666666666666 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) Showing the accuracy score of test data by n_estimators=10 and n_estimators=100, respectively.

```
clf_ada10 = AdaBoost(n_estimators=10)
#print(y_train)
clf_ada10.fit(x_train, y_train)
y_pred1 = clf_ada10.predict(valx_test)
#print(valy_test, valy_test.shape)
#print(y_pred1, y_pred1.shape)
acc1 = accuracy_score(valy_test, y_pred1)
print(acc1)
```

✓ 28.1s Python

```
0.94
```

```
clf_ada100 = AdaBoost(n_estimators=100)
#print(y_train)
clf_ada100.fit(x_train, y_train)
y_pred100 = clf_ada100.predict(valx_test)
acc100 = accuracy_score(valy_test, y_pred100)
print(acc100)
```

✓ 6m 47.5s Python Python

```
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) 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.

```
clf_10tree = RandomForest(n_estimators=10, max_features=np.sqrt(x_train.shape[1]))
clf_10tree.fit(x_train, y_train)
y_pred3 = clf_10tree.predict(valx_test)
acc3 = accuracy_score(valy_test, y_pred3)
print(acc3)
```

✓ 34.7s Python

```
0.93
```

```
clf_100tree = RandomForest(n_estimators=100, max_features=np.sqrt(x_train.shape[1]))
clf_100tree.fit(x_train, y_train)
y_pred4 = clf_100tree.predict(valx_test)
acc4 = accuracy_score(valy_test, y_pred4)
print(acc4)
```

✓ 6m 46.7s Python

```
0.9433333333333334
```

(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.

```
clf_random_features = RandomForest(n_estimators=10, max_features=np.sqrt(x_train.shape[1]))
clf_random_features.fit(x_train, y_train)
ranpred = clf_random_features.predict(valx_test)
clf_all_features = RandomForest(n_estimators=10, max_features=x_train.shape[1])
clf_all_features.fit(x_train, y_train)
allpred = clf_all_features.predict(valx_test)
accran = accuracy_score(valy_test, ranpred)
accall = accuracy_score(valy_test, allpred)
print(accran, accall)
```

✓ 2m 28.7s Python

```
0.9433333333333334 0.9566666666666667
```

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.

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.

Ans: Because if we didn't limit the depth of the tree, the decision tree can divide the training data to many very small pieces. For example, the size of data=100, if we grow a decision tree with 100 nodes, we can customized one node to fit one data, thus get a 100% accuracy. To sum, it's possible for a decision tree to reach a 100% accuracy in the "training" set, but it won't be preferable while applying on testing dataset.

On way to reduce overfitting decision tree is pre-pruning, such as adding MAX_DEPTH limit or MIN_SAMPLES limit. By doing this, we can avoid the tree to grow too completely with customization for the training data.

The second solution is post-pruning. We let the tree to grow completely and then pruning the tree base on ccp (cost complexity pruning) or other methods.

The third solution is bootstrapping. We randomly sample data for one tree and join every tree's result by voting to prevent overfitting.

2.

a. In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.

Ans: True. For misclassified attribute, we will multiply it by e^(-alpha).

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.

Ans: True. The later weak classifiers (in training order) need to classify examples that were wrong previously. We can assume those examples were more difficult to classify. While their weight is higher, the error tends to be higher.

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

Ans: False, if the data is not linear separable, the training error won't be zero no matter how many iterations it tries.

3. 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 (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 C1 and m points are assigned to C2. 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= -k=1Kpk log2 pk** and **Gini index** Gini=1-k=1Kpk2 **for the two trees**. Define pk to be the proportion of data points in region R assigned to class k, where k = 1, . . . , K.