NYCU Introduction to Machine Learning, Homework 3

109700046, 侯均頲

The screenshot and the figures we provided below are just examples. **The results below are not guaranteed to be correct.** Please make sure your answers are clear and readable, or no points will be given. Please also remember to convert it to a pdf file before submission. You should use English to answer the questions. After reading this paragraph, you can delete this paragraph.

Part. 1, Coding (50%):

For this coding assignment, you are required to implement the Decision Tree and Adaboost algorithms using only NumPy. After that, train your model on the provided dataset and evaluate the performance on the testing data.

(30%) Decision Tree

1. (5%) Compute the gini index and the entropy of the array [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1].

```
gini of [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1]: 0.4628099173553719
entropy of [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1]: 0.6554817739013927
```

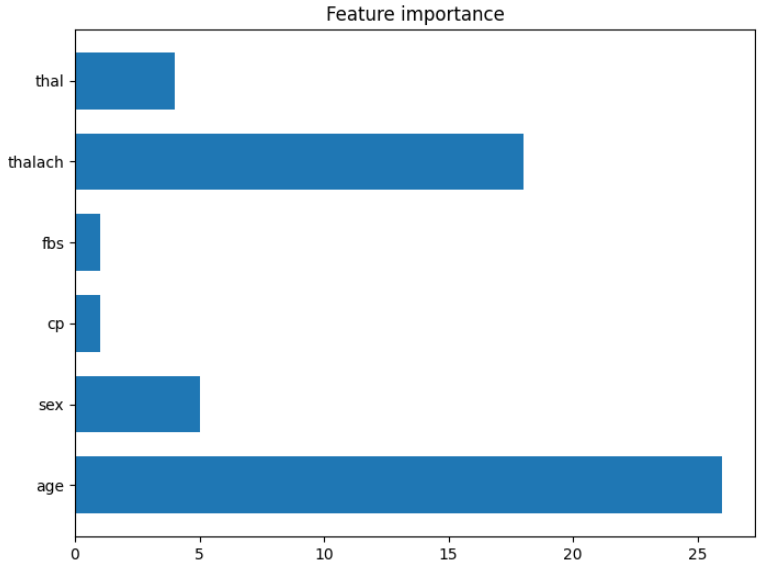
2. (10%) Show the accuracy score of the testing data using criterion="gini" and max_depth=7.

```
Accuracy (gini with max_depth=7): 0.7049180327868853
```

3. (10%) Show the accuracy score of the testing data using criterion="entropy" and max_depth=7.

```
Accuracy (entropy with max_depth=7): 0.7540983606557377
```

4. (5%) Train your model using criterion="gini", max_depth=15. Plot the feature importance of your decision tree model by simply counting the number of times each feature is used to split the data.



(20%) Adaboost

Requirements:

- Implement the Adaboost algorithm by using the decision tree classifier (max_depth=1) you just implemented as the weak classifier.
- The Adaboost model should include the following two arguments:
- criterion: The function to measure the quality of a split of the data. Your model should sup port "gini" and "entropy".
- n_estimators: The total number of weak classifiers.

Tips:

- You can set any random seed to make your result reproducible.

Criteria:

5. (20%) Tune the arguments of AdaBoost to achieve higher accuracy than your Decision Tre es.

```
Part 2: AdaBoost
Accuracy: 0.7704918032786885
```

Part. 2, Questions (50%):

1. (10%) True or False. If your answer is false, please explain.
    1. (5%) In an iteration of AdaBoost, the weights of misclassified examples are increa sed by adding the same additive factor to emphasize their importance in subsequen t iterations.

False, not adding the same additive factor but a multiplicative factor related to caculated alpha.

    2. (5%) AdaBoost can use various classification methods as its weak classifiers, such as linear classifiers, decision trees, etc.

True

2. (10%) How does the number of weak classifiers in AdaBoost influence the model's performance? Please discuss the potential impact on overfitting, underfitting, computational cost, memory for saving the model, and other relevant factors when the number of weak classifiers is too small or too large.

when number of weak classifiers is too small, it will cause underfitting because we are splitting data to casually with just few linear lines, while the computational cost is relitively small, memory for saving the model will be also small. when number of weak classifiers is too large, it will cause overfitting beacause we use too much line segment to split train data, and it will be overfitting when we are testing, the computational cost will be large also casue we use large n_estimators and memory for saving the model will be large because we save a model with every n _estimators.To find a balanced n_estimators which will not cause overfitting or underfitting and have acceptable memory usage and computational cost is important.

3. (15%) A student claims to have a brilliant idea to make random forests more powerful: since random forests prefer trees which are diverse, i.e., not strongly correlated, the student proposes setting m = 1, where m is the number of random features used in each node of each decision tree. The student claims that this will improve accuracy while reducing variance. Do you agree with the student's claims? Clearly explain your answer.

When m = 1, it will lead to less overfitting and have chance reducing the variance of of the model, however, this may cause increase of bias on some features, which may lead to Not capturing the real relationships present in the data, and it may reduce accuracy. And using only one feature may over simplify decision boundary, which limits the complexity of a tree, this may also impact the predtion power of a tree and affect accuracy, and sometimes some features should considered together but not seperately to get the meaning of it, whats more, when m = 1 you may sometimes capture a noise or a useless feature, than that node will be totally useless . m = 1 can create divderity but increase bias, in case like this homework may be useful, but in more complicated case even when features are somehow related to each others, accuracy may not improve.

4. (15%) The formula on the left is the forward process of a standard neural network while the formula on the right is the forward process of a modified model with a specific technique.

   1. (5%) According to the two formulas, describe what is the main difference between the two models and what is the technique applied to the model on the right side.

by applying Bernoulli(p) we generate a mask rl, randomly make only some of y is selected or considered base on probability p on the right, on the left we consider all y in the forward pass, and the right part we encourage the network to learn more because we apply the dropout technique, preventing neurons from being too dependent on specific input features, but left part won’ t do that because there is no such noise.

   2. (10%) This technique was used to deal with overfitting and has many different explanations; according to what you learned from the lecture, try to explain it with respect to the ensemble method.

dropout technique can be seen as a form of ensemble learning in a single neural network, in adaboost we conbine weak learners to form a strong learner, which encourages our model to learn more in many aspect, and the dropout also encourage the network to learn more and more robust by adding noise; decision tree try to avoid overfitting by prunning, and drop out prevent overfitting by ran

domly dropping units; Bagging construct multiple models train on different subsets, drop out can a lso be seen as training different subset in each iterations by randomly dropping out. Random Fores ts tend to be diverse by creating trees, dropout introduces diversity by randomly zeroing out units, make each layer learn more robust features. Overall, they have pretty same spirit.

$$z^{(l+1)} = w^{(l+1)} y^l + b^{(l+1)}$$

$$y^{(l+1)} = f(z^{(l+1)})$$

$$r^l = Bernoulli(p)$$

$$\tilde{y}^l = r^l y^l$$

$$z^{(l+1)} = w^{(l+1)} \tilde{y}^l + b^{(l+1)}$$

$$y^{(l+1)} = f(z^{(l+1)})$$