NYCU Introduction to Machine Learning, Homework 3

110612117 張仲瑜

Part. 1, Coding (50%):

(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]: 0.9456603046006401
```

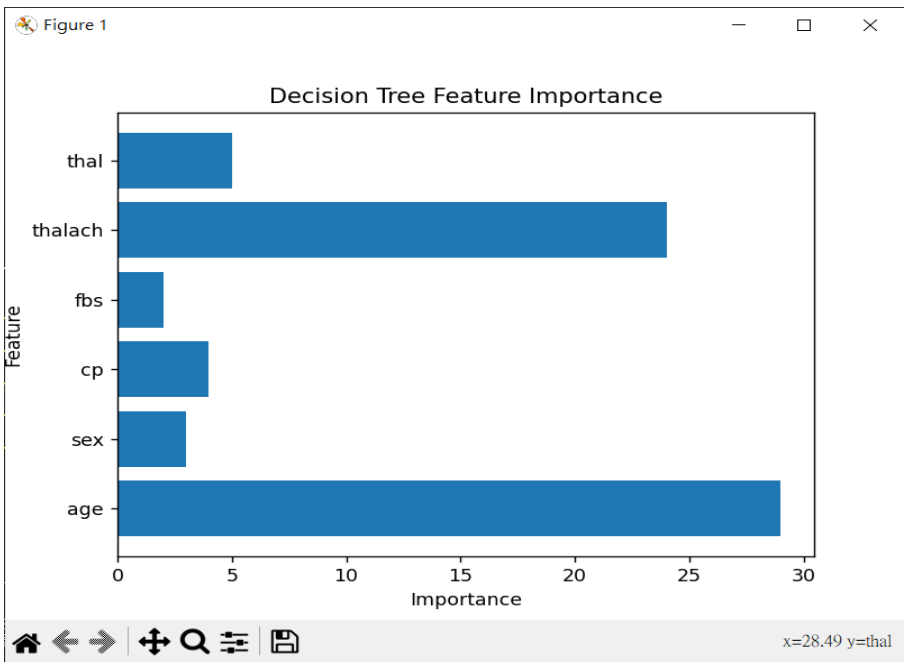
2. **(10%) Show the accuracy score of the testing data using criterion="gini" and max_depth=7. Your accuracy score should be higher than 0.7.**

```
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. Your accuracy score should be higher than 0.7.**

```
Accuracy (entropy with max_depth=7): 0.7213114754098361
```

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

1. Tune the arguments of AdaBoost to achieve higher accuracy than your Decision Trees.

```
ada = AdaBoost(criterion='gini', n_estimators=100)
```

```
if errorRate >= 1.0:
    alpha = 0.0
elif errorRate <= 0.25:
    alpha = 200
else:
    alpha = 0.5 * np.log((1 - errorRate) / errorRate)
```

```
Part 2: AdaBoost
Accuracy: 0.8032786885245902
```

Part. 2, Questions (50%):

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

   Ans:

   False, the weights of misclassified examples are not increased by the same additive factor. It increases by the formula by timing a different exponential.

   Weight *= exp(-alpha * ThatDataMisclassifiedOrNot)

   (Then normalized)

   When the data is correctly classified(the latter = 1), the whole exponential <1 which lowers the original weight, while it higher the original weight when the data is misclassified by timing a exponential > 1.

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

Ans:

True, Multiple classification methods can be utilized as weak classifiers.

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.

a. Impact on Overfitting and Underfitting

Too Few weak classifiers may leads to underfitting since the model lacks the complexity of to capture the underlying patterns in the data, while too many weak classifiers may leads to overfitting since the model becomes too tailored to the training data and fails to predict unseen data

b. Impact on Computational cost and memory usage

The larger the number of weak classifiers is, the higher the computational cost and memory usage are due to iteration of boosting process when training the weak classifiers, while the smaller the number of weak classifiers is, the lower the computational cost and memory usage are.

c. Impact on training time

The larger the number of weak classifiers is, it normally needs more training time to finish all the iteration when training weak classifiers.

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.

Ans

Setting the numbers of random features used in each node of each decision tree will cause the problem below

a. It lowers the diversity of the tree since only one aspect of the data has been considered each time, which can leads to underfitting if the data are complicated.

b. Overfitting may also happen when the tree keep selecting the same attribute by chance.

In conclusion, the robustness of the model will dramatically decrease and leads to poor performance.

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.

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

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

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

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

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

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

a. (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.

Ans:

The main difference between them is that the right formula includes a Bernoulli function when forwarding. It is a technique called dropping since when Bernoulli function output 0, the neuron will be discarded, which can be used to prevent overfitting and increasing the model' s robustness, while when the Bernoulli function output 1, it is the same as the formula in the left side.

b. (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.

Ans

i. Dropout can be viewed as a form of model averaging within a single neuron network. During each training iteration, a random subset of neurons is dropped out, effectively training a different subnetwork. This process can be seen as training and averaging the predictions of multiple subnetworks, just like an ensemble method that predict output by training subnetworks. Eg: weak classifier, bagging etc.

ii. Dropout increases the diversity and reduces the chance of overfitting by randomly deactivating neuron, which can prevent the network from memorizing specific patterns in training data. This reaches the same goal as the ensemble methods was pursued by taking different characteristics in training iterations.