NYCU Intro. to ML
HW3 Report

110550093, 蔡師睿

Part I. Coding

Decision Tree

1. 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
```

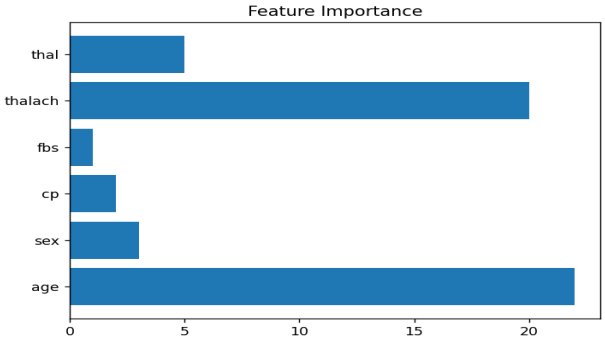
2. Show the accuracy score of the testing data using criterion=’gini’ and max_depth=7.

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

3. Show the accuracy score of the testing data using criterion=’entropy’ and max_depth=7.

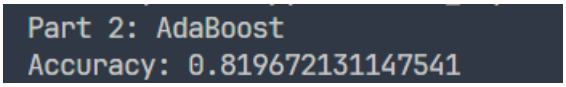
```
Accuracy (entropy with max_depth=7): 0.7213114754098361
```

4. 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.



Adaboost

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



Part II. Questions

1. True of False. If your answer is false, please explain.

a. 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
Explanation: After each of iteration of AdaBoost, the weights are adjusted by multiplying them with a factor, which may either increase or decrease them. Besides, the factors aren't the same in each iteration. It depends on the error rate predicted by the weak classifier.

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

Ans: True

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

- **Overfitting and Underfitting**
  The model may underfit the training data if the number of classifiers is too low. This is because the small number of weak classifiers are too simple to adequately represent the data, making them incapable of capturing essential features and making accuracy predictions. On the other hand, if the number of weak classifiers is too high, there is a risk of overfitting. The model becomes too specialized to the training set as a large amount of weak classifiers, which causes the poor performance in testing set.

- **Computational Cost**
  The more weak classifiers there are, the longer the training time the model takes, and so does the predicting time. This is because the model needs to run through the entire set of weak classifiers to compute the predicted labels. Consequently, the entire process of fitting and predicting with a large AdaBoost model can become computationally expensive.

- **Memory for Saving the Model**
  The model needs to store the information about each weak classifier, inclusive of its selected feature, threshold, and depth; thus, the memory requirements grows with the increase in the number of weak classifiers.

- **Other Relevant Factors**
  The number of weak classifiers in AdaBoost may also influence the model's generalizability. Ensemble models aim to generalize well to unseen data; however, by adding too many weak classifiers may not significantly improve the generalizability of the model due to complexity and overfitting.
  Another factor is robustness. Robustness helps the model reduce sensitivity to noise or outliers and adapt to various patterns in data. Though AdaBoost enhance the robustness by combining multiple weak classifiers, sometimes it will become too robustness to the training set and cause overfitting.
  To achieve the benefits of the AdaBoost, a balanced number of weak classifiers contributes to a more robust and generalizable model.

3. 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?

In a random forest, we often set m to a number between 1 and the number of features. Although increasing the diversity is benefit to a random forest, the diversity still have to be balanced. In this case, by setting m to 1 may lead to weak learners that are too simplistic and fail to capture the essential patterns in the data, because each split in each tree will consider only one randomly selected features. Due to the above reasons, the student's proposal may not lead to the expected improvements in performance. In fact, it's more likely to have the opposite effect.
In summary, the student puts excessive emphasis on allowing for diversity but overlooks to maintain the ability to capture relevant information form the data.

4. 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. According to the two formulas, describe what is the main difference between the two models and what is technique applied to the model on the right side.

The right model introduces the term r^l, which is the Bernoulli distribution. This technique is also known as Dropout in neural network. Therefore, the right model represents the dropout neural network. Dropout layer will randomly set the output of some neurons to zero during training, which is useful to deal with overfitting in neural network.

b. 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.

In ensemble methods, multiple models are trained independently, and their predictions are combined to make the final decision. The idea is that by combining different model with varying sets of features, weights, or subset of the data, the ensemble model can generalize better to unseen data.
Dropout introduces a similar concept when only training within a single neural network. During training, the neurons will randomly be dropped out, and it will effectively creating many different subnetworks from iteration to iteration. This process can be thought of as training an ensemble of neural network.
In summary, dropout in neural networks can be viewed as a form of implicit ensemble learning, where diverse subnetworks are trained within a single model to improve generalization and combat overfitting.