NYCU Introduction to Machine Learning, Homework 1

109550085, 陳欣妤

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%):

(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].

```
Part 1: Decision Tree
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
```

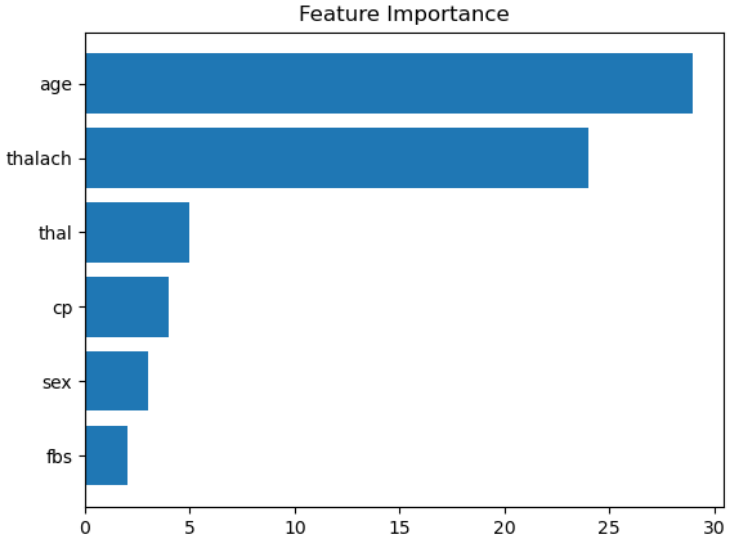
2. (10%) Show the accuracy score of the testing data using criterion="gini" and max_depth=7. Your accuracy score should be higher than 0.7.

```
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.

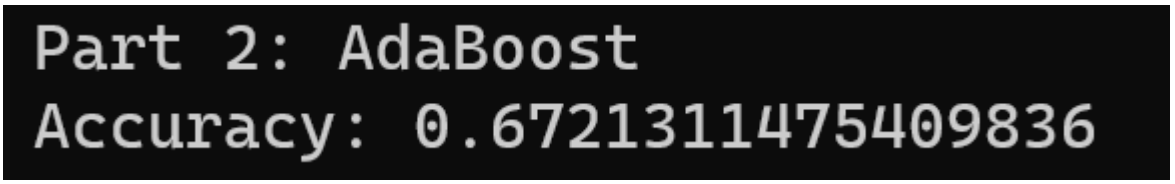
```
Accuracy (entropy with max_depth=7): 0.7213114754098361
```

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

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



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

    a. False: In AdaBoost, the weights of misclassified examples are increased, but the emphasis on these examples decreases as the algorithm progresses. The weights are updated to focus on the misclassified examples so that subsequent weak learners pay more attention to them.

    b. True: AdaBoost is a flexible ensemble method that can use various weak classifiers, including linear classifiers, decision trees, or any other classification algorithm. The key requirement is that the weak classifiers should perform slightly better than random chance.

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. **Underfitting and Overfitting**:
    - **Too Few** Weak Classifiers: If the number of weak classifiers is too small, the model may underfit the training data. It may fail to capture the complexity of the underlying relationships, leading to poor generalization to unseen data.
    - **Too Many** Weak Classifiers: On the other hand, if the number of weak classifiers is too large, there is a risk of overfitting. The model might memorize the training data, including noise, and perform poorly on new, unseen examples.

b. **Computational Cost**:
   - **Too Few** Weak Classifiers: Training a model with too few weak classifiers may be computationally less expensive, but it might sacrifice model performance.
   - **Too Many** Weak Classifiers: As the number of weak classifiers increases, the computational cost of training and prediction also increases.

c. **Memory Requirements**:
   - **Too Few** Weak Classifiers: A model with fewer weak classifiers will generally require less memory to store the model parameters.
   - **Too Many** Weak Classifiers: Increasing the number of weak classifiers increases the size of the model, leading to higher memory requirements for storing the model.

d. **Generalization**:
   - **Sweet Spot**: There is typically a "sweet spot" in terms of the number of weak classifiers that results in the best generalization performance. It is often determined through cross-validation or other model selection techniques.

e. **Training Time**:
   - **Too Few** Weak Classifiers: Training a model with too few weak classifiers might converge quickly, but it may not have learned a sufficiently expressive representation of the data.
   - **Too Many** Weak Classifiers: Training time increases as the number of weak classifiers grows, and it may lead to diminishing returns in terms of performance improvement.

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.

   No, it is not likely to improve accuracy and may not effectively reduce variance. In fact, it may lead to a decrease in the performance of the random forest ensemble. If $m = 1$, it means that each node in every decision tree is considering only one random feature to make a split decision. This severely limits the diversity among trees because they will tend to make decisions based on the same or very similar features at each node. Limiting each node to one random feature may lead to oversimplified trees that are not able to capture the complexity of the underlying relationships in the data. This can result in a decrease in accuracy, especially if the dataset requires considering multiple features for meaningful splits.

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.
   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.
   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.

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

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

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

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

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

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

a. On the right side, the standard neural network forward process is shown, where $z^{(l+1)}$ is computed directly as a linear combination of the output y^{l} from the previous layer, followed by the activation function f.
   On the left side, a stochastic element is introduced. First, a binary mask r^{l} is sampled from a Bernoulli distribution with probability p. This mask is then applied element-wise to the output y^{l}, effectively "dropping out" some units by zeroing them out with probability $1-p$. The modified output y^{-l} is then used to compute $z^{(l+1)}$ and subsequently $y^{(l+1)}$.
   The technique applied on the left side is dropout. Dropout is a regularization technique used to prevent overfitting in neural networks.

b. Ensemble methods are known for their ability to reduce overfitting by combining multiple models that might individually overfit the training data. Similarly, dropout introduces a form of regularization by preventing the network from relying too heavily on specific neurons or complex co-adaptations. The ensemble effect in dropout contributes to better generalization on unseen data. The network learns a more robust and generalized representation of the underlying patterns in the data. Dropout can be seen as a form of bagging. In bagging, multiple models are trained on random subsets of the data. Dropout achieves a similar effect by training on random subsets of neurons.