NYCU Pattern Recognition, Homework 3

312551168,杜祐陞

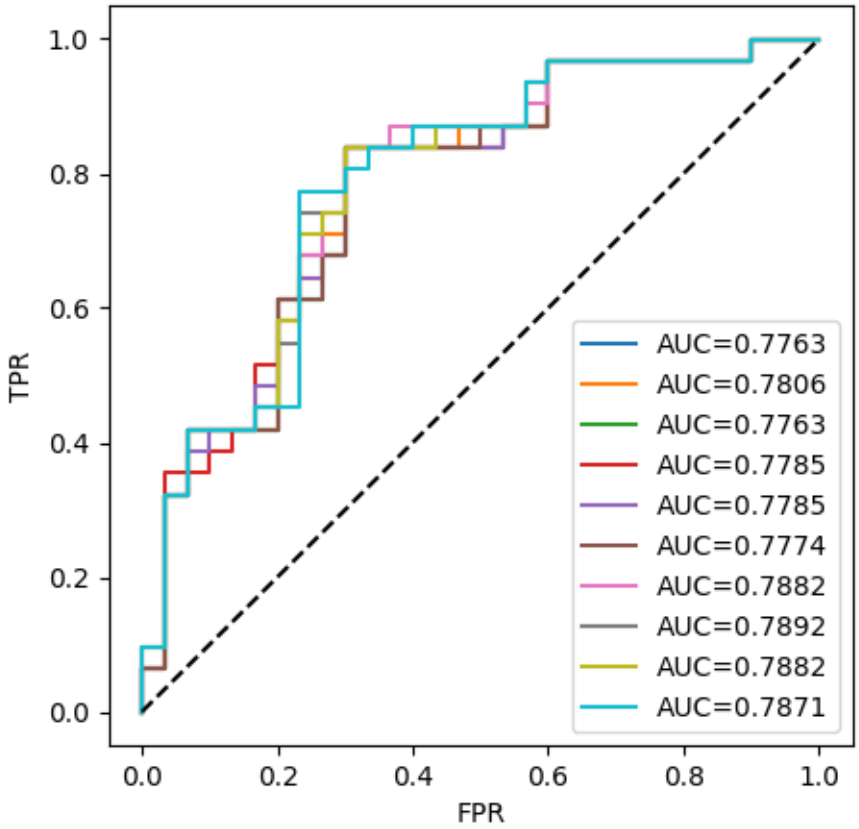
Part. 1, Coding (60%):

(20%) Adaboost

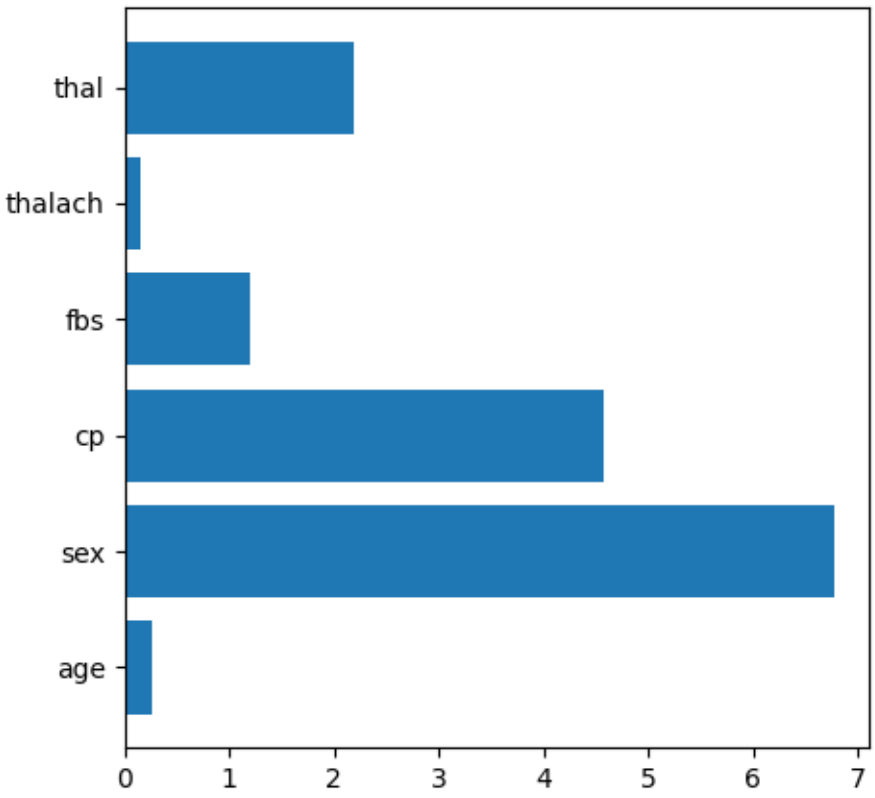
1. (10%) Show your accuracy of the testing data (n_estimators = 10)

```
2024-05-13 19:07:43.021 | INFO     | __main__:main:37 - AdaBoost - Accuracy: 0.7705
```

2. (5%) Plot the AUC curves of each weak classifier.



3. (5%) Plot the feature importance of the AdaBoost method. Also, you should snapshot the i mplementation to calculate the feature importance.



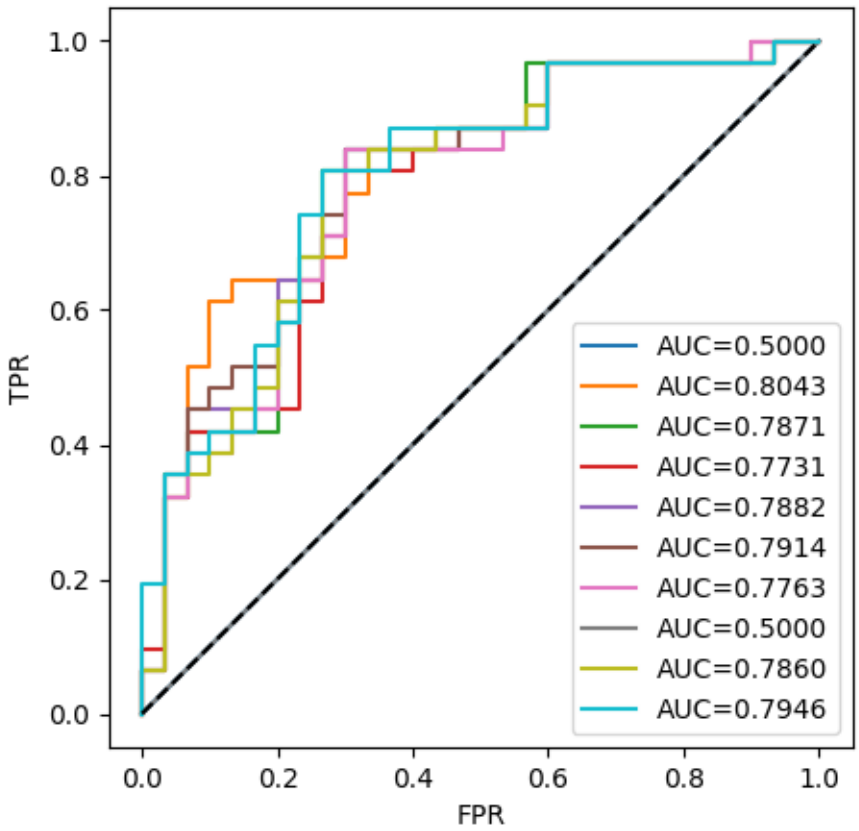

```python
def compute_feature_importance(self) -> t.Sequence[float]:
    """Implement your code here"""
    feature_importance = np.zeros(6)
    # print(feature_importance)
    for alpha, model in zip(self.alphas, self.learners):
        feature_importance += (model.layers[0].weight.abs() * alpha).detach().numpy().squeeze()
    return feature_importance
```

(20%) Bagging

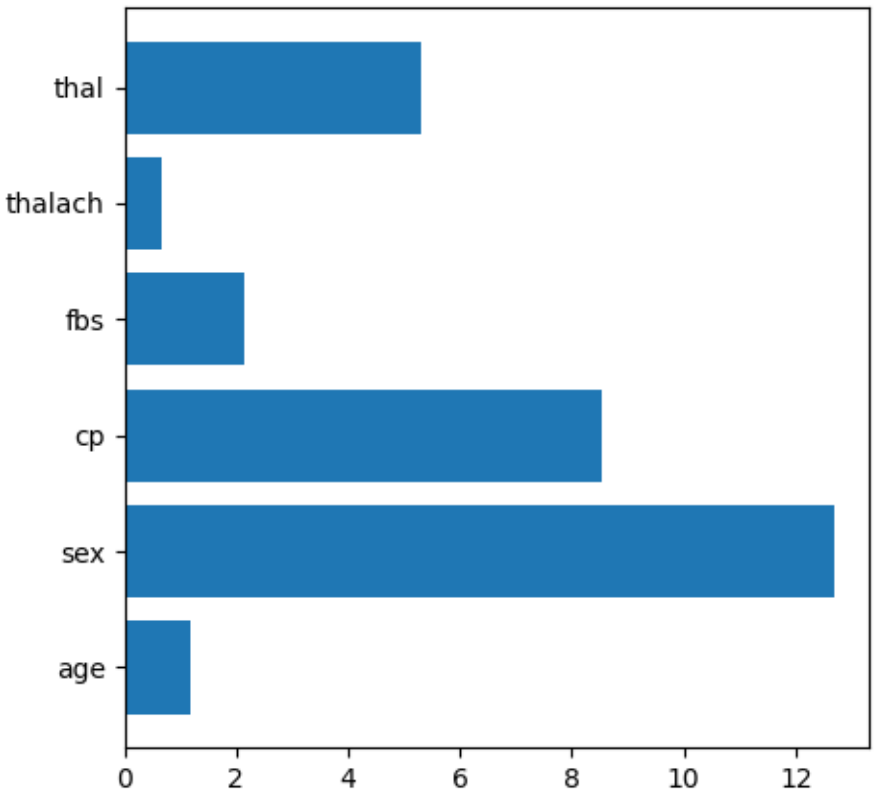
4. (10%) Show your accuracy of the testing data with 10 estimators. (n_estimators=10)

```
2024-05-13 19:07:45.601 | INFO     | __main__:main:64 - Bagging - Accuracy: 0.7705
```

5. (5%) Plot the AUC curves of each weak classifier.



6. (5%) Plot the feature importance of the Bagging method. Also, you should snapshot the im plementation to calculate the feature importance.




```python
def compute_feature_importance(self) -> t.Sequence[float]:
    """Implement your code here"""
    feature_importance = np.zeros(6)
    # print(feature_importance)
    for model in self.learners:
        feature_importance += (model.layers[0].weight.abs()).detach().numpy().squeeze()
    return feature_importance
```

(15%) Decision Tree

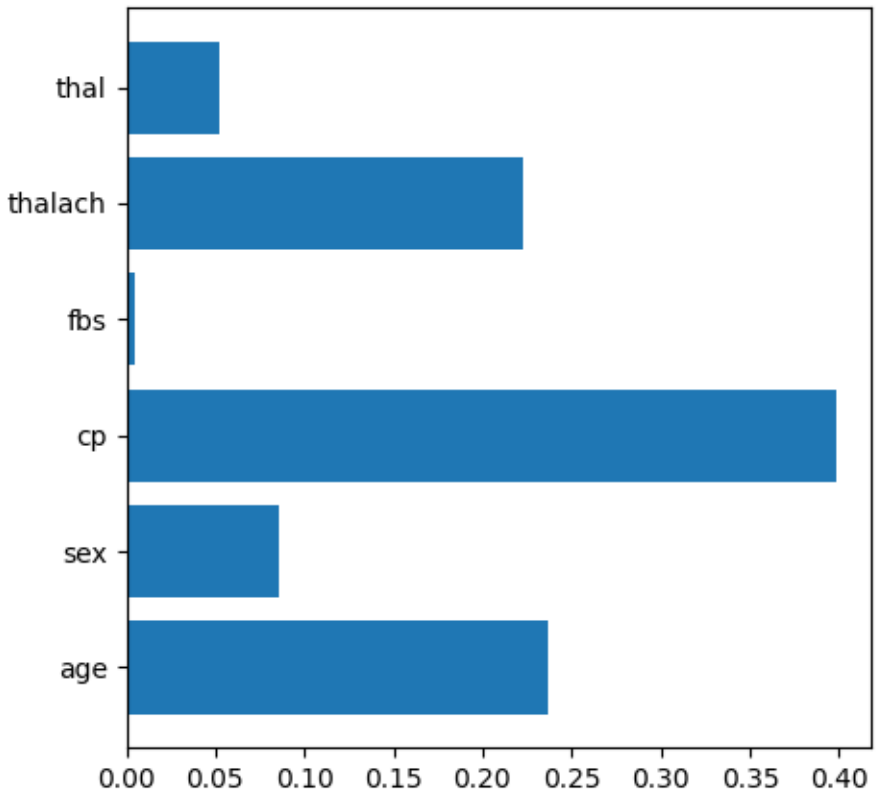
7. (5%) Compute the gini index and the entropy of the array [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1].

```
2024-05-14 16:11:49.262 | INFO     | __main__:main:98 - array [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1] - Gini Index: 0.4628
2024-05-14 16:11:49.262 | INFO     | __main__:main:100 - array [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1] - Entropy: 0.9457
```

8. (5%) Show your accuracy of the testing data with a max-depth = 7

```
2024-05-13 19:07:45.830 | INFO     | __main__:main:87 - DecisionTree - Accuracy: 0.7213
```

9. (5%) Plot the feature importance of the decision tree.



(5%) Code Linting

10. Show the snapshot of the flake8 linting result.

```
(pattern_hw3) (base) gpl@gpl-System-Product-Name:/media/4TB_HDD/pattern_recognition/hw3$ flake8 main.py
(pattern_hw3) (base) gpl@gpl-System-Product-Name:/media/4TB_HDD/pattern_recognition/hw3$
```

Part. 2, Questions (40%):

1. (10%) We have three distinct binary classifiers, and our goal is to leverage them in creating an ensemble classifier through the majority voting strategy to make decisions.
Assuming each individual binary classifier operates independently of the others with an accuracy of 60%, what would be the accuracy of the ensemble classifier?

We assume that the three classifiers are independent. Since we use the majority voting strategy, the output is correct if at least two classifiers predict right. Therefore, we have to consider two cases: the first case is all the classifiers predict correctly, and the second is two of them predict correctly.

First case:

$$0.6 \times 0.6 \times 0.6 = 0.216$$

Second case:

$$3 \times 0.6 \times 0.6 \times 0.4 = 0.432$$

So, sum up the probability of the two cases, we get $0.216 + 0.432 = 0.648$ as the accuracy of the ensemble classifier.

2. (15%) For the decision tree algorithm, we can use the "pruning" technique to avoid overfitting. Does the random forest algorithm also need pruning?

In decision tree, we use pruning to avoid overfitting since decision tree tend to learn noise of the data if it is very deep or full depth. However, in a random forest, we already use bagging, which makes each tree train on a different sub-dataset and use random features. These two methods make each tree strong but not correlated to others. Therefore, the noise or the error will be average by majority voting. That's the reason why we usually do not use pruning to avoid overfitting in random forest.

3. (15%) Activation functions are core components of neural networks. They need to be differentiable to ensure backpropagation works correctly. Please calculate the derivatives of the following commonly used activation functions.

(For questions 1. and 2., consider the cases where $x > 0$ and $x \leq 0$)

1. f(x) = relu(x),	df(x)/dx = ?
2. f(x) = leaky_relu(x) with negative_slope=0.01,	df(x)/dx = ?
3. f(x) = sigmoid(x),	df(x)/dx = ?
4. f(x) = silu(x),	df(x)/dx = ?
5. f(x) = tanh(x),	df(x)/dx = ?

1. $f(x) = \text{relu}(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases}$

$$df(x)/dx = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$$

2. $f(x) = \text{leaky\_relu}(x) \text{ with negative\_slope} = 0.01 = \begin{cases} 0.01x, & x \leq 0 \\ x, & x > 0 \end{cases}$

$$df(x)/dx = \begin{cases} 0.01, & x \leq 0 \\ 1, & x > 0 \end{cases}$$

3. $f(x) = \text{sigmoid}(x) = \frac{1}{1+e^{-x}}$

$$df(x)/dx = \frac{e^{-x}}{(1+e^{-x})^2}$$

4. $f(x) = \text{silu}(x) = \frac{x}{1+e^{-x}}$

$$df(x)/dx = \frac{1 + e^{-x} + xe^{-x}}{(1 + e^{-x})^2}$$

5. $f(x) = \tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$

$$\frac{df(x)}{dx} = \frac{2e^{2x}(e^{2x} + 1) - (e^{2x} - 1)2e^{2x}}{(e^{2x} + 1)^2} = \frac{4e^{2x}}{(e^{2x} + 1)^2}$$