NYCU Introduction to Machine Learning, Homework 3

Deadline: Nov. 28, 23:59

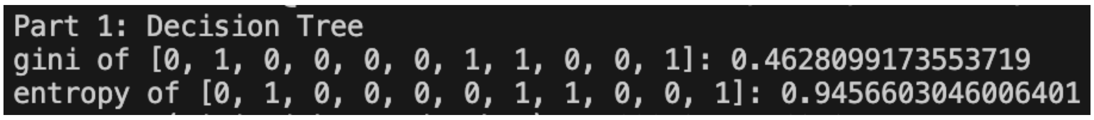
Part. 1, Coding (50%): 109612019 林伯偉

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

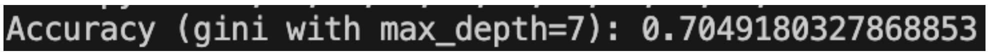
Criteria:

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



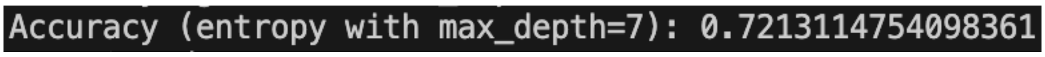
```
Part 1: Decision Tree
gini of [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1]: 0.4628099173553719
entropy of [0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1]: 0.9456603046006401
```

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.



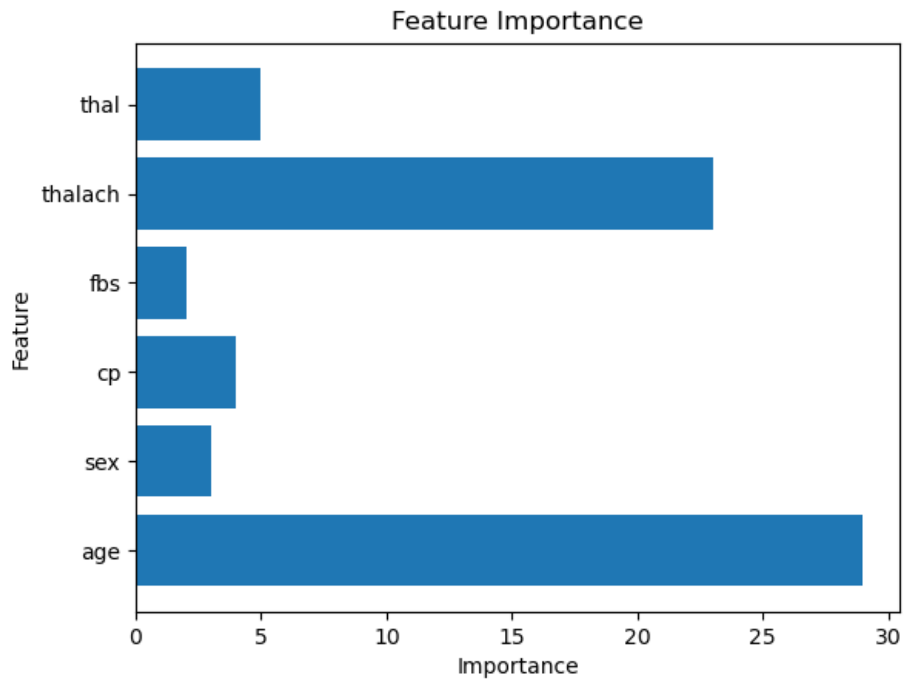
```
Accuracy (gini with max_depth=7): 0.7049180327868853
```

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
```

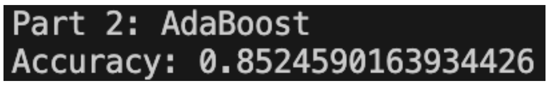
4. (5%) Train your model using criterion="gini", max_depth=15. Plot the feature importance of your decision tree model by simply counting the number of times each feature is used to split the data.



(20%) Adaboost

Criteria:

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



```
Part 2: AdaBoost
Accuracy: 0.8524590163934426
```

Points	**Testing Accuracy**
20 points	0.8 <= acc
15 points	0.78 <= acc < 0.8
10 points	0.76 <= acc < 0.78
5 points	**0.74 <= acc < 0.76**
0 points	**acc < 0.74**

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.

      False. In an iteration of AdaBoost, the weights of misclassified examples are increased, but the emphasis on the importance of these examples is achieved by increasing their weights “multiplicatively”, not by adding the same additive factor. The idea is to give more weight to the misclassified examples so that the subsequent weak learners focus more on correctly classifying these examples in the next iteration. The weight update is typically done using an exponential function to increase the influence of misclassified examples.

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

      True. AdaBoost is a versatile ensemble learning algorithm that can be used with various weak classifiers, which includes linear classifiers, decision trees, and even more complex models. The key requirement for a weak classifier in AdaBoost is that it performs slightly better than random chance, as the algorithm aims to improve overall accuracy by combining multiple weak classifiers. The flexibility to use different types of weak classifiers makes AdaBoost applicable to a wide range of machine learning tasks.

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.

- **Not Enough Weak Classifiers**
  - **Underfitting:**
    If the number of weak classifiers is too small, the model may underfit the data. It might not capture the complexity of the underlying patterns in the training data.
  - **Poor Generalization**
    The model may struggle to generalize well to unseen data, as it lacks the capacity to learn intricate relationships in the training set.
  - **Bias:**
    The model may exhibit high bias, leading to systematic errors and poor performance on both the training and test datasets.

- **Too Many Weak Classifiers**
  - **Overfitting:**
    Increasing the number of weak classifiers can lead to overfitting, where the model starts to memorize noise in the training data rather than learning the true underlying patterns.
  - **Computational Cost:**
    As the number of weak classifiers increases, the computational cost of training also rises. Training a large number of classifiers can be computationally expensive and time-consuming.
  - **Memory Usage:**
    Storing a large number of weak classifiers requires more memory. This could become an issue, especially in resource-constrained environments.
  - **Increased Complexity:**
    A very large number of weak classifiers might introduce unnecessary complexity to the model, making it harder to interpret and debug.

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.

The student set m=1, meaning to use only one random feature at each node of each decision tree in a Random Forest, may not necessarily lead to improved accuracy and reduced variance. The value of m in a Random Forest represents the number of features randomly sampled at each node when splitting a tree.

Disadvantages of setting m = 1

- **Lack of diversity:**
  The strength of Random Forests lies in the diversity of individual trees. By using only one random feature at each node, trees become more similar, and the ensemble loses the benefits of incorporating diverse perspectives. This can lead to a reduction in the overall performance of the ensemble.

- **Overfitting:**
  Using only one feature at each node increases the risk of overfitting. Trees might become too specialized in capturing noise or outliers in the training data, leading to poor generalization on unseen data.

- **Reduced Robustness:**
  Random Forests are robust to outliers and noisy features because they aggregate the predictions of multiple trees. By using only one feature, the ensemble becomes more sensitive to outliers and noisy data.

- **Variance Increase:**
  The primary purpose of random feature selection is to decorrelate trees and reduce variance. When using only one feature, the correlation between trees may increase, potentially leading to higher overall variance.

In conclusion, while the student's idea may have been motivated by the desire for diversity, setting m=1 is not recommended. It is crucial to strike a balance between diversity and correlation among trees in a Random Forest to achieve the best trade-off between bias and variance and to ensure robust generalization to unseen data

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.

      The two formula illustrates the mathematical expressions of a Standard Neural Network and the differences introduced by incorporating the technique “Dropout”. It is evident that only the red portion is added, indicating that each neuron generates a 0 with a probability of p from a Bernoulli Distribution and a 1 with a probability of (1-p). Suppose the layer before the Dropout layer is a Dense/Linear Layer with a total of 1000 neurons. If we set the Dropout probability (p) to 0.5, each neuron will have a 50% chance of being deactivated. In other words, approximately 500 neurons out of the 1000 will be deactivated (with their output set to 0).

   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.

      In ensemble method, we train multiple models and consider their predictions together for the final outcome. In Ensemble method, the concept of averaging or majority voting is used to enhance the overall performance of the models. The Dropout technique is akin to an ensemble concept: in each iteration, we "turn off" certain neurons, as if removing them from the Neural Network, creating a "different structure" for the network in each iteration.

      Throughout the training process, it appears as if we're training a single model, but in reality, it integrates various structures. Hence, the output of the Neural Network is not just the output of one network but a synthesis of outputs from "multiple" Neural Networks.

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

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