NYCU Introduction to Machine Learning, Homework 3

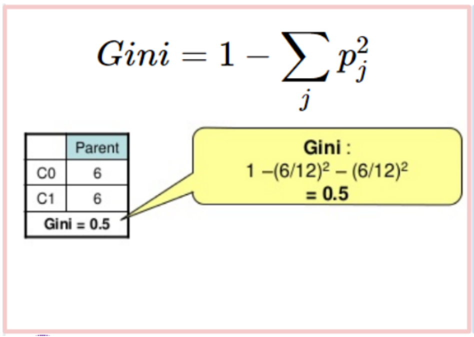
Deadline: Nov. 15, 23:59

Part. 1, Coding (80%):

In this coding assignment, you need to implement the Decision Tree, AdaBoost and Random Forest algorithm by using only NumPy, then train your implemented model by the provided dataset and test the performance with testing data. Find the sample code and data on the GitHub page
https://github.com/NCTU-VRDL/CS_CS20024/tree/main/HW3

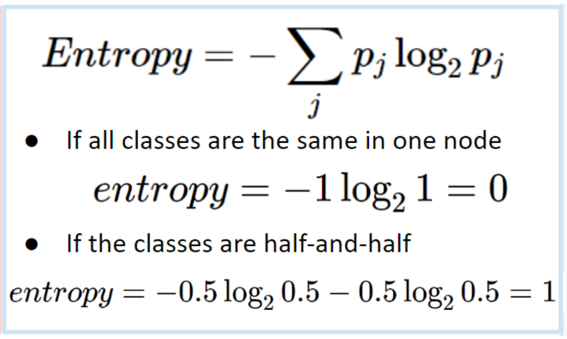
Please note that only NumPy can be used to implement your model, you will get no points by simply calling sklearn.tree.DecsionTreeClassifier.

1. (5%) Gini Index or Entropy is often used for measuring the "best" splitting of the data. Please compute the Entropy and Gini Index of this array `np.array([1,2,1,1,1,1,2,2,1,1,2])` by the formula below. (More details on page 5 of the hw3 slides, 1 and 2 represent class1 and class 2, respectively)



$$Gini = 1 - \sum_j p_j^2$$

	Parent
C0	6
C1	6
Gini = 0.5	



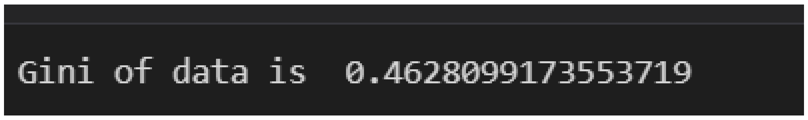
$$Entropy = -\sum_j p_j \log_2 p_j$$

- If all classes are the same in one node

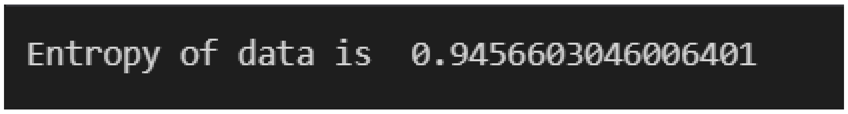
$$entropy = -1\log_2 1 = 0$$

- If the classes are half-and-half

$$entropy = -0.5\log_2 0.5 - 0.5\log_2 0.5 = 1$$

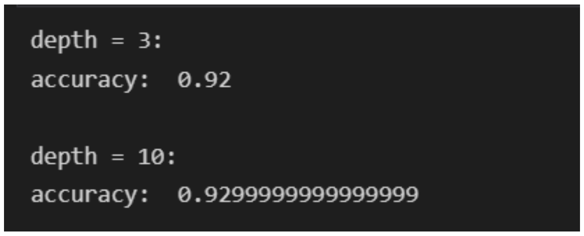


```
Gini of data is  0.4628099173553719
```



```
Entropy of data is  0.9456603046006401
```

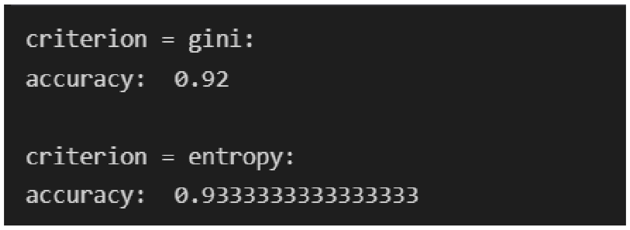
2. (10%) Implement the Decision Tree algorithm (CART, Classification and Regression Trees) and train the model by the given arguments, and print the accuracy score on the test data. You should implement **two arguments** for the Decision Tree algorithm, 1) **Criterion**: The function to measure the quality of a split. Your model should support "gini" for the Gini impurity and "entropy" for the information gain.
2) **Max_depth**: The maximum depth of the tree. If Max_depth=None, then nodes are expanded until all leaves are pure. Max_depth=1 equals split data once
2.1. Using Criterion='gini', showing the accuracy score of test data by Max_depth=3 and Max_depth=10, respectively.



```
depth = 3:
accuracy:  0.92

depth = 10:
accuracy:  0.9299999999999999
```

2.2. Using Max_depth=3, showing the accuracy score of test data by Criterion='gini' and Criterion='entropy', respectively.



```
criterion = gini:
accuracy:  0.92

criterion = entropy:
accuracy:  0.9333333333333333
```

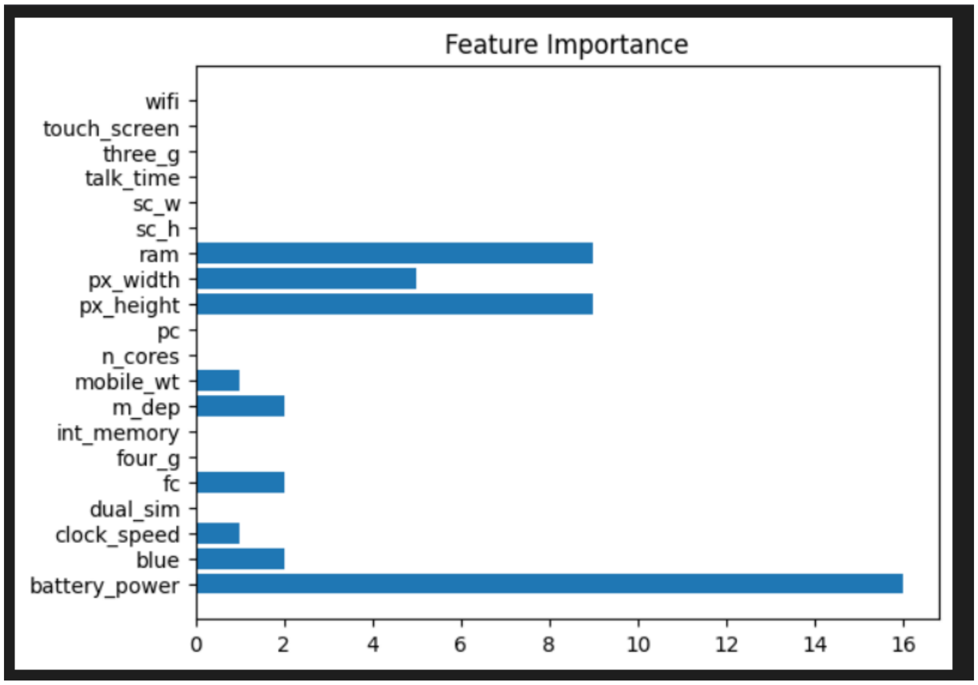
Note: Your decision tree scores should be over ***0.9****. It may suffer from overfitting, if so, you can tune the hyperparameter such as `max_depth`*

Note: You should get the same results when re-building the model with the same arguments, ***no need to prune the trees***

Note: You can find the best split threshold by both methods. First one: 1) Try N-1 threshold values, where the i-th threshold is the average of the i-th and (i+1)-th sorted values. Second one: Use the unique sorted value of the feature as the threshold to split

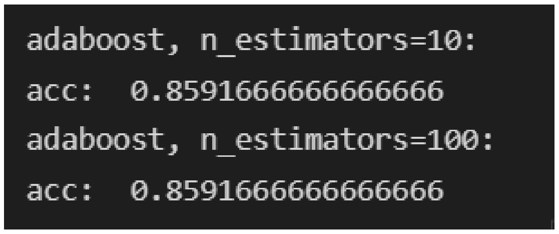
Hint: You can use the recursive method to build the nodes

3. (5%) Plot the feature importance of your Decision Tree model. You can use the model from Question 2.1, max_depth=10. (You can use simply counting to get the feature importance instead of the formula in the reference, more details on the sample code. **Matplotlib** is allowed to be used)



4. (15%) Implement the AdaBoost algorithm by using the CART you just implemented from question 2. You should implement **one argument** for the AdaBoost.
   1) **N_estimators**: The number of trees in the forest.

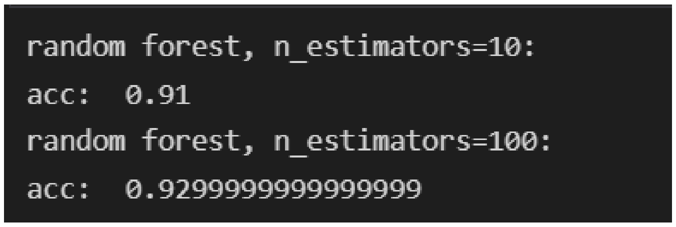
   4.1. Showing the accuracy score of test data by n_estimators=10 and n_estimators=100, respectively.



```
adaboost, n_estimators=10:
acc:  0.8591666666666666
adaboost, n_estimators=100:
acc:  0.8591666666666666
```

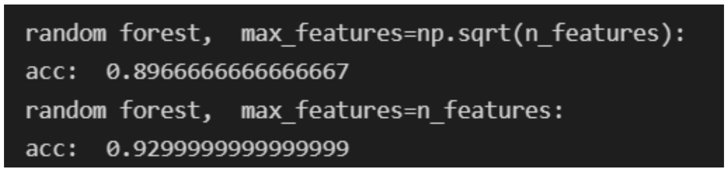
5. (15%) Implement the Random Forest algorithm by using the CART you just implemented from question 2. You should implement **three arguments** for the Random Forest.
   1) **N_estimators**: The number of trees in the forest.
   2) **Max_features**: The number of features to consider when looking for the best split
   3) **Bootstrap**: Whether bootstrap samples are used when building trees

   5.1. Using Criterion='gini', Max_depth=None, Max_features=sqrt(n_features), Bootstrap=True**,** showing the accuracy score of test data by n_estimators=10 and n_estimators=100, respectively.



```
random forest, n_estimators=10:
acc:  0.91
random forest, n_estimators=100:
acc:  0.9299999999999999
```

5.2. Using Criterion='gini', Max_depth=None, N_estimators=10, Bootstrap=True, showing the accuracy score of test data by Max_features=sqrt(n_features) and Max_features=n_features, respectively.



```
random forest,  max_features=np.sqrt(n_features):
acc:  0.8966666666666667
random forest,  max_features=n_features:
acc:  0.9299999999999999
```

Note: Use majority votes to get the final prediction, you may get different results when re-building the random forest model

6. (20%) Tune the hyperparameter, perform feature engineering or implement more powerful ensemble methods to get a higher accuracy score. Please note that only the ensemble method can be used. The neural network method is not allowed.

Accuracy	**Your scores**
acc > 0.975	**20 points**
0.95 < acc <= 0.975	**15 points**
0.9 < acc <= 0.95	**10 points**
acc < 0.9	**0 points**

Part. 2, Questions (30%):

1. Why does a decision tree have a tendency to overfit to the training set? Is it possible for a decision tree to reach a 100% accuracy in the training set? please explain. List and describe at least 3 strategies we can use to reduce the risk of overfitting of a decision tree.

當訓練資料雜訊多、數據不足時，decision tree深度太深容易把訓練資料雜訊也當成是分類依據，造成overfit：在training set上可以表現很好，若分得夠細時也可能達到100% accuracy (例如分類到每個leaf node只有一筆資料)。

strategies to reduce overfitting:

1. 減少training set的雜訊，例如data pruning
2. 設定合理不過深的max depth
3. 加入更多訓練資料

2. This part consists of three True/False questions. Answer True/False for each question and briefly explain your answer.

   a. In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor.

      T，因為AdaBoost對於分錯類的資料會加重權重：

$$D_{t+1}(i) = \frac{D_t(i)\, e^{(-a_t y_i h_t(x_i))}}{Z_t}$$

      上圖為權重更新公式，會針對每個資料(Xi)由分類器Ht判斷分類，作為權重決定因素之一

   b. In AdaBoost, weighted training error ε_t of the t_{th} weak classifier on training data with weights D_t tends to increase as a function of t.

      T，資料每次被分錯類時都會造成權重(D_t)增加、所以第t_{th}個 weak classifier 的weighted training error ε_t 也會增加

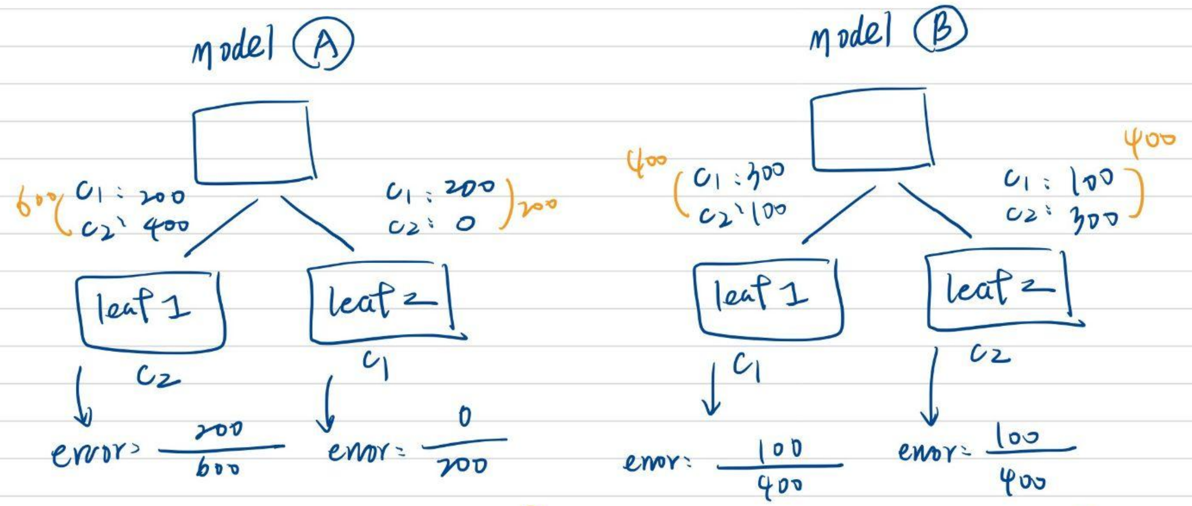
   c. AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

      F，如果無法從weak classifier線性組合出能完全分類資料的分類器則無法達成zero training error

3. Consider a data set comprising 400 data points from class C_1 and 400 data points from class C_2. Suppose that a tree model A splits these into (200, 400) at the first leaf node and (200, 0) at the second leaf node, where (n, m) denotes that n points are assigned to C_1 and m points are assigned to C_2. Similarly, suppose that a second tree model B splits them

into (300, 100) and (100, 300). **Evaluate the <u>misclassification rates</u> for the two trees and hence show that they are equal**. Similarly, **evaluate the cross-entropy** $Entropy = -\sum_{k=1}^{K} p_k \log_2 p_k$ and **Gini index** $Gini = 1 - \sum_{k=1}^{K} p_k^2$ **for the two trees**. Define p_k to be the proportion of data points in region R assigned to class k, where k = 1, . . . , K.

data $\begin{cases} C_1 = 400 \\ C_2 = 400 \end{cases}$



$\hookrightarrow \frac{600}{800} \times \frac{200}{600} + \frac{200}{800} \times \frac{0}{200} = \frac{1}{4}$ $\hookrightarrow \frac{400}{800} \times \frac{100}{400} + \frac{400}{800} \times \frac{100}{400} = \frac{1}{4}$

$\rightarrow$ misclassification rate both $= \frac{1}{4}$.

4.