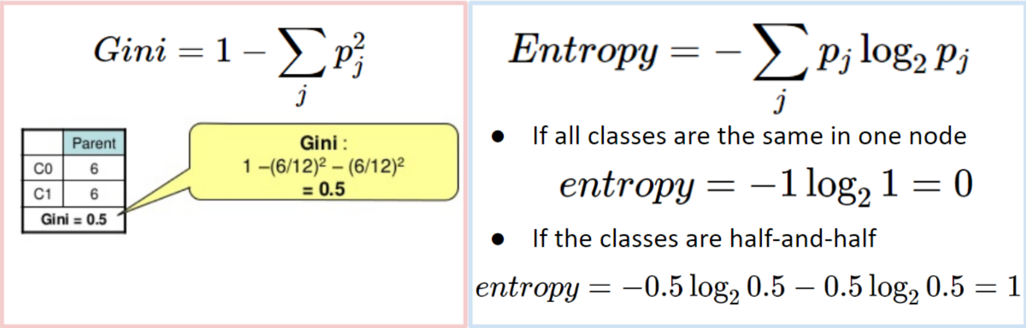
NYCU Introduction to Machine Learning, Homework

109550136 邱弘竣

Part. 1, Coding (80%):

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 of data is  0.4628099173553719
Entropy of data is  0.9456603046006401
(1200, 21)
(300, 21)
```

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time	three_g
0	1583	1	2.1	1	11	0	14	0.7	148	7	...	942	1651	1704	17	13	2	1
1	745	1	0.6	1	5	0	35	0.8	102	8	...	89	1538	2459	14	1	16	1
2	832	0	0.7	1	2	1	39	0.7	103	4	...	125	1504	1799	5	2	11	1
3	1175	1	1.3	0	2	0	19	0.3	164	7	...	873	1394	1944	9	4	9	1
4	695	0	0.5	0	18	1	12	0.6	196	2	...	1649	1829	2855	16	13	7	1

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.

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

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`

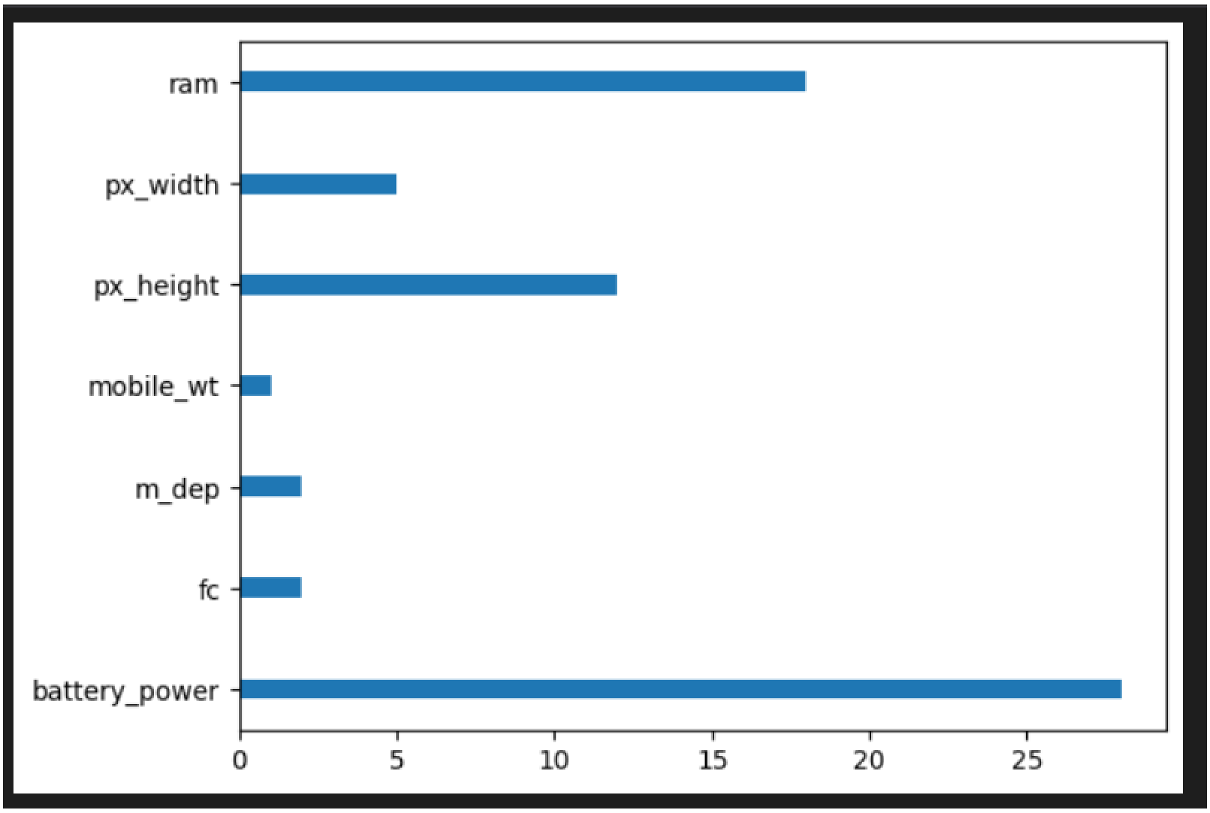
*Note: You should get the same results when re-building the model with the same arguments, **no need to prune the trees***

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

```
0.92
0.93
0.92
0.9333333333333333
```

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.

```
0.95
0.9733333333333334
```

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.

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.

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

```
0.9133333333333333
0.9366666666666666
```

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.

ans.

The reason is that they are very data intensive. They examine the data in a lot of ways and look at every possible split of every independent variable. Even with a relatively small number of variables, they can be a log of things to examine, especially if one of them is a categorical variable with more than a few levels.

It is possible for the training set to reach a 100% accuracy if there is enough depth. However, it might increase the risk of overfitting.

Here are the three strategies we can use to reduce the risk of overfitting of a decision tree. To begin with, pruning is a technique aiming to remove the parts of the decision tree to prevent growing to its full depth. The first method is pre-pruning. The pre-pruning technique refers to the early stopping of the growth of the decision tree. The second method is post-pruning. The post-pruning technique allows the decision tree model to grow to its full depth, then removes the tree branchese to prevent the model from overfitting. Besides, the third method to deal with the problem of overfitting is random forest. Random forest is an ensemble technique follows bootstrap sampling and aggregation techniques to prevent overfitting.

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.

      ans.

      True. The weights of all misclassified points will be multiplied by exp(amount_of_say) before normalization. The formua of amount_of_say is (1/2)*log((1-total_error)/total_error).
   b. In AdaBoost, weighted training error $\boldsymbol{\varepsilon}_t$ of the t_{th} weak classifier on training data with weights D_t tends to increase as a function of t.

      ans.

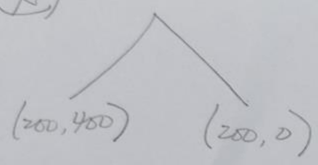
      True. The weights will increase for the data that are repeatedly misclassified by the weak classifiers. The weighted training error of the t th weak classifier on the training data therefore tends to increase.
   c. AdaBoost will eventually give zero training error regardless of the type of weak classifier it uses, provided enough iterations are performed.

      ans.

      The answer is false if the data in the training set cannot be separated by a linear combination of the specific type of weak classifiers we are using. No matter how many iterations are performed, we cannot get 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 p

oints 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.

3. (A)

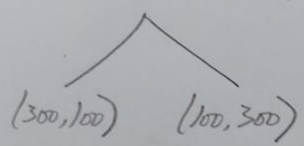


misclassification rate $= \frac{200}{800} = \frac{1}{4}$

entropy
left: $-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} \fallingdotseq 0.918$
right: $-1\log_2 1 = 0$

gini
left: $1-(\frac{1}{3})^2-(\frac{2}{3})^2 = \frac{4}{9}$
right: $1-(1)^2 = 0$

(B)



misclassification rate $= \frac{100+100}{800} = \frac{1}{4}$

entropy
left: $-\frac{3}{4}\log_2(\frac{3}{4}) - \frac{1}{4}\log_2(\frac{1}{4}) \fallingdotseq 0.811$
right: $-\frac{1}{4}\log_2(\frac{1}{4}) - \frac{3}{4}\log_2(\frac{3}{4}) \fallingdotseq 0.811$

gini
left: $1-(\frac{3}{4})^2-(\frac{1}{4})^2 = \frac{3}{8}$
right: $1-(\frac{1}{4})^2-(\frac{3}{4})^2 = \frac{3}{8}$