NYCU Introduction to Machine Learning, Homework 3

109550116 楊傑宇

Part. 1, Coding (80%):

1.

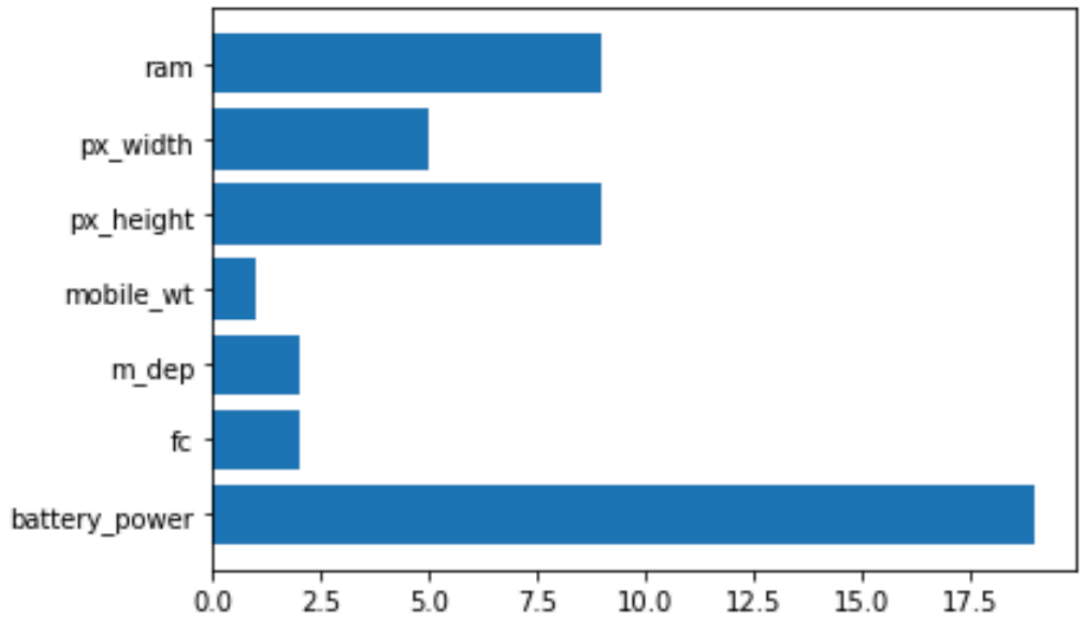
```
print("Gini of data is ", gini(data))
✓ 0.7s
Gini of data is  0.4628099173553719

print("Entropy of data is ", entropy(data))
✓ 0.5s
Entropy of data is  0.9456603046006401
```

2.

```
Accuracy of depth=3 0.92
Accuracy of depth=10 0.93
```

```
Accuracy of gini 0.92
Accuracy of entropy 0.9333333333333333
```

3.



4.

```
Accuracy of n_estimators=10 0.94
Accuracy of n_estimators=100 0.9733333333333334
```

5.

```python
clf_10tree = RandomForest(n_estimators=10, max_features=np.sqrt(x_data.sha
clf_100tree = RandomForest(n_estimators=100, max_features=np.sqrt(x_data.s
clf_10tree.fit(x_data, y_data)
clf_100tree.fit(x_data, y_data)

pred_10tree = clf_10tree.predict(x_val)
pred_100tree = clf_100tree.predict(x_val)

acc_10tree=accuracy_score(y_val, pred_10tree)
acc_100tree=accuracy_score(y_val, pred_100tree)

print(f"Accuracy of n_estimators=10 {acc_10tree}")
print(f"Accuracy of n_estimators=100 {acc_100tree}")
```

✓ 2m 40.1s Python

```
Accuracy of n_estimators=10 0.9233333333333333
Accuracy of n_estimators=100 0.95
```

```python
clf_random_features = RandomForest(n_estimators=10, max_features=np.sqrt(x
clf_all_features = RandomForest(n_estimators=10, max_features=x_data.shape

clf_random_features.fit(x_data, y_data)
clf_all_features.fit(x_data, y_data)

pred_clf_random_features = clf_random_features.predict(x_val)
pred_clf_all_features = clf_all_features.predict(x_val)

acc_clf_random_features=accuracy_score(y_val, pred_clf_random_features)
acc_clf_all_features=accuracy_score(y_val, pred_clf_all_features)

print(f"Accuracy of random features {acc_clf_random_features}")
print(f"Accuracy of all features {acc_clf_all_features}")
```

✓ 1m 19.6s Python

```
Accuracy of random features 0.95
Accuracy of all features 0.9633333333333334
```

6.

```
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  1.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0. -1.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0. -1.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
0.9866666666666667
```

這是我用 n_estimators=150 的 adaboost 去跑 validation set 得到的結果，我便拿 n_estimators=150 的 adaboost training validation set + training set 來當 model

Part. 2, Questions (30%):

1.

Decision tree 有 overfitting 的趨勢是因為當層數越來越大，對 data 的分類就越來越嚴格，然而這僅僅只能使 training data 分的越清楚，但對於非 training data 的 data 準確度就會下降因此會有 overfitting 的趨勢

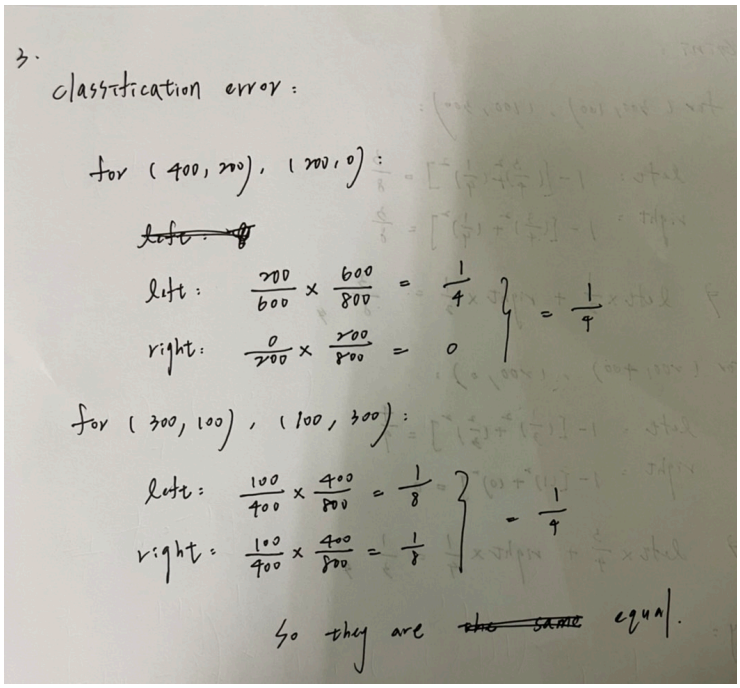
100%是有可能的若是層數足夠大我一定可以讓整棵樹變成二分法的樣子，而二分法的樹對原本的 data 的 accuracy 一定是 100%

- Pre-pruning: 加入參數讓樹提早停止製造 node
- Post-pruning: use ccp to remove node from full depth tree
- Ensemble: like random forest use bootstrapping to prevent overfitting

2.

- a. yes 根據 exp()update function weight 會增加，為了要讓它再下一顆樹被特別關注，加強分類它
- b. yes 因為 classifiers 為了讓 difficult data 加強分辨，因此如果 difficult 一直被 misclassified，weight 就會一直增加。所以 weighted training error tends to increase.
- c. no 若我們用原本的 classifier，每棵樹只有一個特徵來分辨，那無論 iterations 多大都沒辦使 EXOR example 達成 zero training error

3.


3.
classification error:
for (400, 200), (200, 0):
left: 200/600 × 600/800 = 1/4
right: 0/200 × 200/800 = 0
= 1/4
for (300, 100), (100, 300):
left: 100/400 × 400/800 = 1/8
right: 100/400 × 400/800 = 1/8
= 1/4
So they are equal.

Gini:

for (300, 100), (100, 300):

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

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

$\Rightarrow$ left $\times\frac{1}{2}$ + right $\times\frac{1}{2}=\frac{3}{8}$ ✓

for (200, 400), (200, 0):

left: $1-\left[\left(\frac{1}{3}\right)^2+\left(\frac{2}{3}\right)^2\right]=\frac{4}{9}$

right: $1-\left[(1)^2+(0)^2\right]=0$

$\Rightarrow$ left $\times\frac{3}{4}$ + right $\times\frac{1}{4}=\frac{1}{3}$ ✓

entropy:

for (300, 100), (100, 300):

left: $-\left[\frac{3}{4}\log_2\frac{3}{4}+\frac{1}{4}\log_2\frac{1}{4}\right]\approx 0.81$

right: $-\left[\frac{3}{4}\log_2\frac{3}{4}+\frac{1}{4}\log_2\frac{1}{4}\right]\approx 0.81$

$\Rightarrow$ left $\times\frac{1}{2}$ + right $\times\frac{1}{2}=0.81$

for (200, 400), (200, 0)

left: $-\left[\frac{2}{6}\log_2\frac{2}{6}+\frac{4}{6}\log_2\frac{4}{6}\right]\approx 0.92$

right: 0.

$\Rightarrow$ left $\times\frac{6}{8}$ + right $\times\frac{1}{4}=0.69$ ✓