0816170 Homework 3

Part 1

1. Gini and Entropy

```
[26]  1 print("Gini of data is ", gini(data))
```

```
Gini of data is  0.4628099173553719
```

```
[27]  1 print("Entropy of data is ", entropy(data))
```

```
Entropy of data is  0.9456603046006402
```

2. Decision Tree

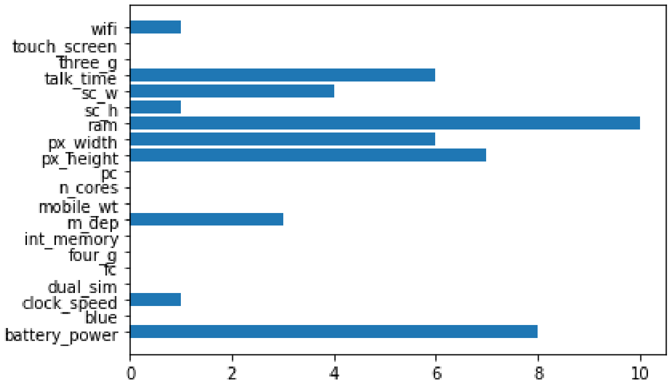
2.1 Max_depth = 3 and Max_depth = 10

```
Decision Tree
max_depth = 3, accuracy score: 0.92
max_depth = 10, accuracy score: 0.93
```

2.2 Criterion = 'gini' and Criterion = 'entropy'

```
Decision Tree
Criterion= 'gini', accuracy score: 0.92
Criterion='entropy' , accuracy score: 0.9333333333333333
```

3. Feature importance



4. AdaBoost

4.1

```
AdaBooest
n_estimators=10, accuracy score: 0.9366666666666666
n_estimators=100, accuracy score: 0.9733333333333334
```

5. Random Forest

5.1

```
Random Forest
n_estimators=10, accuracy score: 0.93
n_estimators=100, accuracy score: 0.96
```

5.2

```
Random Forest
random features, accuracy score: 0.9333333333333333
all features, accuracy score: 0.9666666666666667
```

6. My_model

使用 adaboost，n_estimators = 150 (經實驗發現最高可以到達 0.98 的準確率)

將 train_df 跟 val_df 一起送進去訓練

得到最終的 my_model

大約在 google colab 上跑 5 分鐘

```
[63]
from sklearn.metrics import accuracy_score

def train_your_model(data):
    ## Define your model and training
    x_train = data.drop(labels=["price_range"], axis="columns")
    x_train = x_train.values
    y_train = data["price_range"].values

    ada = 150
    adatest = AdaBoost(ada)
    adatest.fit(x_train, y_train)
    predtest = adatest.predict(x_val)
    return adatest
```

```
[64]
trainval_df = train_df.append(val_df)
my_model = train_your_model(trainval_df)
```

```
test_df = pd.read_csv('x_test.csv')
x_test = test_df.values
y_pred = ada100.predict(x_test)
```

```
assert y_pred.shape == (500, )
```

Final result

```
*** We will check your result for Question 3 manually *** (5 points)
*** We will check your result for Question 6 manually *** (20 points)
Approximate score range: 45.0 ~ 70.0
*** This score is only for reference ***
```

Part 2

1. ① 因為decision tree會想盡辦法把train data中的兩種class分開來，因此只要還有branch資料不純，decision tree會為了把不純的地方挑出來而選用一些不太有代表性的feature和threshold，如此就可能導致overfit

② 可能達到100%，若training set的X資料皆獨一無二，decision tree總會找到feature和threshold把不同class的data隔開（code實作中也有發生）

③

(1) pre-pruning：例如調整max_depth控制tree的深度讓tree不會過度split

(2) post-pruning：先生成一棵完整的tree，再移除部分branch，讓tree不要那麼雜

(3) random forest：應用boostrap sampling和data aggregation使tree減少overfitting的機率

2

a True, 根據 update equation 知, 若分類錯誤, 該 data 的 weight 變大

b True, 在訓練過程中, weak classifier 被迫嘗試分類更困難的 examples
若某 example 一直被分類錯, 該 example 的 weight 會增加, 而 ϵ_t of the t^{th} weak classifier 也因此傾向增加

c false, 在部分情況中, 若 training set 中有 data 無法被我們使用的 weak classifier 分割, 無論遞迴多少次都無法達到 zero training error

3

1. misclassification rates

model A : $\frac{200+0}{800} = \frac{1}{4}$

model B : $\frac{100+100}{800} = \frac{1}{4}$

$\Rightarrow$ their misclassification rates are equal

2. model A

(200, 400)

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

$entropy = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} \fallingdotseq 0.918$

(200, 0)

$Gini = 1-1^2 = 0$

$entropy = -1\log_2 1 = 0$

model B

$$\begin{tikzcd}[column sep=small] & \text{model B} \arrow[dl] \arrow[dr] & \\ (300, 100) & & (100, 300) \end{tikzcd}$$

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

$entropy = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} \fallingdotseq 0.811$

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

$entropy = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} \fallingdotseq 0.811$