# Part 1. Coding

Q1: Gini: 0.462, Entropy: 0.9456

```
print("Gini of data is ", gini(data))
```

```
Gini of data is  0.4628099173553719
```

```
print("Entropy of data is ", entropy(data))
```

```
Entropy of data is  0.9456603046006401
```

Q2.1: 0.9166, 0.94

## Question 2.1

Using `criterion=gini`, showing the accuracy score of validation data by `max_depth=3` and `max_depth=10`, respectively.

```
clf_depth3 = DecisionTree(criterion='gini', max_depth=3)
clf_depth3.fit(x_train, y_train)
print("clf_depth3: ", accuracy_score(y_test, clf_depth3.predict(x_test)))

clf_depth10 = DecisionTree(criterion='gini', max_depth=10)
clf_depth10.fit(x_train, y_train)
print("clf_depth10: ", accuracy_score(y_test, clf_depth10.predict(x_test)))
```

```
clf_depth3:  0.9166666666666666
clf_depth10:  0.94
```

Q2.2: 0.9166, 0.93

## Question 2.2

Using `max_depth=3`, showing the accuracy score of validation data by `criterion=gini` and `criterion=entropy`, respectively.

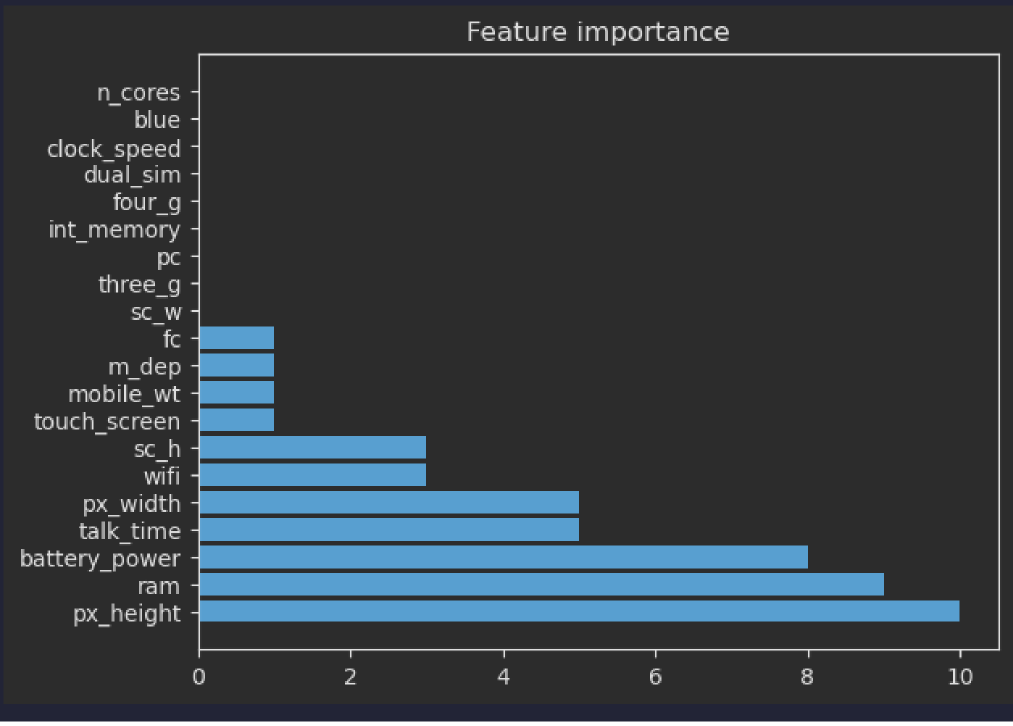
```
clf_gini = DecisionTree(criterion='gini', max_depth=3)
clf_gini.fit(x_train, y_train)
print("clf_gini: ", accuracy_score(y_test, clf_gini.predict(x_test)))

clf_entropy = DecisionTree(criterion='entropy', max_depth=3)
clf_entropy.fit(x_train, y_train)
print("clf_entropy: ", accuracy_score(y_test, clf_entropy.predict(x_test)))
```

```
clf_gini:  0.9166666666666666
clf_entropy:  0.93
```

Q3:

```
plt.show()
```




Q4.1: 0.94, 0.966

## Question 4.1

Show the accuracy score of validation data by `n_estimators=10` and `n_estimators=100`, respectively.

```
clf_adaboost10 = AdaBoost(n_estimators=10)
clf_adaboost10.fit(x_train, y_train)
print("clf_adaboost10: ", accuracy_score(y_test, clf_adaboost10.predict(x_test)))

clf_adaboost100 = AdaBoost(n_estimators=100)
clf_adaboost100.fit(x_train, y_train)
print("clf_adaboost100: ", accuracy_score(y_test, clf_adaboost100.predict(x_test)))
```

```
clf_adaboost10:  0.94
clf_adaboost100:  0.9666666666666667
```

Q5.1: 0.94, 0.9366

### Question 5.1

Using `criterion=gini`, `max_depth=None`, `max_features=sqrt(n_features)`, showing the accuracy score of validation data by `n_estimators=10` and `n_estimators=100`, respectively.

```
clf_10tree = RandomForest(n_estimators=10, max_features=np.sqrt(x_train.shape[1]))
clf_10tree.fit(x_train, y_train)
print("clf_10tree: ", accuracy_score(y_test, clf_10tree.predict(x_test)))

clf_100tree = RandomForest(n_estimators=100, max_features=np.sqrt(x_train.shape[1]))
clf_100tree.fit(x_train, y_train)
print("clf_100tree: ", accuracy_score(y_test, clf_100tree.predict(x_test)))
```

```
clf_10tree:  0.94
clf_100tree:  0.9366666666666666
```

Q5.2: 0.95, 0.95

### Question 5.2

Using `criterion=gini`, `max_depth=None`, `n_estimators=10`, showing the accuracy score of validation data by `max_features=sqrt(n_features)` and `max_features=n_features`, respectively.

```
clf_random_features = RandomForest(n_estimators=10, max_features=np.sqrt(x_train.shape[1]))
clf_random_features.fit(x_train, y_train)
print("clf_random_features: ", accuracy_score(y_test, clf_random_features.predict(x_test)))

clf_all_features = RandomForest(n_estimators=10, max_features=x_train.shape[1])
clf_all_features.fit(x_train, y_train)
print("clf_all_features: ", accuracy_score(y_test, clf_all_features.predict(x_test)))
```

```
clf_random_features:  0.95
clf_all_features:  0.95
```

# Part 2. Questions

Q1:

When the tree grows one level deeper, the dataset is separated into halves. So, it can converge fast and easy to overfit.

When we do not give the tree a seperation limit, it will eventually become one training data on each leaf, which perfectly fits the training set.

1. Limit the max depth.
2. Randomly choose the label to calculate best split.
3. Terminate when the information gain is too low.

Q2:

a. False. The factor is related to the sum of misclassified examples' weights.
b. True. The weak classifier tends to fit the data with higher weight, and ignore those with lower weight, so the sum of errors tends to increase.
c. True. Given enough lines/surfaces, they can eventually cut a space to separate each data from different label.

Q3:

$$\text{Misclassification rate of A: first leaf} = C_2, \text{second leaf} = C_1$$

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

Misclassification rate of B: first leaf $= \mathrm{C}_1$,second leaf $= \mathrm{C}_2$

$$\frac{100+100}{800}=\frac{1}{4}=\text{Misclassification rate of A}$$

Tree A:

$$\text{Entropy} = -\left(\frac{200}{600}\log_2\frac{200}{600}+\frac{400}{600}\log_2\frac{400}{600}\right)*\frac{600}{800}-\left(\frac{200}{200}\log_2\frac{200}{200}\right)*\frac{200}{800}$$

$$= -\left(\frac{1}{3}\log_2\frac{1}{3}+\frac{2}{3}\log_2\frac{2}{3}\right)*\frac{3}{4}\approx 0.68872$$

$$\text{Gini} = \left(1-\left(\left(\frac{200}{600}\right)^2+\left(\frac{400}{600}\right)^2\right)\right)*\frac{600}{800}+\left(1-\left(\frac{200}{200}\right)^2\right)*\frac{200}{800}$$

$$= \left(1-\left(\left(\frac{1}{3}\right)^2+\left(\frac{2}{3}\right)^2\right)\right)*\frac{3}{4}\approx 0.33333$$

Tree B:

$$\text{Entropy} = -\left(\frac{300}{400}\log_2\frac{300}{400}+\frac{100}{400}\log_2\frac{100}{400}\right)*\frac{400}{800}$$

$$-\left(\frac{100}{400}\log_2\frac{100}{400}+\frac{300}{400}\log_2\frac{300}{400}\right)*\frac{400}{800}$$

$$= -\left(\frac{3}{4}\log_2\frac{3}{4}+\frac{1}{4}\log_2\frac{1}{4}\right)\approx 0.81127$$

$$\text{Gini} = \left(1-\left(\left(\frac{300}{400}\right)^2+\left(\frac{100}{400}\right)^2\right)\right)*\frac{400}{800}+\left(1-\left(\left(\frac{100}{400}\right)^2+\left(\frac{300}{400}\right)^2\right)\right)*\frac{400}{800}$$

$$= \left(1-\left(\left(\frac{3}{4}\right)^2+\left(\frac{1}{4}\right)^2\right)\right)=0.375$$