Name: Lam Chun Ting Jeff
ITSC: ctjlam
SID: 12222973

Q1.
Decision Tree

Summary Table

Classifier	Accuracy	Precision	Recall	F1	Training Time
Decision Tree Entropy, depth 5	0.52	0.5	0.52	0.48	0.01106
Decision Tree Entropy, depth 10	0.53	0.52	0.53	0.52	0.05246
Decision Tree Entropy, depth 15	0.56	0.55	0.56	0.55	0.06496
Decision Tree Entropy, depth 20	0.59	0.58	0.59	0.58	0.06730
Decision Tree Gini, depth 5	0.53	0.52	0.53	0.49	0.02451
Decision Tree Gini, depth 10	0.55	0.54	0.55	0.54	0.04212
Decision Tree Gini, depth 15	0.58	0.58	0.58	0.58	0.05089
Decision Tree Gini, depth 20	0.59	0.59	0.59	0.59	0.05395

We can see for both entropy and Gini computation, when depth increase, both training time and accuracy will increase, as more depth will implies more computation, and more details can be separated. For the difference between Gini and entropy, the training time at first entropy takes shorter time comparing to Gini, but later Gini computes faster than entropy. Entropy should takes longer time as the computation as entropy requires to compute split info while Gini will have one equation less than entropy. But we can see rather Gini is better than Entropy from the result that for all accuracy, precision, recall and F1, the value is a bit higher. In others words, the true-positive case has a more proportion to other cases in prediction result.

Name: Lam Chun Ting Jeff
ITSC: ctjlam
SID: 12222973

Program Output

Entropy

```
==================================================
Summary for decision tree (Entropy, depth = 5)
Training time: 0.011060953140258789s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.23      0.07      0.11        40
           5       0.54      0.55      0.55       289
           6       0.51      0.71      0.59       434
           7       0.59      0.19      0.29       176
           8       0.11      0.03      0.04        37

    accuracy                           0.52       981
   macro avg       0.33      0.26      0.26       981
weighted avg       0.50      0.52      0.48       981

==================================================
```

```
==================================================
Summary for decision tree (Entropy, depth = 10)
Training time: 0.052455902099609375s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.32      0.23      0.26        40
           5       0.55      0.54      0.54       289
           6       0.55      0.65      0.60       434
           7       0.51      0.38      0.44       176
           8       0.18      0.11      0.14        37
           9       0.00      0.00      0.00         0

    accuracy                           0.53       981
   macro avg       0.30      0.27      0.28       981
weighted avg       0.52      0.53      0.52       981

==================================================
```

```
==================================================
Summary for decision tree (Entropy, depth = 15)
Training time: 0.06495714187622207s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.33      0.20      0.25        40
           5       0.59      0.61      0.60       289
           6       0.59      0.59      0.59       434
           7       0.52      0.55      0.54       176
           8       0.31      0.27      0.29        37

    accuracy                           0.56       981
   macro avg       0.39      0.37      0.38       981
weighted avg       0.55      0.56      0.55       981

==================================================
```

```
==================================================
Summary for decision tree (Entropy, depth = 20)
Training time: 0.067301750183105475s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.30      0.23      0.26        40
           5       0.62      0.60      0.61       289
           6       0.62      0.66      0.64       434
           7       0.56      0.54      0.55       176
           8       0.34      0.32      0.33        37
           9       0.00      0.00      0.00         0

    accuracy                           0.59       981
   macro avg       0.35      0.34      0.34       981
weighted avg       0.58      0.59      0.58       981

==================================================
```

Gini

```
==================================================
Summary for decision tree (gini, depth = 5)
Training time: 0.024505138397216797s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.56      0.12      0.20        40
           5       0.55      0.56      0.55       289
           6       0.51      0.72      0.60       434
           7       0.59      0.22      0.32       176
           8       0.00      0.00      0.00        37

    accuracy                           0.53       981
   macro avg       0.37      0.27      0.28       981
weighted avg       0.52      0.53      0.49       981

==================================================
```

```
==================================================
Summary for decision tree (gini, depth = 10)
Training time: 0.042118072509765625s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.44      0.28      0.34        40
           5       0.61      0.57      0.59       289
           6       0.56      0.64      0.60       434
           7       0.48      0.47      0.47       176
           8       0.33      0.08      0.13        37

    accuracy                           0.55       981
   macro avg       0.40      0.34      0.35       981
weighted avg       0.54      0.55      0.54       981

==================================================
```

```
==================================================
Summary for decision tree (gini, depth = 15)
Training time: 0.05089378356933594s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.37      0.33      0.35        40
           5       0.60      0.60      0.60       289
           6       0.63      0.65      0.64       434
           7       0.55      0.54      0.55       176
           8       0.32      0.24      0.28        37

    accuracy                           0.58       981
   macro avg       0.41      0.39      0.40       981
weighted avg       0.58      0.58      0.58       981

==================================================
```

```
==================================================
Summary for decision tree (gini, depth = 20)
Training time: 0.05394577980041504s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.37      0.33      0.35        40
           5       0.62      0.60      0.61       289
           6       0.61      0.65      0.63       434
           7       0.57      0.57      0.57       176
           8       0.35      0.30      0.32        37
           9       0.00      0.00      0.00         0

    accuracy                           0.59       981
   macro avg       0.36      0.35      0.35       981
weighted avg       0.59      0.59      0.59       981

==================================================
```

Name: Lam Chun Ting Jeff
ITSC: ctjlam
SID: 12222973

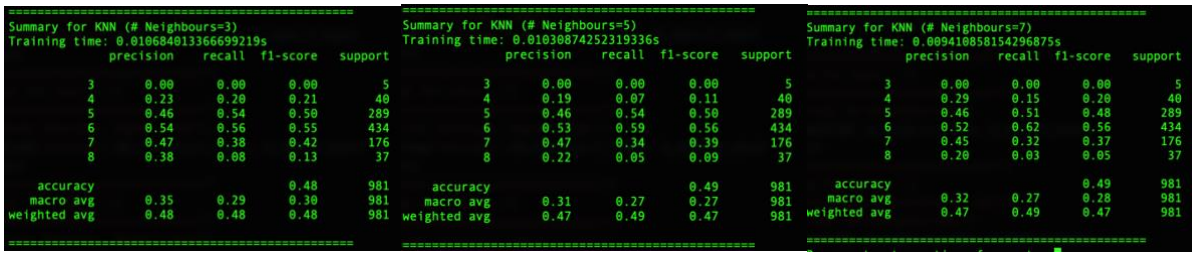
Classifier	Accuracy	Precision	Recall	F1	Training Time
KNN # Neighbors 3	0.48	0.48	0.48	0.48	0.01068
KNN # Neighbors 5	0.49	0.47	0.49	0.47	0.01031
KNN # Neighbors 7	0.49	0.47	0.49	0.47	0.00941
Random Forest # Estimator=10	0.66	0.66	0.66	0.64	0.08291
Random Forest # Estimator=50	0.66	0.66	0.66	0.65	0.30515
Random Forest # Estimator=100	0.65	0.67	0.65	0.63	0.56236

Generally, training time in Random Forest is much higher than KNN. The major reasons as Random Forests will generate lots of classifier and helps to better generalize the problem. And therefore, the accuracy is much higher than KNN, in which KNN is a simple and straight forward classifier.

For KNN, we can see that although accuracy will increase when number of neighbors increase from 3 to 5, while stay the same at 7, the true positive among with false positive, is smaller than false negative, which precision is smaller than recall.

For random forest, in this case we choose sklearn default value for number of estimators, 10 before 0.22 version, and 100 after 0.22 version, and a middle value 50. We can actually see 10 to 50, F1 score increases, in other words, true positive cases have larger proportion, which means the real accuracy increases. As we cannot see an actual growth in accuracy in 10 to 50, then for random forest 100, it is believed that it has been over-generalize. Therefore, no reference value for the cases. In short, increase in number of estimators will increase in accuracy, and training time.

KNN

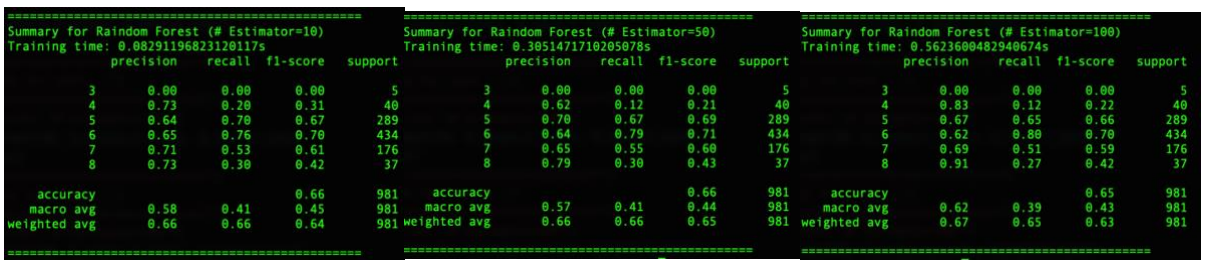


```
Summary for KNN (# Neighbours=3)
Training time: 0.010684013366699219s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.23      0.20      0.21        40
           5       0.46      0.54      0.50       289
           6       0.54      0.56      0.55       434
           7       0.47      0.38      0.42       176
           8       0.38      0.08      0.13        37

    accuracy                           0.48       981
   macro avg       0.35      0.29      0.30       981
weighted avg       0.48      0.48      0.48       981
```

```
Summary for KNN (# Neighbours=5)
Training time: 0.010308742523193336s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.19      0.07      0.11        40
           5       0.46      0.54      0.50       289
           6       0.53      0.59      0.56       434
           7       0.47      0.34      0.39       176
           8       0.22      0.05      0.09        37

    accuracy                           0.49       981
   macro avg       0.31      0.27      0.27       981
weighted avg       0.47      0.49      0.47       981
```

```
Summary for KNN (# Neighbours=7)
Training time: 0.009410858154296875s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.29      0.15      0.20        40
           5       0.46      0.51      0.48       289
           6       0.52      0.62      0.56       434
           7       0.45      0.32      0.37       176
           8       0.20      0.03      0.05        37

    accuracy                           0.49       981
   macro avg       0.32      0.27      0.28       981
weighted avg       0.47      0.49      0.47       981
```

Random Forest



```
Summary for Raindom Forest (# Estimator=10)
Training time: 0.08291196823120117s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.73      0.20      0.31        40
           5       0.64      0.70      0.67       289
           6       0.65      0.76      0.70       434
           7       0.71      0.53      0.61       176
           8       0.73      0.30      0.42        37

    accuracy                           0.66       981
   macro avg       0.58      0.41      0.45       981
weighted avg       0.66      0.66      0.64       981
```

```
Summary for Raindom Forest (# Estimator=50)
Training time: 0.3051471710205078s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.62      0.12      0.21        40
           5       0.70      0.67      0.69       289
           6       0.64      0.79      0.71       434
           7       0.65      0.55      0.60       176
           8       0.79      0.30      0.43        37

    accuracy                           0.66       981
   macro avg       0.57      0.41      0.44       981
weighted avg       0.66      0.66      0.65       981
```

```
Summary for Raindom Forest (# Estimator=100)
Training time: 0.5623600482940674s
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         5
           4       0.83      0.12      0.22        40
           5       0.67      0.65      0.66       289
           6       0.62      0.80      0.70       434
           7       0.69      0.51      0.59       176
           8       0.91      0.27      0.42        37

    accuracy                           0.65       981
   macro avg       0.62      0.39      0.43       981
weighted avg       0.67      0.65      0.63       981
```

Name: Lam Chun Ting Jeff
ITSC: ctjlam
SID: 12222973

Q2.

```
========== Classifier 1 ==========
Aplhas: 1.7380493449176364
Weak Classifier - x > v (where v = [1.5]) then y = -1 else y = 1
====================================
========== Classifier 2 ==========
Aplhas: 2.6663593966326844
Weak Classifier - x > v (where v = [0.5]) then y = 1 else y = 1
====================================
========== Classifier 3 ==========
Aplhas: 2.5652451279522674
Weak Classifier - x > v (where v = [8.5]) then y = 1 else y = -1
====================================
========== Classifier 4 ==========
Aplhas: 2.6179223685983244
Weak Classifier - x > v (where v = [6.5]) then y = -1 else y = 1
====================================
========== Classifier 5 ==========
Aplhas: 3.7439507138604458
Weak Classifier - x > v (where v = [4.5]) then y = 1 else y = -1
====================================
Prediction: [ 1.  1. -1. -1. -1.  1.  1. -1. -1.  1.]
Accuracy: 1.0
```

Note: result may various during runtime because it purely depends on how the training data randomly sampled, which follow the instruction in lecture slide. There is commented code to get static training set for training such that the result will be much constant (but it will violate the lecture instruction. However it can get a static minimal number of classifier for classification)

C*(x) = sign[1.738*C1 + 2.666*C2+ 2.565*C3 + 2.6179*C4+ 3.744*C5]

Where

C1: x>1.5, then y = -1, else y = 1
C2: x>0.5, then y = 1, else y = 1
C3: x>8.5, then y = 1, else y = -1
C4: x>6.5, then y = -1, else y = 1
C5: x>4.5, then y = 1, else y = 1