Assignment 1 – Report

Navjot Kaur (301404765)

a. Pseudo-code for a simple decision tree:

```
DecisionTree(self, node n, dataset X, prediction column y, features f):
        # Termination condition
        If (sizeOfDataset < self.minimum_sample_splits or n.depth >= self.max_depth):
                return
        random_features := apply feature randomness without repetition using
random.sample(f, self.max_features)

        for all i in range(len(random_features)):
                score := using Gini index the scores are calculated for every features
        feature_used : = The feature having minimum score is used as the feature on
which the splitting is going to be happened

    #for categorical feature
    If the feature to be split is categorical then:
        unique_categories := all unique variables contained in that feature
      for all i in range(unique_categories.shape[0])
        split := contains features of first kind
        create new node n_new as child of node n
        DecisionTree(n_new, split, y, features)
    else:
        #for numerical
        threshold := meanOf(feature_name)
        split1 := X[feature_name < threshold], all feature values less than threshold
        split2 := X[feature_name >= threshold], all feature values more than threshold
        create new node s1 as child of node
        DecisionTree(s1, split1, y, features)

        create new node s1 as child of node
        DecisionTree(s2, split2, y, features)
    return
```

b. Random forest algorithm is implemented using Python and all the functions that were asked to be implemented i.e, fit, split_node, predict, gini, and entropy are completed. The data is cleaned first since it contained many ' ?' values which were dropped. The column fnlwgt is being dropped because ETL suggests the lowest score for it.

Tasks:

1. The Random Forest model is trained on the following parameters:
   n_classifiers = 10
   maxdepth = 10
   min_sample_split = 20
   max_features = 11
   Using Gini index the accuracy of training and testing data are:

```
gini
0.8737815794708573
0.8312084993359894
```

2. The Random Forest model is trained on the following parameters:
   n_classifiers = 10
   maxdepth = 10
   min_sample_split = 20
   max_features = 11
   Using Information Gain, the accuracy of training and testing data are:

```
entropy
0.8745441283734501
0.8284860557768924
```

   The Information Gain reports slightly less accuracy for test data than Gini Index which states that (as mentioned in lecture), Gini is a good option for bigger distribution sets and is known to be a good splitting criterion for bigger partitions. Information gain however is 0.001% more accurate than Gini on training dataset, which can be due to outliers and/or when depth increases partitions get smaller and Information gain favors smaller datasets.
   From above discussion I would conclude Gini is a better splitting criterion for the given dataset.

3. The accuracy of training data is not equal to 100% due to outliers and/or missing values.
   a. Yes, there is a way with which accuracy on the training dataset can go up to 100% if the min_sample_splits are few and n_classifiers are increased with increase in depth, however this could mean overfitting, and this would drastically decrease the accuracy of test data.
   b. The parameters to be changed are decreaing the sample splits and increasing number of trees and depth.
4. The accuracy plot is:

```
5. ----------Depth  1 ----------
6. train:  0.7510775147536636
7. test:  0.7543160690571049
8. ----------Depth  2 ----------
9. train:  0.8172866520787746
```

```
test:  0.8172642762284197
----------Depth  3 ----------
train:  0.8298852861216099
test:  0.8280212483399735
----------Depth  4 ----------
train:  0.8458325044758305
test:  0.8315405046480744
----------Depth  5 ----------
train:  0.8569060407134805
test:  0.8347277556440903
----------Depth  6 ----------
train:  0.8662223990451562
test:  0.8358565737051793
----------Depth  7 ----------
train:  0.871659704263643
test:  0.8340637450199203
----------Depth  8 ----------
train:  0.8770307008819044
test:  0.8317397078353254
----------Depth  9 ----------
train:  0.881174988395995
test: 0.8315455078357804
----------Depth  10 ----------
train: 0.8847480642579753
test: 0.8303596900642692
```