Ayush Garg
PUID: 0033794803

Homework 4

CS 57300

1. Q1

```
Command to be entered on terminal: python preprocess-assg4.py
Output: Provides two files trainingSet.csv and testSet.csv
```

2. Q2

1. For running Decision Trees
   ```
   Command to be entered on terminal:
   python trees.py trainingSet.csv testSet.csv 1
   Output: Training Accuracy DT: 0.77
           Testing Accuracy DT: 0.72
   ```

2. For running Bagging
   ```
   Command to be entered on terminal:
   python trees.py trainingSet.csv testSet.csv 2
   Output: Training Accuracy BT: 0.79
           Testing Accuracy BT: 0.75
   ```

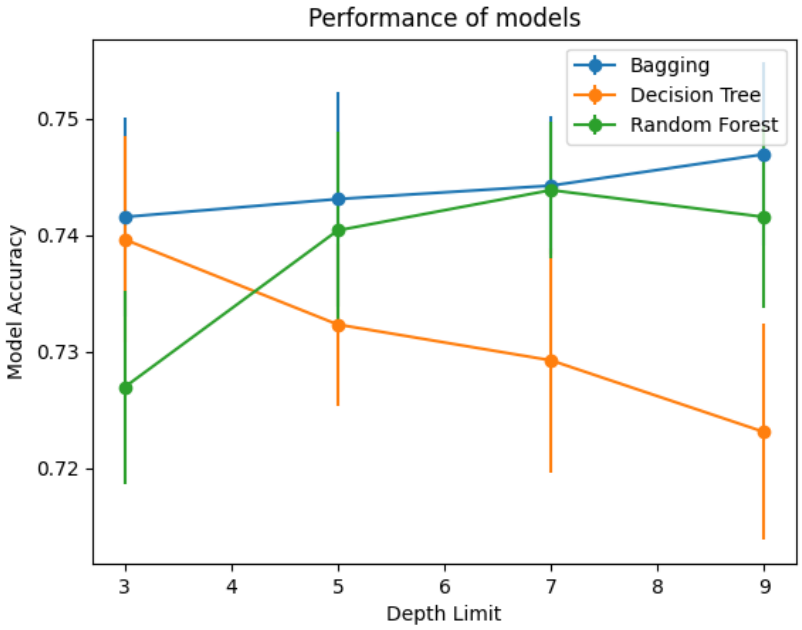
3. For running Random Forest
   ```
   Command to be entered on terminal:
   python trees.py trainingSet.csv testSet.csv 3
   Output: Training Accuracy RF: 0.76
           Testing Accuracy RF: 0.73
   ```

3. Q3

1. Comparing models accuracy

   ```
   Command to be entered on terminal: python cv_depth.py
   Output: Plot showing the accuracies for all the three models
   ```

2. Plot obtained after comparing the accuracies of different models



From the observed plot, we can infer that:
1. Accuracy of Decision Tree classifier decreases with increase in the depth of the decision tree.
2. Accuracy of Bagging and Random Forest model increases with increase in the depth of the decision tree.
3. Bagging and Random Forest model have similar performance trends.

3. Hypothesis formulated (H_1) :
Random Forest is better than Bagging

Null Hypothesis (H_0):
Random Forest is no better than Bagging

We are given an alpha level of 5% i.e 0.05.

The calculated p-value is : 0.11291033074348622
As p-value $>$ 0.05, we cannot reject the null hypothesis.

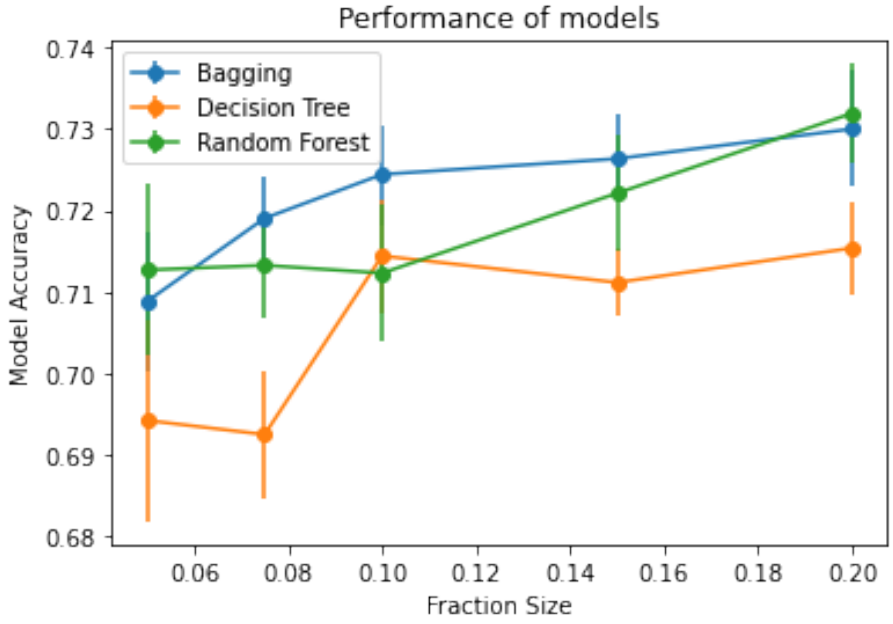
Thus, we conclude that Random Forest performs similarly to the Bagging model.

4. Q4

1. Comparing models accuracy

   Command to be entered on terminal: python cv_frac.py
   Output: Plot showing the accuracies for all the three models

2. Plot obtained after comparing the accuracies of different models



   From the observed plot, we can infer that:
   1. Accuracy of Decision Tree classifier increases with increase in the fraction of the training data used.
   2. Accuracy of Bagging model increases with increase in the fraction of the training data used.
   3. Accuracy of Random Forest model first decreases slightly and then increases with increase in the fraction of the training data used.
   4. Bagging outperforms Decision Tree model.

3. Hypothesis formulated (H_1) :
   Bagging is better than Decision Tree

   Null Hypothesis (H_0):
   Bagging is no better than Decision Tree

We are given an alpha level of 5% i.e 0.05.

The calculated p-value is : 0.004164989924206315
As p-value < 0.05, we cannot accept the null hypothesis. Hence, we accept the alternate hypothesis formulated above (H_1)

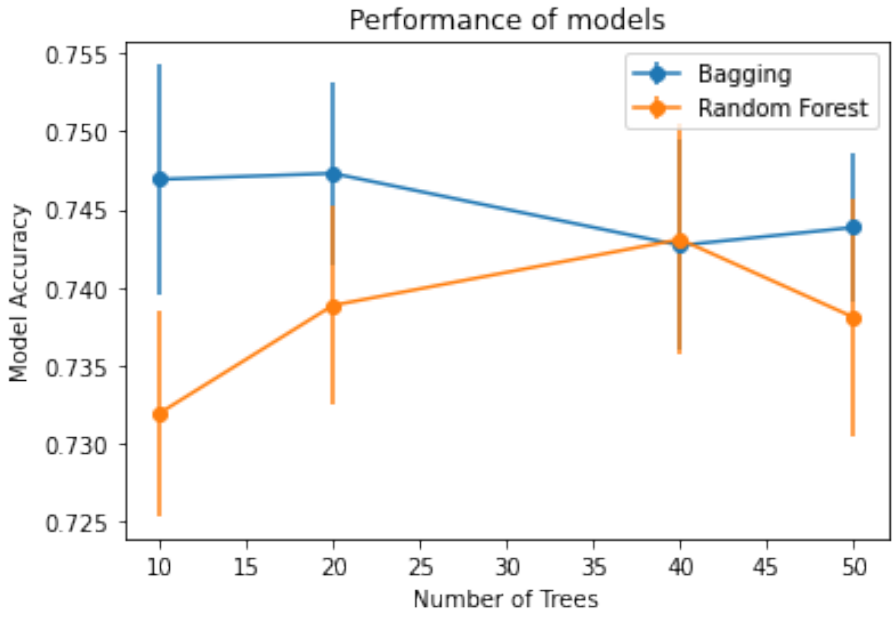
Thus, we conclude that Bagging is better than Decision Tree model.

5. Q5

1. Comparing models accuracy

   Command to be entered on terminal: python cv_numtrees.py
   Output: Plot showing the accuracies for all the three models

2. Plot obtained after comparing the accuracies of different models



From the observed plot, we can infer that:
1. Accuracy of Bagging classifier first decreases and then increases with increase in the number of trees used.
2. Accuracy of Random Forest model first increases and then decreases with increase in the number of trees used.

3. Hypothesis formulated (H_1) :
   Random Forest is better than Bagging

   Null Hypothesis (H_0):
   Random Forest is no better than Bagging

   We are given an alpha level of 5% i.e 0.05.

   The calculated p-value is : 0.10875726789387437
   As p-value > 0.05, we cannot reject the null hypothesis.

   Thus, we conclude that Random Forest performs similarly to the Bagging model.

6. Q6 Bonus Question

1. Comparing models accuracy

   ```
   Command to be entered on terminal: python neuralNetwork.py
   Training Accuracy NN: 0.73
   Testing Accuracy NN: 0.72
   ```

2. I have designed a 3 layer neural network for this problem, using sigmoid as the activation function for the output layer. The first layer takes nodes equal to the size of the features in the dataset, the second layer makes use of 50 intermediate nodes whereas the third layer uses 20 intermediate nodes.
   The model initialises weights randomly initially and then builds upon the update as it sees more data. The fit method consists of the training of the model and returns the final weight vectors learnt by the model. After that, we check the accuracy of the model on the training dataset as well as the testing dataset.
   Hyperparameters like learning rate and epochs were tested for different values and the optimal values found for those are as follows:
   Learning Rate = 0.001, Epochs = 50

   The final accuracy which I got with these values of the hyper-parameters is:
   Training Accuracy - 0.73
   Testing Accuracy - 0.72