

# CS57300 Data Mining

## Assignment 4

November 14, 2021

Vivek Gupta  
gupta690@purdue.edu

---

### *Environment:*

Experiment done on Apple Silicon M1.

Python version: **3.9.7**

Numpy version: **1.21.2**

Pandas version: **1.3.3**

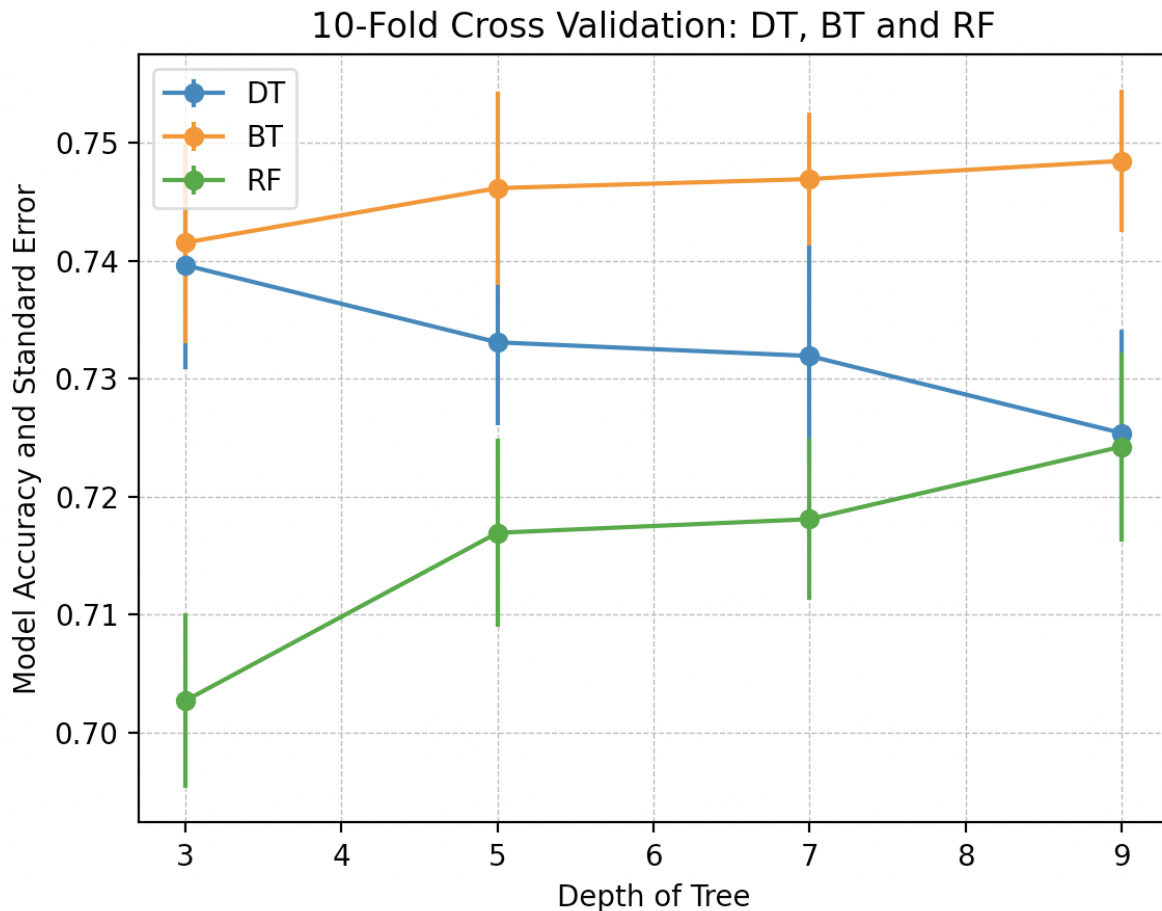
## 2. Implement Decision Trees, Bagging and Random Forests

```
(dm) → assg4 python trees.py trainingSet.csv testSet.csv 1
Training Accuracy DT: 0.77
Testing Accuracy DT: 0.72
Total Time Elapsed: 1.4101760387420654 seconds
(dm) → assg4 python trees.py trainingSet.csv testSet.csv 2
Training Accuracy BT: 0.79
Testing Accuracy BT: 0.75
Total Time Elapsed: 19.86403203010559 seconds
(dm) → assg4 python trees.py trainingSet.csv testSet.csv 3
Training Accuracy RF: 0.75
Testing Accuracy RF: 0.71
Total Time Elapsed: 12.675251960754395 seconds
```

### 3. Influence of Tree Depth on Classifier Performance

(a) *Running Time* using python multiprocessing (6 processes): 8 mins 40 seconds

- (a) DT - Decision Trees
- (b) BT - Bagging Trees
- (c) RF - Random Forests



(b) **Formulating Hypothesis:**

Null Hypothesis: Bagging Trees doesn't perform any better than Decision Trees as the depth limit of the tree increases to 7.

Alternative Hypothesis: Bagging Trees performs better than Decision Tree as the depth limit of the tree increases to 7.

Alpha Level: Set threshold to 0.05.

**Testing our formulated hypothesis:**

We will use paired t-test for the paired samples we get from Bagging Trees and Decision Trees classifier.

t-statistic is calculated as:

$$t = \frac{\bar{X}_D - \mu_0}{s_D / \sqrt{n}}$$

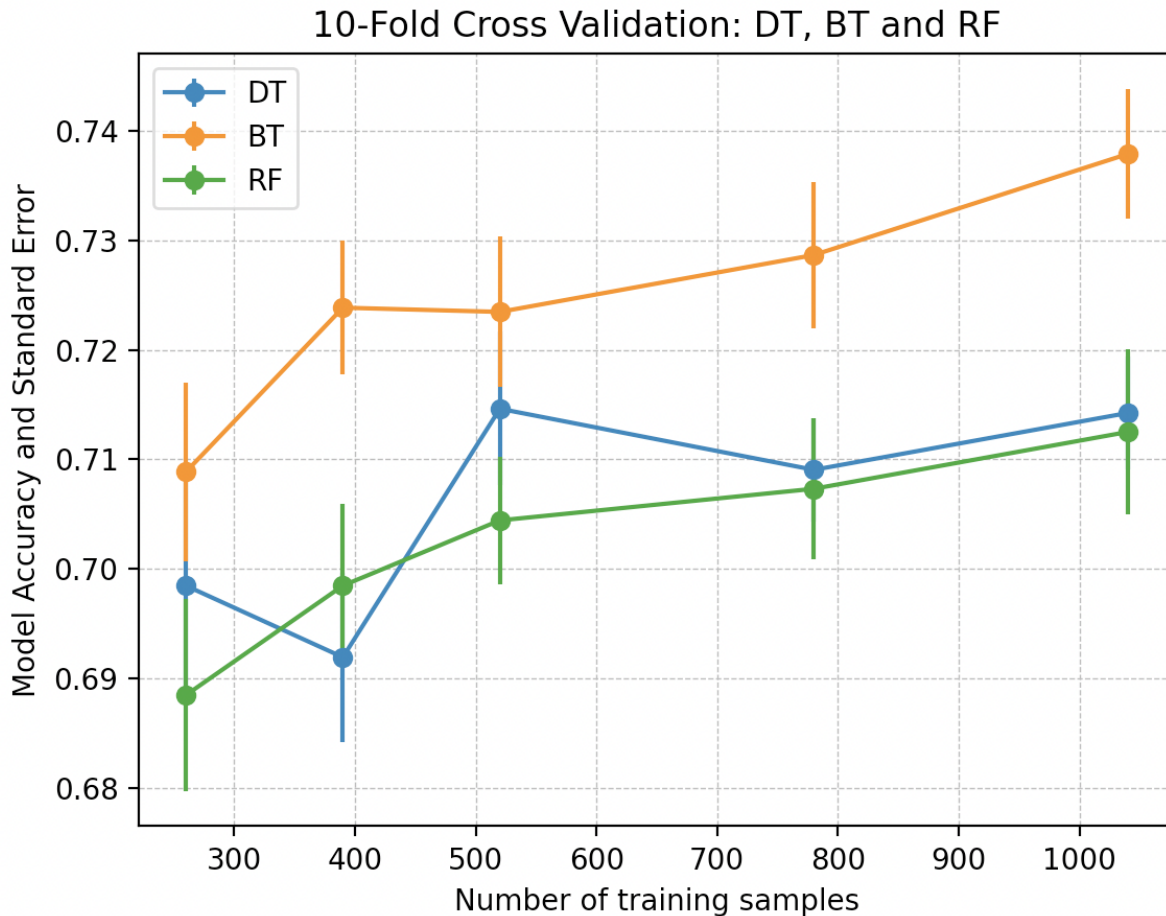
where  $\bar{X}_D$  and  $s_D$  are the average and standard deviation of the differences between all pairs at different depth of the tree. The pairs, here are accuracies of the two models DT and BT we want to compare trained using different sized samples.

Using the scipy package to compute the t-stastic and greater p-value at different depths of the trees taking the accuracies obtained at different folds, we observe p-values of 0.08, 0.06, 0.018 and 0.006 at depths 3, 5, 7 and 9 respectively. Since the p-value is less than the threshold value of 0.05 when the depth limit is 7 or 9, we REJECT the null hypothesis, and accept the alternative hypothesis to conclude that Bagging Trees model performs better than Decision Trees model as the depth of the tree changes(increases) to atleast 7 for the data in consideration. This is also evident from the graph shown above that the performance of bagging trees becomes better than decision trees in the case of dating classification as we increase the depth limit of the trees.

## 4. Compare Performance of Different Models

(a) *Running Time* using python multiprocessing (6 processes): 5 mins 20 seconds

- (a) DT - Decision Trees
- (b) BT - Bagging Trees
- (c) RF - Random Forests



(b) **Formulating Hypothesis:**

Null Hypothesis: Bagging Trees doesn't perform any better than Decision Trees.

Alternative Hypothesis: Bagging Trees performs better than Decision Trees.

Alpha Level: Set threshold to 0.05.

Testing our formulated hypothesis:

We will use paired t-test for the paired samples we get from DT and BT classifier.

t-statistic is calculated as:

$$t = \frac{\bar{X}_D - \mu_0}{s_D / \sqrt{n}}$$

where  $\bar{X}_D$  and  $s_D$  are the average and standard deviation of the differences between all pairs. The pairs, here are accuracies of the two models DT and BT we want to compare trained using different sized samples.

Using the scipy package to compute the t-stastic and greater p-value, we observe a p-value of less than 0.05 ( $p \approx 0.002$ ). Since the p-value is less than the threshold value of 0.05, we REJECT the null hypothesis, and accept the alternative hypothesis to conclude that Bagging Trees model performs better than Decision Trees

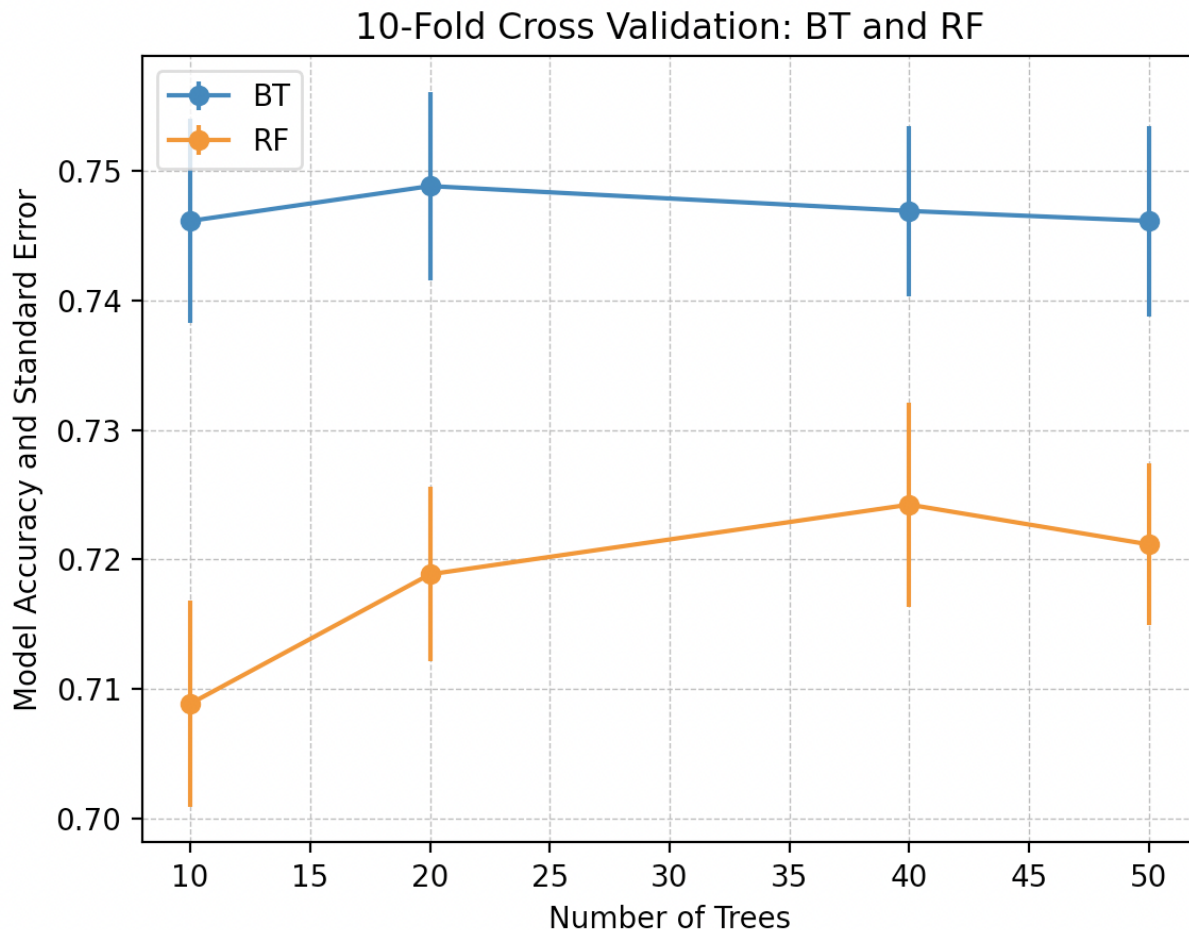
model for the data in consideration. This is also evident from the graph shown above that the performance of bagging trees is better than decision trees in the case of dating classification.

## 5. The Influence of Number of Trees on Classifier Performance

(a) *Running Time* using python multiprocessing (6 processes): 10 mins 18 seconds

(a) BT - Bagging Trees

(b) RF - Random Forests



(b) **Formulating Hypothesis:**

Null Hypothesis: Bagging Trees doesn't perform any better than Random Forests.

Alternative Hypothesis: Bagging Trees performs better than Random Forests.

Alpha Level: Set threshold to 0.05.

Testing our formulated hypothesis:

We will use paired t-test for the paired samples we get from BT and RF classifier.

t-statistic is calculated as:

$$t = \frac{\bar{X}_D - \mu_0}{s_D / \sqrt{n}}$$

where  $\bar{X}_D$  and  $s_D$  are the average and standard deviation of the differences between all pairs. The pairs, here are accuracies of the two models BT and RF we want to compare trained using different sized samples.

Using the scipy package to compute the t-statistic and greater p-value, we observe a p-value of less than 0.05 ( $p \approx 0.0015$ ). Since the p-value is less than the threshold value of 0.05, we REJECT the null hypothesis, and accept the alternative hypothesis to conclude that the Bagging Trees model performs better than Random Forests for the data in consideration. This is also evident from the graph shown above that the performance of bagging trees is better than random forests in the case of dating classification.

## Bonus Question

Implemented Neural Networks with two hidden layers of size 10 and 5 respectively. Code can be found inside *neural\_net.py* file and can be executed using simple *python neural\_net.py trainingSet.csv testSet.csv* script. 3000 epochs were used for training. Hyperparameter tuning was done by keeping aside some part of training set to be used as validation set. A grid search approach was used to select the best learning rate, batch size and number of epochs. Learning rate was set to 0.3. Loss was seen converging on both train and test dataset. Data was processed in batch size of 64. A random seed of 0 for numpy random has been set to initialize the weights for reproducibility purpose. The training accuracy rounded to two decimal places is 0.81 and test accuracy is 0.76.

```
(dm) → assg4 python neural_net.py trainingSet.csv testSet.csv
Training Accuracy NN: 0.8094230769230769
Testing Accuracy NN: 0.7576923076923077
Total Time Elapsed: 18.2378408908844 seconds
(dm) → assg4 █
```