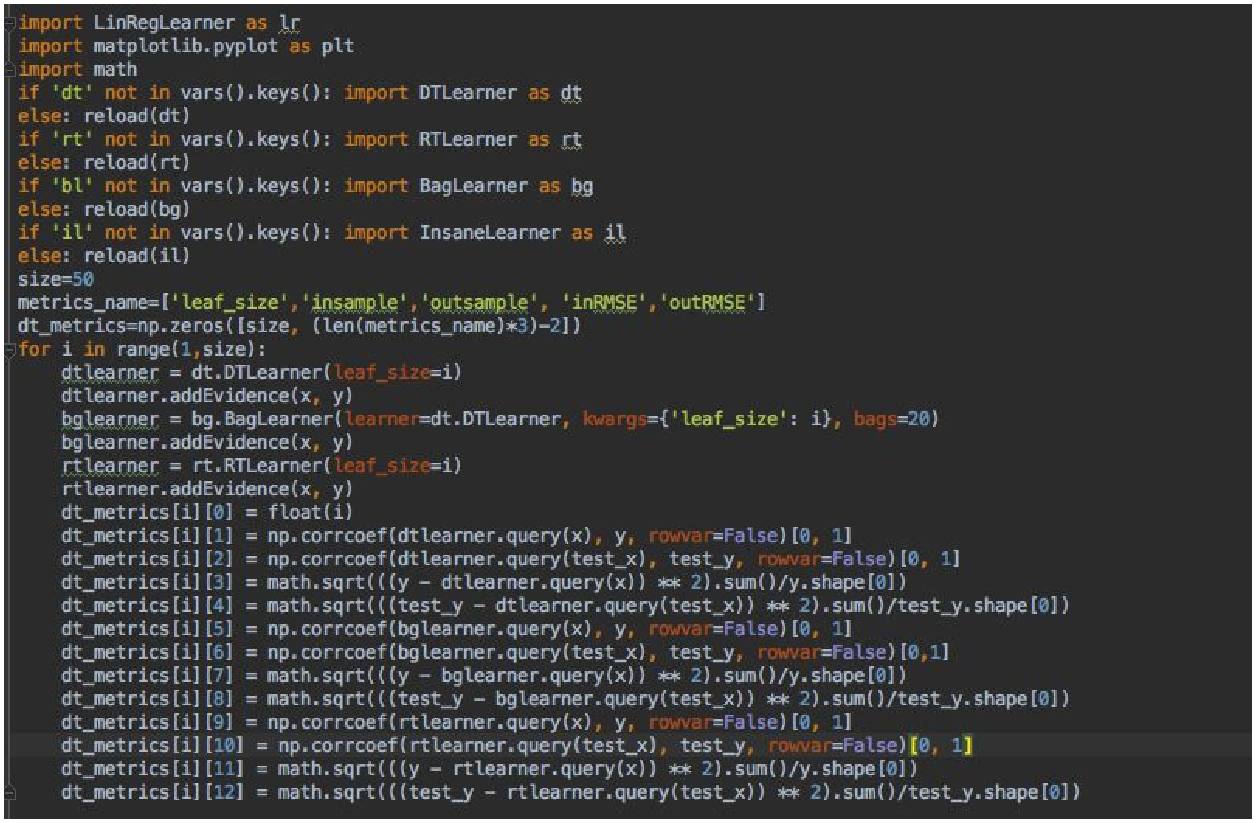
Experiment Report

1.Experiment Design:

To compare the performance of the DTLearner, the BagLearner and the RTLearner over the leaf size. I run a loop with the range of leaf sizes. In each loop, three learners are trained and tested while testing results are recorded with their leaf sizes.

2.Experiment Python Script Implementation:



```
import LinRegLearner as lr
import matplotlib.pyplot as plt
import math
if 'dt' not in vars().keys(): import DTLearner as dt
else: reload(dt)
if 'rt' not in vars().keys(): import RTLearner as rt
else: reload(rt)
if 'bl' not in vars().keys(): import BagLearner as bg
else: reload(bg)
if 'il' not in vars().keys(): import InsaneLearner as il
else: reload(il)
size=50
metrics_name=['leaf_size','insample','outsample', 'inRMSE','outRMSE']
dt_metrics=np.zeros([size, (len(metrics_name)*3)-2])
for i in range(1,size):
    dtlearner = dt.DTLearner(leaf_size=i)
    dtlearner.addEvidence(x, y)
    bglearner = bg.BagLearner(learner=dt.DTLearner, kwargs={'leaf_size': i}, bags=20)
    bglearner.addEvidence(x, y)
    rtlearner = rt.RTLearner(leaf_size=i)
    rtlearner.addEvidence(x, y)
    dt_metrics[i][0] = float(i)
    dt_metrics[i][1] = np.corrcoef(dtlearner.query(x), y, rowvar=False)[0, 1]
    dt_metrics[i][2] = np.corrcoef(dtlearner.query(test_x), test_y, rowvar=False)[0, 1]
    dt_metrics[i][3] = math.sqrt(((y - dtlearner.query(x)) ** 2).sum()/y.shape[0])
    dt_metrics[i][4] = math.sqrt(((test_y - dtlearner.query(test_x)) ** 2).sum()/test_y.shape[0])
    dt_metrics[i][5] = np.corrcoef(bglearner.query(x), y, rowvar=False)[0, 1]
    dt_metrics[i][6] = np.corrcoef(bglearner.query(test_x), test_y, rowvar=False)[0,1]
    dt_metrics[i][7] = math.sqrt(((y - bglearner.query(x)) ** 2).sum()/y.shape[0])
    dt_metrics[i][8] = math.sqrt(((test_y - bglearner.query(test_x)) ** 2).sum()/test_y.shape[0])
    dt_metrics[i][9] = np.corrcoef(rtlearner.query(x), y, rowvar=False)[0, 1]
    dt_metrics[i][10] = np.corrcoef(rtlearner.query(test_x), test_y, rowvar=False)[0, 1]
    dt_metrics[i][11] = math.sqrt(((y - rtlearner.query(x)) ** 2).sum()/y.shape[0])
    dt_metrics[i][12] = math.sqrt(((test_y - rtlearner.query(test_x)) ** 2).sum()/test_y.shape[0])
```

In the script, I created objects below:

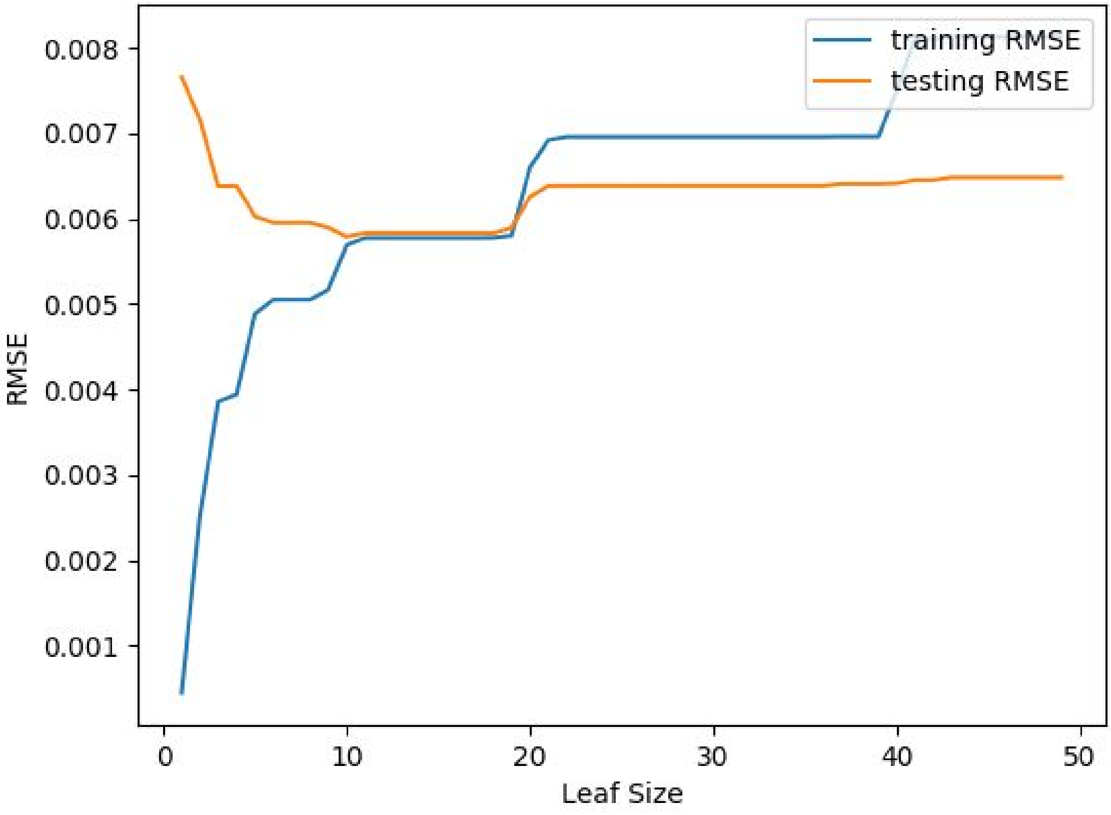
- The range of leaf size: 0 to 50 (looping through the range to record result for each leaf size)
- An ndarray object with 50 rows and 13 columns:
  [column 0: leaf size;
  column 1: correlation of training data for DTLearner
  column 2: correlation of testing data for DTLearner
  column 3: RMSE of training data for DTLearner
  column 4: RMSE of testing data for DTLearner
  column 5: correlation of training data for BagLearner
  column 6: correlation of testing data for BagLearner
  column 7: RMSE of training data for BagLearner
  column 8: RMSE of testing data for BagLearner
  column 9: correlation of training data for RTLearner

column 10: correlation of training data for RTLearner
column 11: RMSE of testing data for RTLearner
column 12: RMSE of testing data for RTLearner]

- An DTLearner object (recreated for each loop)
- An BagLearner object (recreated for each loop)
- An RTLearner object (recreated for each loop)

Question 1:Does overfitting occur with respect to leaf_size? Consider the dataset istanbul.csv with DTLearner. For which values of leaf_size does overfitting occur? Use RMSE as your metric for assessing overfitting. Support your assertion with graphs/charts. (Don't use bagging).

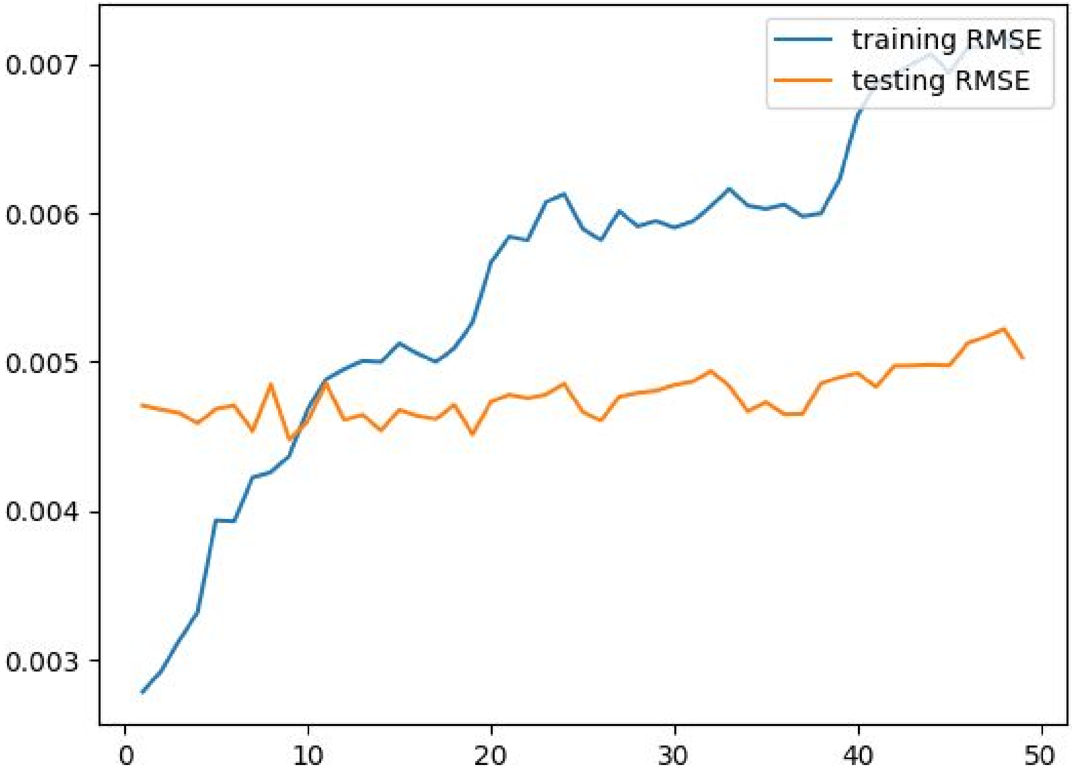
I plot the the column 0, column 3 and column 4 to compare the performance of DTLeaners on training RMSE and testing RMSE over different leaf sizes. A significant difference between the training RMSE and the testing RMSE indicates overfitting.



From the figure, we can tell that before leaf size of 10, there is obvious overfitting problem in the DTLeaners. As the leaf size increases from 1 to 10, the overfitting decreases. However, after leaf size increases over 20, DTLeaners appears poor performance in both the training and the testing data.

Question 2: Can bagging reduce or eliminate overfitting with respect to leaf_size? Again consider the dataset istanbul.csv with DTLearner. To investigate this choose a fixed number of bags to use and vary leaf_size to evaluate. Provide charts and/or tables to validate your conclusions.

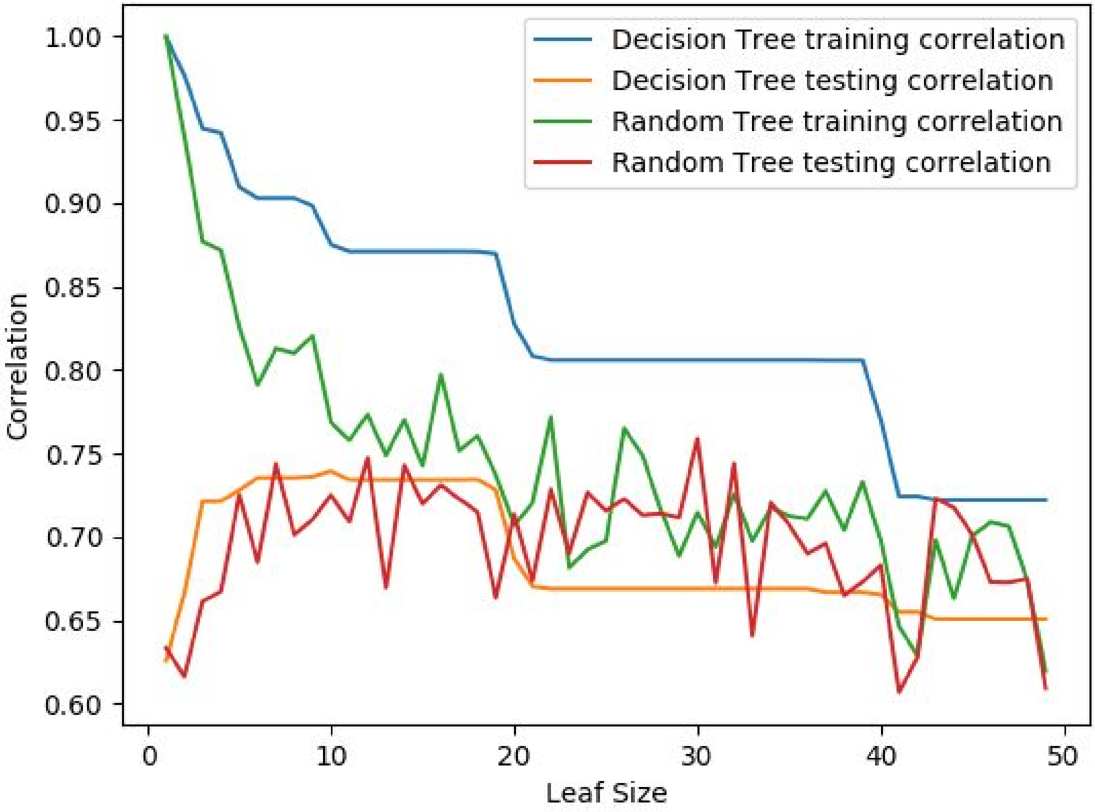
I plot the the column 0, column 7 and column 8 to compare the performance of BagLeaners on training RMSE and testing RMSE over different leaf sizes. A significant difference between the training RMSE and the testing RMSE indicates overfitting.



From the figure, we can tell that before leaf size of 10, there is still overfitting problem in the BagLeaners. However, the biggest difference between training RMSE and testing RMSE, about 0.002, is much smaller than what appears in the DTLearner's plot, about 0.007. So we can conclude that Bagging can reduce the overfitting problem but not able to eliminate it.

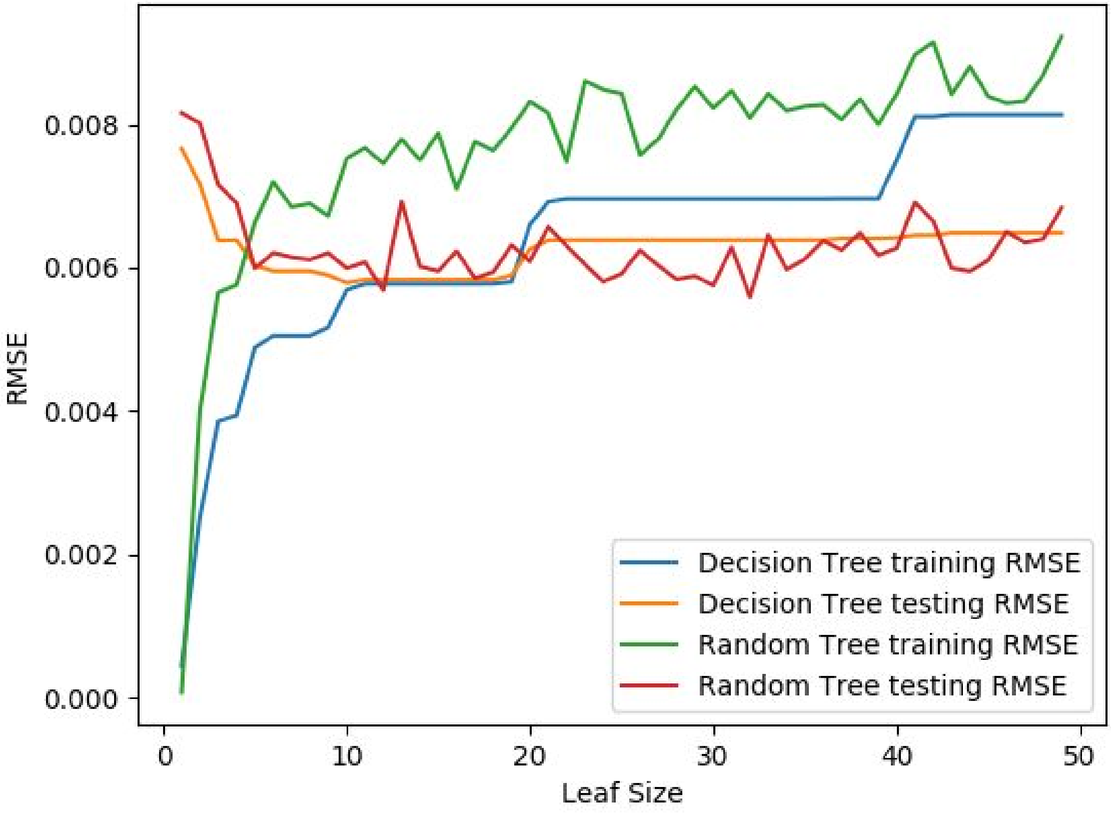
Question 3:Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other?

I plot the the column 0, column 1, column 2, column 9, column 10 to compare the performance of DTLearners and RTLearners on training correlation and testing correlation over different leaf sizes. A high correlation indicate good performance and a significant difference between the training correlation and the testing correlation indicates overfitting.



From the figure, we can tell that DTLearner performs better the RTLearner generally because of the higher correlation. However, RTLeaner has fewer overfitting problem than the DTLearner since it has a smaller gap in training and testing results. Besides that, DTLearner appears to be much more stable in terms of the experiment results.

I plot the the column 0, column 3, column 4, column 11, column 12 to compare the performance of DTLearners and RTLearners on training RMSE and testing RMSE over different leaf sizes. A significant difference between the training RMSE and the testing RMSE indicates overfitting.



From the figure, we can tell that DTLearner performs better the RTLearner generally because of the lower correlation. However, RTLeaner has fewer overfitting problem than the DTLearner since it has a smaller gap in training and testing results(before leaf size reach 10). Besides that, DTLearner appears to be much more stable in terms of the experiment results.