1. Training and testing set has been used to train. The result has been summarized as below:

|  |  |  |
| --- | --- | --- |
| n\_estimators | Training Set  Accuracy score | Testing Set  Accuracy Score |
| 1 |  |  |
| 20 |  |  |
| 100 |  |  |

We can see the trend easily for both training and testing data set. When *n\_estimators=1*, the accuracy score is the lowest as it just generates 1 tree from large data set and it has low reliability. The accuracy score starts to increase when *n\_estimators=20* as more trees were generated as more attributes were considered. However, when *n\_estimators=100* the score goes down slightly as there are too much trees and it may make the outliers become significant.

Apparently, testing data set gets higher score than training because testing contains more data. Although it contains outliers, since random forest won’t overfit, it won’t influence the accuracy score.

Random forest doesn’t have overfit problem; however, it ignores outliers so the result may be not reliable.

1. The accuracy score for original optimal tree and each tree has been summarized as below:

|  |  |
| --- | --- |
|  | Training Set  Accuracy score |
| Optimal n\_estimators =12 |  |
| Lowest |  |
| Highest |  |



I ran the script for 10k by using a for loop and set 1 to 40 as my *n\_estimators.* Since I have made a few test cases the largest value of accuracy score should be around 0.84 and I have printed out all the values who have their accuracy score larger than 0.84 (*Image 1*). All result is smaller than 30, it means that when the *n\_estimators* is larger than 30 the significance of the outliers in the component trees will be magnified causing errors in the results. After taking the average, the optimal should be 12.

(c)