Q 6

Part a. In this part, there is a comparison between the scikit-learn's version of random forest and the random forest developed from scratch.

For execution of max-features from 1 to 40

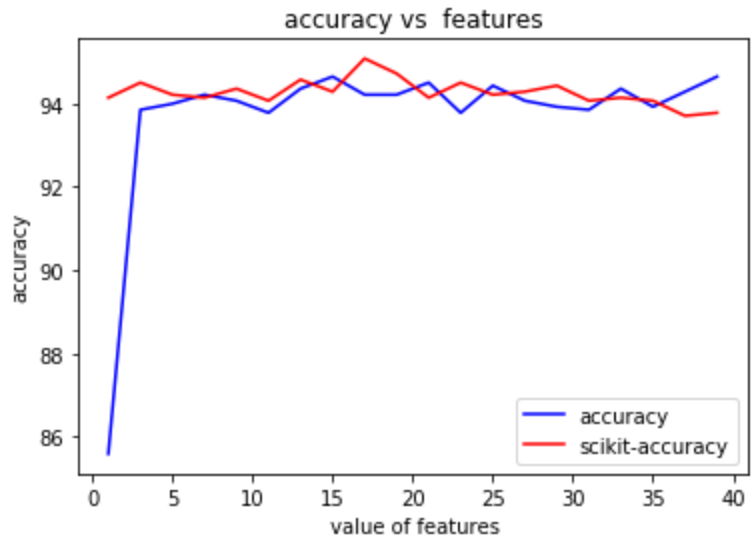
Average time to execute scikit learn algo is **0.2116417407989502**

Average time to execute own algo is **20.463728165626527**

For various values of max-features, I got the following accuracy and time for execution for tree_size as **20**

Max_features:	**Testing accuracy Scikit**	**Time taken Scikit**	**Testing accuracy -own**	**Time taken-own**
5	94.20709630	0.0594074	93.9174511223	6.32674407
15	94.279507	0.1386353	94.35191889	16.05
29	94.4243301	0.299488	94.3519188	29.32376
35	94.062273714	0.4572	94.06227371	35.106

This graph is obtained by keeping n_estimators=20, and varying the feature list



For varying number of trees, we get the accuracy and time as

For execution of number of trees from 20 to 100

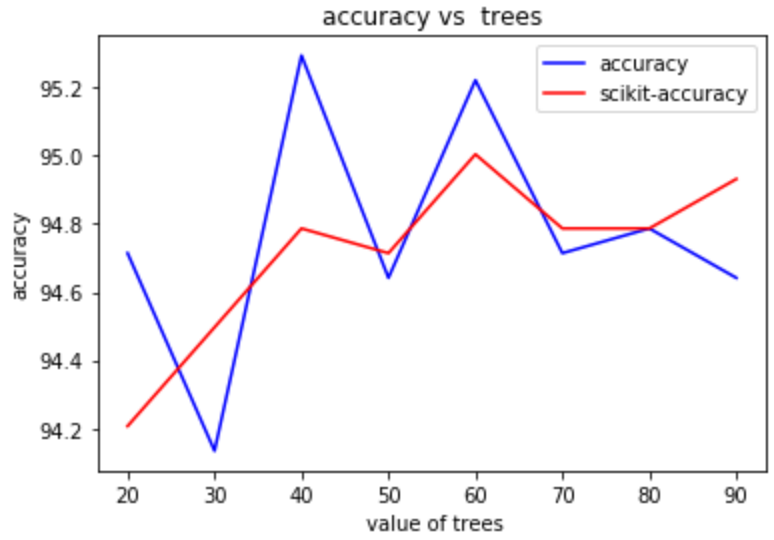
Average time to execute scikit learn algo is **0.2474014163017273**

Average time to execute own algo is **27.15326488018036**

For various values of number of trees, I got the following accuracy and time for execution for max_features as **sqrt(max_features)**

Number_trees:	**Testing accuracy Scikit**	**Time taken Scikit**	**Testing accuracy -own**	**Time taken-own**
20	94.2070963070	0.0739660263	94.71397538	9.58505296707
50	94.713975380	0.2072293758	94.641564083	22.99193739891
60	94.7863866763	0.28723287582	94.713975380	32.2607173919
80	94.93120926	0.4869375228	94.641564083	50.291978120

This graph is obtained by keeping max_features=sqrt, and varying the number of trees



Part b.) This sub-question plots the value of sensitivity of the max-features to the random forest. To get the sensitivity, we plot the testing error that we obtain when varying the max_features from 1 to 57. This graph has a lot of jitters and the least value of error is obtained around the range of **sqrt(max_features)** for this dataset from my algorithm. The error keeps on increasing and decreasing around the range of accuracy of **93-94.**

The testing-error was found out by taking a majority vote for each point after being classified by the random forest. This majority vote was taken as the output which was later compared to the label to obtain the accuracy

This graph is plotted with taking the **sample size of 1500, n_estimators(number of trees) as 20** .

Part c)

OOB error: In order to find the OOB-error , the following steps were followed:

1. The data was first split into training and testing data
2. The training data was then given to create the random forest for all values of the parameter ‘m’.
3. After the forest was formed for a particular value of m, the OOB error is calculated by
    a. Iterating over the entire training data points and classifying it with the trees to which it did not act as a training point.
    b. The results were accumulated over all such trees and the class was determined by taking a majority vote for a class.
    c. This class was compared with the label to find its accuracy
    d. This was done across all the points and the error was found out.

The testing error was found out similar to the previous sub-part

For this, the **n_estimators(number of trees) were fixed as 20 and sample of data was 1000 points.** This iterated over all the feature size (iterated by 2)