Part 1 : Digit Recognition - KNN and Boosting

a) Status and stopping point-

In the first dataset for optical recognition of digits, the maximum accuracy percentage is more than 98% which can be considered as an acceptable value. In case of KNN, KNeighborsClassifier is used. Based on the varying value of neighbors the prediction accuracy varies. The best results are coming when number of neighbors are 8. The accuracy was almost 98% for varying number of neighbors. For Boosting I did a comparison between results using Decision tree and Ada Booster Classifier, which uses Decision Tree as well.The results were very helpful as for lower depth of tree, boosting algorithm really improved accuracy. For example when depth was 2 the boosting improved accuracy from 30 to 80%. For higher values the accuracy is about the same.

b) Additional functions and analysis:

KNN -

- Based on the number of neighbors, the accuracy was almost always 97%+
- Even for lesser neighbors accuracy was 97%

Boosting -

1. Boosting really improves accuracy when max depth for decision tree is less than 6.
2. For higher values using boosting is not that helpful as results are almost similar to the ones without boosting.

c) Challenging part was to decide on number of neighbors for KNN, as having large number of neighbors causes program to run for longer time. Also understanding when to use boosting was crucial as for higher values of depth its not useful to use boosting.

Part 2: Amazon Data Set - KNN and Boosting

a) Status and stopping point -

For amazon data set increasing number of neighbors didn't had much effect on accuracy. When neighbors are 3 that's when I got the maximum accuracy. Otherwise it was always around 57-58%. Whereas for Boosting when I used Decision trees with and without boosting; there was not much of a difference between the two. The results with boosting and without it were the same.

b) Additional functions and analysis -

After analysing data I realized that it's necessary to clean the data as data set had rows with no reviews. To get rid of those rows I used dropna() method on data frame.
Then review itself had some html markup words. To counter that I used Beautiful soup. And then in order to get rid of stop words such as 'is','and' I used nltk package.

If I don’t perform all these operations the accuracy of the code is lower than 54%.

c) Challenges -

Amazon data set is really huge and playing with such huge data was challenging for me. Testing large data set for KNN was a challenge as well. For some reason, KNN classifier took too much of time to predict the result. That prevented me from testing with large data set for KNN.