Assignment 2

Please look at the Jupyter notebook for the assignment 2 answers.

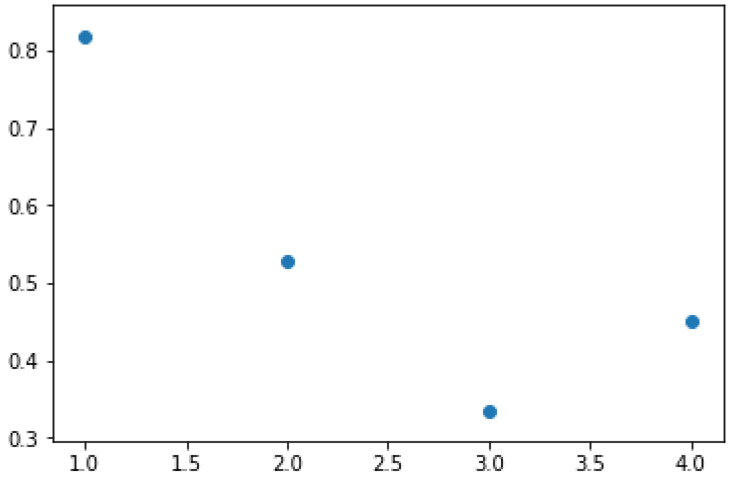
> LINK to all files

Question 1

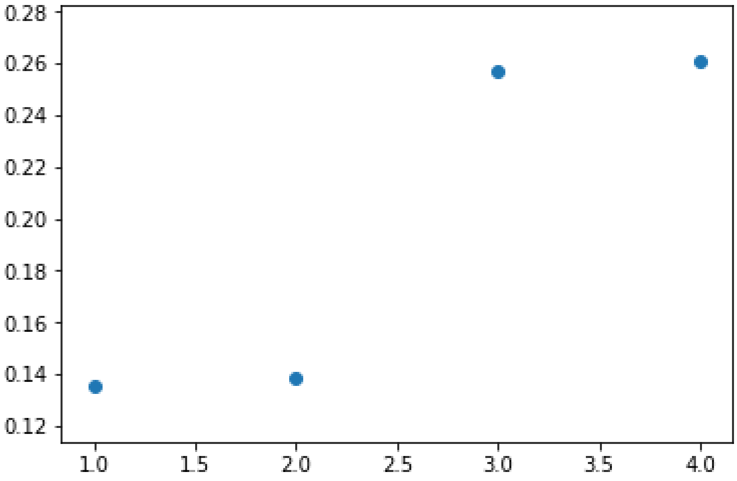
a) and b) link, link to the jupyter notebook and to the paralleled random forest implementation, resp.

c) Graph showing the performance increase with increase in the number of jobs.

- Training



- Prediction



d) We see that the accuracy is actually decreasing as the data is not complex enough that we would like to reduce any bias or variance from the model obtained from Decision Trees. Also in Random Forests we try to form the constituent trees by randomly selecting the features, in this case this might have affected the accuracy of the model negatively. Because the model Decision Tree was learning wasn't biased or having a high variance.

```
rf_acc: 0.35555555555555557
dt_acc: 0.9111111111111111
```

e) After using the nested cross validation we were getting the following as the optimal depth.

```
optimal num_of_trees: 50 | with accuracy: 0.3687356321839081
```

The reason for a selection of 50 number of trees might be selection of *bad* features while testing for `num_of_trees = 1`.

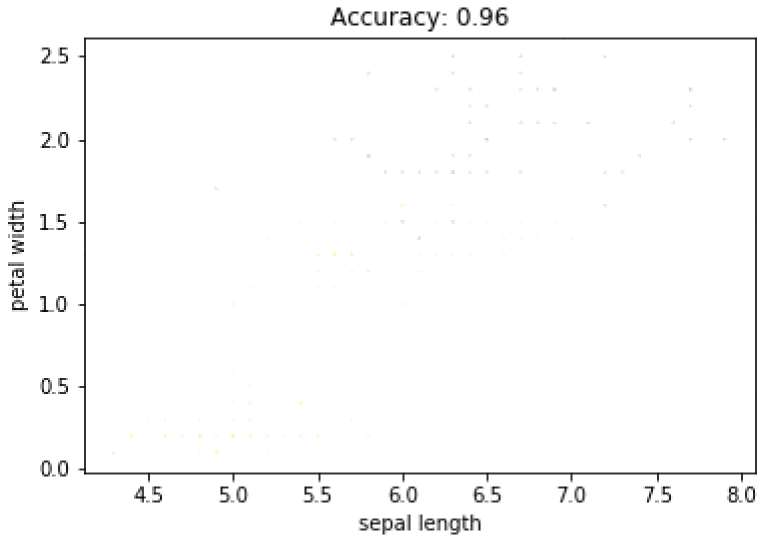
We observe bad performance in this case as we are randomly selecting only 2 of the 4 features available and those maybe bad features for the classification task

Question 2

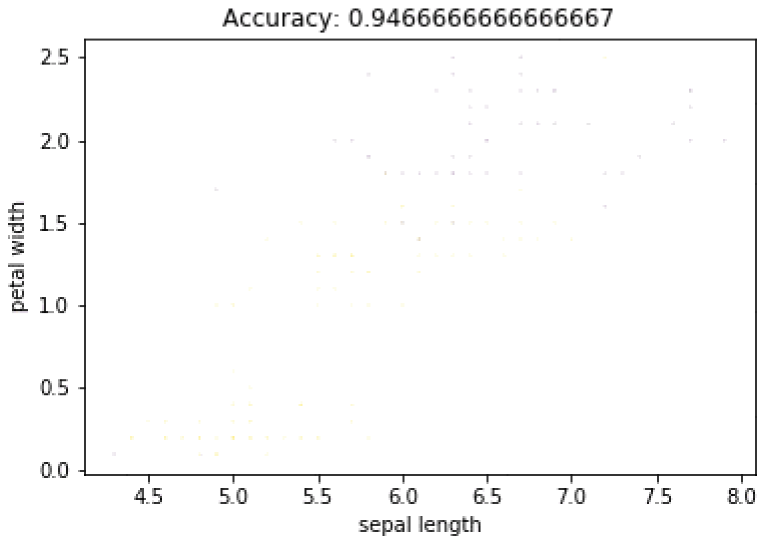
Submitted.

Question 3

a) 4 Iteration of Adaboost weights on running on actual data-set.



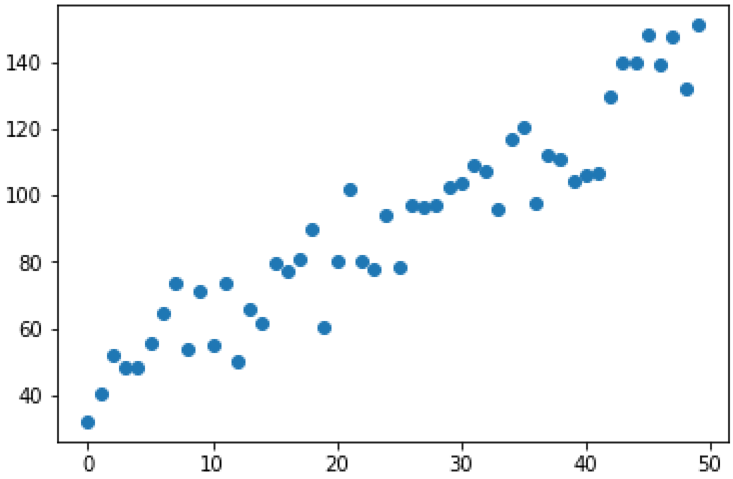
b) 8 Iteration of Adaboost weights on running on noisy data-set.



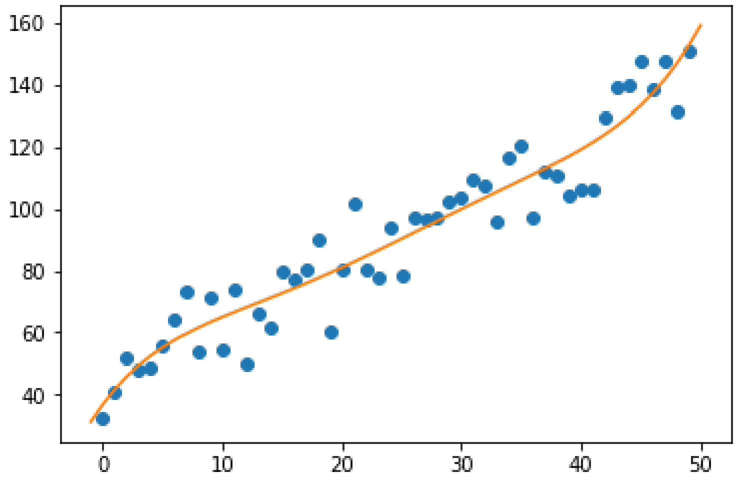
We see that the Adaboost is trying to increase the weight-age of the noisy miss-classified labels. We infer that Adaboost is really sensitive to outliers. If a human was to learn the labelling of the data-set, it would be apparent that these labels are noisy data and therefore instead ignore these misguiding examples.

Question 4

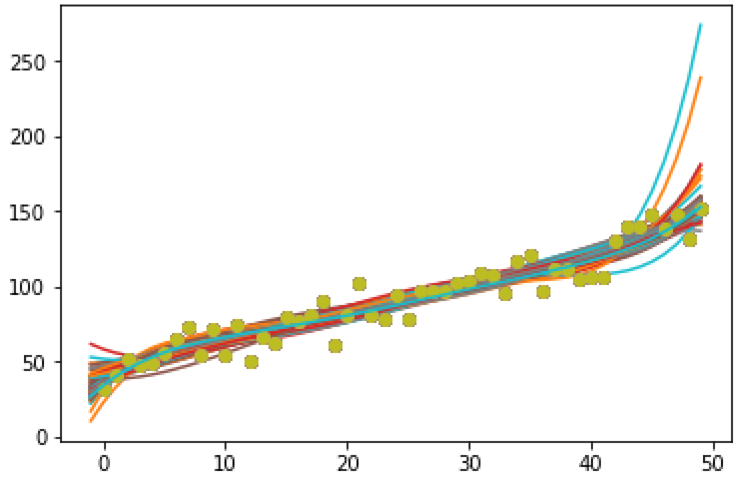
a) Generated 50 points where `y = mx + c + random_noise`. The `random_noise` I used was Gaussian. ▼



b) Fitting a 5 degree polynomial to the data provided. ▼



c) Fitting 100 20 degree polynomials to the data. ▼

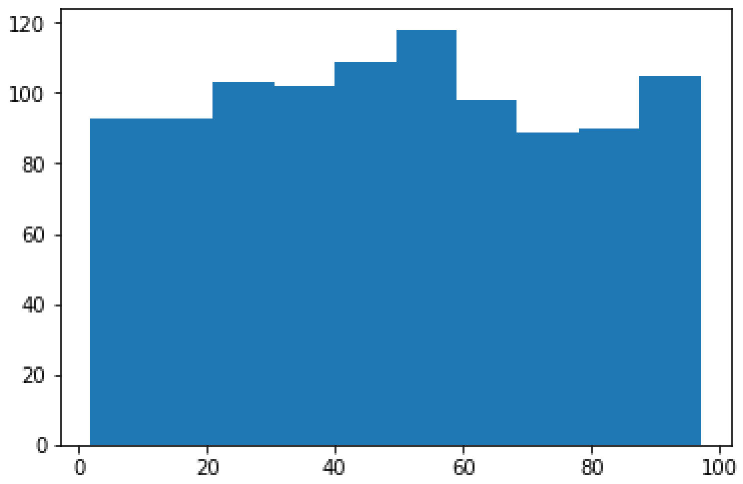


The using of the concept of bagging in this case leads to a model that seems to be less prone to variance due to the reason that the collective polynomials lead to an averaging effect on the bagged model.

Question 5

a) Link to the file containing the random number generator implemented by me.

b) Yes, I am able to get a nearly uniform distribution for `1000` numbers in the case of `N = 100`. Look below for a histogram of the distribution.


120
100
80
60
40
20
0
0
20
40
60
80
100