AdaBoost

1. The AdaBoost Algorithm is implemented on total of 3 datasets.
2. Letter-Recognition Dataset
    a. The test-accuracy was achieved around 94% for this dataset computed for 50 rounds and depth of Decision Tree as 1 depth
    b. The test-error on 1st round and 50th round is 0.2325 and 0.2252 respectively.
    c. The model was also trained for 50 rounds and depths ranging from 1 to 10. The accuracy for the depth 1 was around 94% and Depth 10 round was around 96%
3. German-numeric Dataset
    a. The test-accuracy was achieved around 69% for this dataset computed for 50 rounds Decision Tree as 1 depth
    b. The test-error on 1st round and 50th round is 0.230 and 0.224 respectively.
    c. The model was also trained for 50 rounds and depths ranging from 1 to 10. The accuracy for the 1st round was around 69% and 100th round was around 70%
4. Spambase Dataset
    a. The test-accuracy was achieved around 93% for this dataset computed for 50 rounds Decision Tree as 1 depth
    b. The test-error on 1st round and 50th round is 0.233 and 0.226 respectively.
    c. The model was also trained for 50 rounds and depths ranging from 1 to 10. The accuracy for the 1st round was around 93% and 100th round was around 95.5%

- The Accuracy is and the error is the best for the Spambase dataset and lowest for the German numeric Dataset
- The models accuracy increases as the depth of decision tree increasese, but it will decrease after increasing the depth >15 and can cause overfit issue

Part 2)

Bagging:

The aim of the bagging process is to lower the variance and prevent overfitting in circumstances where perturbation in the data can cause substantial variation in the individual model trained.

Build a Bagging Classifier in Python:

We have created a class to capture the functionality of bagging ensemble method.

__init__(self, no_ele = 20) initializer function executes inevitably when class instance is declared. The model components are declared, and it takes no of elements as an input which defines no of models to be used in the ensemble method.

bootstrap_sampling(self, df) function creates bootstrap samples.

fit (self, train_X, train_y) function trains(train_X) ensemble using given predictors(train_y). Then, the function evaluates the out-of-bag samples (extra samples) to compute the standard error on the evaluation matrix.

predict (self, X) function gives predicted values from the fitted ensemble applying the input.

Dataset summary:

For all the given dataset, I noticed that the model achieves its top accuracy in the first element itself and it remains constant throughout the 20 elements. This could be because the dataset might not have large variance in it. Therefore, the model achieved its accuracy on the first element itself. For such results, the plot doesn't make sense as it would be just a single point rather than the positive linear points.

If the given dataset had large variance, the model would have improved accuracy and reduced standard error on each successive elements.

Dataset: 1-

Accuracy score: 0.98

Dataset: 2

Accuracy score: 0.71

Dataset: 3

Accuracy score: 0.90

Conclusion: Bagging works better when compared to Adaboost. The Highest score was achieved from the 1st Dataset and of 98% by the bagging classifier. However, the accuracy for the 2nd dataset is nearly the same for both the classifiers, it seems more data preprocessing must be required in order to achieve more accuracy. The 3rd Dataset works better in case of Adaboost compared to bagging classifier.