Machine Learning
Homework 2 : Ensemble Methods
(due Noon Jan 27)

Instructions

1. In case you are unfamiliar with the Python data ecosystem (NumPy, Pandas), you are recommended to study the first four chapters of the Python data science handbook. A doubt clearing session would be organised in case you have any difficulties in the data science ecosystem.
2. The deadline for full score is Noon Jan 27. You can get 50% credit for late submission (Noon Jan 29).
3. Total marks = 20
4. You have to type the assignment using a word processing engine, create a pdf and upload on the form. Please note that only pdf files will be accepted.
5. All code/Jupyter notebooks must be put up as **secret gists** and linked in the created pdf submission. Again, only secret gists. Not public ones.
6. Any instances of cheating/plagiarism will not be tolerated at all.
7. Cite all the pertinent references in IEEE format.
8. The least count of grading would be 0.5 marks.

1. (a) Extend the decision tree you created in first assignment to now implement random forests. **[2 marks]**
   (b) Now create a parallelised version of the above where different trees can be built in parallel. You can use this article as a reference. **[1 mark]**
   (c) Using simulations compare the performance of serial and parallel versions of your implementation, i.e. as you increase parallelisation do you get performance benefits? **[1 mark]**
   (d) Now use your random forest implementation on the IRIS dataset. Like before, the first 70% of the data should be used for trainining and 30% for test purposes. Compare the accuracy v/s decision tree for the following parameters: i) # of estimators = 20, ii) # feature choices to use = $\sqrt{(n)}$ where n indicates the total number of features available. **[1 mark]**
   (e) Now use 5-fold cross-validation on the IRIS dataset. Using nested cross-validation find the optimum number of estimators in the set of [1, 2, 5, 10, 50, 100] estimators. **[1 mark]**
2. Submit your score on Kaggle (IITGN internal competition) for the blue book for bulldozers competition using a) decision tree and b) random forests. You are free to use scikit-learn for this competition. **[5 marks]**
3. (a) Re-encode the IRIS dataset class as 'setosa' and 'not-setosa'. Now apply ADABoost (on depth-1 decision trees as weak learner) while considering only 'sepal length' and 'petal width' as the features. Train on 100% of the data and create a Matplotlib animation for the first 8 iterations of ADAboost. The title of the plot should show the iteration number and the accuracy on the train set. The plot should color the 'setosa' and 'not-setosa' points differently and their marker size should correspond to their weights as per the ADABoost algortihm. **[2 marks]**
   (b) Now, add some 'noise' to the above dataset, i.e. add some 'setosa' points to the vicinity of the 'not-setosa' points and vice-versa. A human should be able to recorgnise these as outliers. Run the animation on this dataset and comment on the behaviour of ADABoost. **[2 marks]**

4. (a) Create a dataset of 50 points where y = mx + c + random noise. Ensure that the noise is significant compared to the trend in the data. Plot this data using Matplotlib. **[1 mark]**

   (b) Using this article as a reference, plot a 5 degree fit for the data **[1 mark]**

   (c) Illustrate the concept of bagging on this dataset. For n=100 rounds of bagging, fit 5 degree curve and comment on the average (bagged) regressor. Is less prone to variance **[1 mark]**

5. (a) Write a program (without using any builtin function) to randomly choose between a list of N numbers [1, ..., N]. You are free to use any algorithm for the same. The goal of this exercise is to enable you to understand how computers would randomly choose from a set of choices. You would used such a routine to implement bagging.**[1 mark]**

   (b) Invoke the above program 1000 times where N = 100. Plot the distribution of the generated random numbers. Do you get a roughly uniform distribution?**[1 mark]**

Some useful references for the homework:

1. Scikit-learn page on decision trees
2. Scikit-learn page on ensemble methods