**MTH655: Assignment 2**

*Due: November 5, 9pm*

*Please submit your solutions through D2L. Penalty for late submission is 10 marks per day.*

**Problem 1: Lasso regression and ensemble methods (bagging)**

**(50 points)**

In this problem you will need to predict house prices using many lasso regressions. The data (house prices and features) are in file Houseprice\_data\_scaled.csv

You need:

* Create 100 lasso models and train each of them on 900 data points randomly sampled from the original 1800 data points in the train set. In particular, each model should be trained (estimated) on a different sample of 900 observation. To sample randomly, you might want to use **random** package. For lasso models use the penalty parameter alpha=0.05. You might want to fix random number generator as **random.seed(i)** where i is the model number.
* Your predictions should be based on a set of models (bagging). First, you use the train set to obtain the output of 1 (first) lasso model as the first prediction. Second, you average the output of 1 (first) and 2 (second) lasso models, thereby getting second prediction. Third, you average the output of 1 (first), 2 (second), and 3 (third) lasso models, thereby getting third prediction. Etc. For example, first lasso model predicts (normalized) house price as 0.1 and the second lasso predicts the house price to be 0.2. Then the first prediction is 0.1 and the second prediction is (0.1+0.2)/2 = 0.15.
* For each prediction in the previous step you should find the error of the prediction (standard deviation of “target value **y\_train** minus the prediction”) and plot the error as a function of number of models that you averaged to get the prediction. The goal is to show that the error *usually* decreases as the number of models used in each prediction increases (bagging).

The start of your Python code is in file **BaggingStart.html**

**To submit:**

1. Well-documented Python code that implements the above bagging approach with lasso regressions. The code must be structured similarly to what we did in class (for example, the block that plots the results must be present in your code as well) (**45 points**)
2. Your explanation of what bagging is. (**5 points**)

**Problem 2: Decision tree to predict house prices**

**(50 points)**

In this problem you need to implement the Decision tree approach to predict house prices (see Section 4.5 and also the slides for the description of the approach). The data (house prices and features) are in file Data\_DecisionTree.xlsx

You need:

* Evaluate the expected Mean-Squared Error for each threshold value of house Quality and Area; find the threshold that maximizes expected MSE for each feature (Quality and Area). Using expected MSE decide what feature should be used in the root node.
* Determine the threshold for the remining feature in each branch of the tree (see Figure 4.4 in the textbook and slides).
* Obtain the price predictions (third column in Table 4.10 in the textbook).

The start of your Python code is in file **DecisionTreeStart.html**

**To submit:**

1. Well-documented Python code that implements the price predictions using the Decision Tree approach. The code must be structured similarly to what we did in class (for example, the block that plots the results must be present in your code as well) (**50 points**)