**LAB6 - Random Forests**

In this lab session, we are going to implement random forests with many different parameters, visualize the results and try to examine the differences between different parameters.

We are going to use sklearn’s RandomForestRegresson class for this task. You can examine its documentation here:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

Random forests have many parameters which we can alter to control tree characteristics. We are going to investigate 3 of those in this lab: max\_depth, n\_estimators and max\_features. Please browse through the documentation link given above to clearly understand what each of those parameters refer to.

Our .csv file is the same one we used for the previous lab. Our input consists of 7 columns (which will form an input matrix later):

FSP.1 First Serve Percentage for Player 1

ACE.1 Aces Won by Player 1

DBF.1 Double Faults Committed by Player 1

WNR.1 Winners Earned by Player 1

UFE.1 Unforced Errors Committed by Player 1

BPC.1 Break Points Created by Player 1

NPA.1 Net Points Attempted by Player 1

Our output is going to be the total number of sets won by player 1. This corresponds to the sum of 5 different columns in the .csv file:

ST1.1 Set 1 Result for Player 1

ST2.1 Set 2 Result for Player 1

ST3.1 Set 3 Result for Player 1

ST4.1 Set 4 Result for Player 1

ST5.1 Set 5 Result for Player 1

First 200 entries are going to be training data, and the rest will become test data.

For regression fitting, we will first create the regression object, then call the function *fit(x,y)* with our training data. Your complete tasks are:

1. After building your X and y for train and test data, do the regression fitting in a loop from n\_estimators = 1 to 200. Inside this loop, you should calculate three different regressions for max\_features = “auto”, max\_features = “sqrt” and max\_features = 4. For each of these fitting processes, your max\_depth parameter should be 7. An example line of regression fitting is as follows:

reg=RandomForestRegressor(max\_depth=1,n\_estimators=200,max\_features=4)

Simply replace these parameters with corresponding values inside the loop, and then call the *fit* function with your training data.

You should also keep 3 different arrays (of size 200) for the MSE values for each regression for later plotting purposes. After fitting, you should do a prediction using the test data to compute proper MSE values. You should fill in these MSE arrays inside this loop.

1. After the loop ends, do the regression two more times for:

* max\_depth = 7, n\_estimators = 200, max\_features = 4, and
* max\_depth = 1, n\_estimators = 200, max\_features = 4.

Then, do predictions using your test data for both regressions.

1. Now, the plotting. You should have 2 different figures (2 different plotting windows) for these two tasks simultaneously. You can achieve this by doing the following:

plt.figure()

# code for the 1st figure……

plt. figure()

# code for the 2nd figure……

plt.show()

For plotting Task 1, you will use the MSE arrays. The 3 different MSE arrays will be your 3 lines, which will be showing how error rates behave. Your x values will be simple arrays going from 1 to 200, and your y values will be the MSE arrays. Please plot each different line in a different color. It’s also useful if you implement a legend.

For plotting Task 2, you will plot your estimated output as your x values against each estimation’s error rate for both regressions. Both plots will be on the same figure as scatter plots, with different colors. Here are example figures:



