**LAB7 - Support Vector Machines**

In this lab, we are going to implement support vector machines (SVM). We will perform a classification task (unlike other labs, which were regression tasks) using sklearn’s “SVC” class. The details of the class, its instantiators, parameters and functions can be seen in detail here:

[https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html%23sklearn.svm.SVC)

Our input is the same .csv file we used in the previous two labs. It is a collection of tennis match results taken from Wimbledon 2013 and Roland Garros 2013. Now, we are going to try to predict if a player has won the match or not (denoted by 0 or 1) based on the player’s number of successful winners.

Our input (X matrix) is going to consist of the columns “WNR.1” (number of winners of Player 1) and “WNR.2” (number of winners of Player 2). Our Y is going to be the column called “Result”.

We are going to implement 3 different SVCs and plot their results. Each SVC will have a different kernel function:

- Linear (kernel = ‘linear’ in SVC parameters)

- Polynomial (kernel = ‘poly’ in SVC parameters)

- Radial (kernel = ‘rbf’ in SVC parameters)

Similar to the previous lab, we are going to use the first 200 samples as training data, and the rest as test data.

For each SVC, we are going to create a plot in a *separate* figure, so you should have 3 different plot windows. In each figure, you are going to implement a scatter plot using the test data. But you should separate the test data according to their *predictions*, and end up with two separate data for each SVC. In each plot, you should scatter plot these two data in different colors, so we can clearly see which observations were classified as 0 or 1.

Don’t be intimidated by the time required for running the script, it does take a long time. And again, for those who are interested, please feel free to experiment with different parameter values to see their effects more clearly. Here are the resulting plots: