LAB3 - Multiple Linear Regression

In this lab, we will perform multiple linear regression in Python. To do this, we will need the .csv provided in the LAB2 folder.

Again, this is the same .csv file we used in the previous labs. This time, we will investigate the effects of age, experience and power to the player's salary and try to find which metric has greater importance over the others.

In order to implement multiple linear regression, we need to have our independent variables as a matrix (X). Therefore, you will need to extract each independent variable column ("Age", "Experience" and "Power" in this case) and concatenate them together along with a ones column to form the correct matrix, which should look like this:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{21} & x_{31} & \dots & x_{k1} \\ 1 & x_{22} & x_{32} & \dots & x_{k2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{2n} & x_{3n} & \dots & x_{kn} \end{bmatrix}$$

You can implement the algorithm by manually calculating the coefficients, or using the "sklearn" package, which contains very useful functions for data science in general. In that case, you will have to create a LinearRegression object and later use its functions to automatically calculate the coefficients (and much more!). You can learn about the LinearRegression class and its functions here: http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

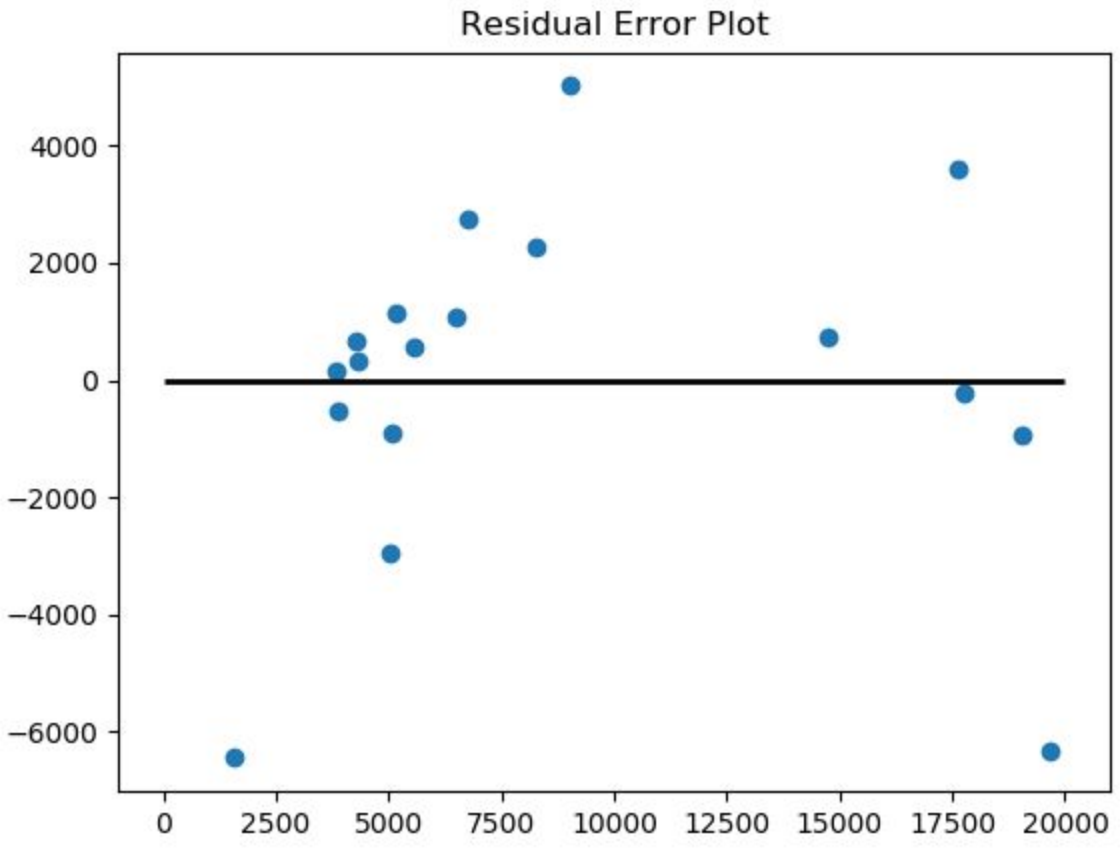
The formula to manually calculate the coefficients is given below:

$$\hat{\beta} = [\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'\mathbf{y}$$

After calculating the coefficients, we need to plot the results. Unlike single linear regression, we don't have a regression *line*, but a regression *hyperplane* (with more than 3 dimensions), which unfortunately cannot be plotted. We can, however, plot the *errors* for each observation. We can plot our predicted Y values against each one's error margin. Again, you can calculate these or use sklearn functions. The formulas are (for predicted Y values and error margins, respectively):

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} \qquad \hat{\mathbf{u}} = \mathbf{y} - \hat{\mathbf{y}}$$

The output of the plot should look like this:



After finishing the plot, you should display the independent variables *in order of importance* in the console, from high to low. Example:

Power
Age
Experience

Please use numpy arrays for any numerical arrays or matrices. Similar to the previous labs, we will use matplotlib for plotting purposes.