Practical work 1 - ML : Linear regression

The objective of this work is to learn how to use the gradient descent algorithm on a dataset in order to create a linear regression model.

The file "**linreg_data.csv**" provided with this project contains data of dimension (250, 2). You will use this file as training data.

Exercise 1

Create linear regression model in new Jupyter notebook.

1. Use the Machine Learning tool **scikit learn** to develop and train the model, as we've learned previously.
2. What is the optimal *Learning Rate* and how many iterations does the **Gradient Descent** need to converge.
3. Display the Scatter Plot and the Regression Line.
4. What is the best accuracy value of the model.
5. Display coefficients of linear regression equation.
6. Compute the prediction of the value (6.5).

Exercise 2

1. Recalculate the coefficients of the model equation using the least squares method. Use this time the **For** loop to find b_0 and b_1.
2. Display on a single graph: the scatter plot, the model learned by **Gradient Descent** and the model obtained by the least squares method, in different colors. What do you notice ?

Exercise 3

1. Implement the gradient descent algorithm in a function, then use this function to calculate the model parameters as well as the cost function.
2. Display the scatterplot and regression line.
3. Find the correlation coefficient.
4. Compare the parameters (b_0, b_1) obtained by: the **scikit-learn** library, the least squares method and the implemented gradient descent function.

Exercise 4

1. After having implemented **Batch gradient descent**, write two other functions: One that implements **Stochastic Gradient Descent** (SGD) and a second that implements **Mini-Batch Stochastic Gradient Descent** (Mini-Batch SGD).
2. Perform a vector implementation of **batch gradient descent**..

Good luck