Lab 2: Simple linear regression

Submit:
- Your notebook (within a zip file) to http://deei-mooshak.ualg.pt/~jvo/ML/submissions/
- Up to October 9, 2025

Distributed with these instructions is a Jupyter Notebook named **regression-intro.ipynb**, along with a dataset file named **demodataset.csv**. Download the notebook and the dataset file, inspect and execute all the code cells in the Jupyter Notebook.

Download the file **Lab2-regression.ipynb**, complete the notebook by answering the questions below, and submit it as your assignment in zip format.
Note: define functions when needed to prevent redundant code.

1.

a) Compute θ_0^* and θ_1^* for the line of best fit:
   - i) using scipy.stats.linregress()
   - ii) implementing the linear regression model below using **NumPy** functions:

$$\theta_0 = \bar{y} - \theta_1 \bar{x}$$

$$\theta_1 = \frac{SS_{xy}}{SS_{xx}} = \frac{\sum_{i=1}^{m} \left[(x_i - \bar{x}) (y_i - \bar{y}) \right]}{\sum_{i=1}^{m} (x_i - \bar{x})^2}$$

   - iii) check that i) and ii) produce the same results

b) Print the dataset and superimpose the line of best fit.

c) Predict y for $x = 2.231$. What is the corresponding residual? Superimpose on the graphic obtained in b)

2.

a)
   - i) Express $J(\theta_0^*, \theta_1^*)$ in vector notation and compute it for the given dataset
   - ii) Is there any advantage to computing in vector notation with **NumPy**?
   - iii) Show that point (average of X, average of Y) belongs to the line of best fit.

b) For θ_0^* and θ_1^*, plot the residuals *vs* the independent variable x. Print the mean and variance of the residuals. Briefly comment on what you observe.

c) For θ_0^* and θ_1^*, analytically prove that the sum of residuals is zero. Print the sum of the residuals.

d) Prove that the derivative of the cost function, with respect to θj is:

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_j^{(i)}) - y^{(i)}) x_j^{(i)} \quad , \quad j = 0, 1 \ldots n$$

3.

a) Express in vector notation the following gradient descent updating expressions:

$$\theta_j = \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_j^{(i)}) - y^{(i)}) x_j^{(i)} \quad , \quad j = 0, 1 \ldots n$$

b) Apply gradient descent and linear regression to the dataset. Find θ_0^*, θ_1^* and $J(\theta_0^*,\theta_1^*)$.

c) What is the number of iterations and the value of the learning rate α, that approximate the θ vector and the cost J obtained in 3. b) to those found in 1. a) and 2. a)?

d) Plot $J(\theta_0,\theta_1)$ as a function of the number of iterations. Briefly comment on what is observed.

Bibliography

[1] Jupyter notebook. https://jupyter.org/
[2] mathisfun. List of derivative rules:
https://www.mathsisfun.com/calculus/derivatives-rules.html