Lab 4: Multivariate and logistic regression

Submit:
- Your group report (within a zip file) to http://deei-mooshak.ualg.pt/~jvo/ML/submissions/

  Up to October 24, 2024

Distributed with this quiz is a brief Object oriented tutorial for Python and ML, in the form of a Jupyter Notebook named **PythonOOtutorial.ipynb.**
Read this tutorial, inspect and execute all the Jupyter Notebook code cells before answering this quiz.

Additionally, distributed with this quiz are the datasets file **demodataset.csv** and **lab04data.csv**, and another notebook, **lab04-base.ipynb**, which contains the definition of the base classifier classes discussed in the previous notebook.
Download this notebook and the dataset files, execute all the cells, and then, add to the end of this notebook your answers to the following questions:

1. Express the gradient descent update with regularization in vector form:

$$\theta_0 = \theta_0 - \alpha\frac{1}{m}\sum_{i=0}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)} \quad , \quad j = 0$$

$$\theta_j = \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \quad , \quad j = 1, 2, \ldots, n$$

Suggestion: start with the gradient descent update without regularization, obtained in lab 3

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha\frac{1}{m}(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})^T\mathbf{X}$$

2. Load the dataset in the companion file **demodataset.csv**. Shuffle it and divide it into disjoint training and testing datasets.

3.
   a) Implement a linear (in the parameters) basis function regression model using the Gradient descent model with regularization, with a new **Classifier** subclass named **GradDescReg**.

      GradDescReg constructor must accept the following parameters to parametrize the optimizer:

```
__init__(self, maxIter:int=1000, convDiff:float=10**-8, alpha:int=0.001, lambda_:int):
    //…
```

      where:

maxIter	is the maximum number of iterations.
convDiff	is the θ improvement early stop threshold, which means that when the difference between the current iteration θ and the previous iteration θ is smaller than ***convDiff*** the gradient descent loop ends, even if the number of iterations did not reach ***maxIter***.
alpha	is the learning rate.
lambda_	is the λ regularization parameter.

b) Using **GradDescReg** class, fit the regression model to the training set and predict with the test dataset, for values of the regularization term λ = [0, 200], with resolution 1.
Plot the mean square error (mse) for the test dataset for all values of λ.
What is the λ that minimizes the mse?

c) Plot the train dataset, test datasets, and the regression for the λ that minimizes the mse obtained in b).

d) Repeat b) with the **NormalEQReg** class.

e) Compare the results in c) and d)

4. Propose a dataset $\{(x1^{(i)}, x2^{(i)}, y^{(i)}) \mid i=1,\ldots, 4\}$ where *x1* and *x2* are independent binary variables and *y* is a linearly separable dependent binary variable. Apply logistic regression and compute the estimated probability of $y = 1$ when $x1 = x2 = 0.5$.

5. Load the companion data file **lab04data.csv** in the format $\{(x1^{(i)}, x2^{(i)}, y^{(i)}) \mid i=1,\ldots, m\}$ where *x1* and *x2* are the independent variables and *y* denotes the corresponding class.

   a. Plot the 2 available classes in the space *x1, x2*

   b. Apply regularized logistic regression using the following feature vector: [1 *x1 x2* $x1^2$ *x1*x2* $x2^2$ $x1^3$ … $x1^5$*x2 $x2^6$]' (28 elements overall) – see the companion file **map.py** – and a suitable λ value. Visualize the boundary decision