FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

Machine Learning 1
Assessed Coursework 2

This assignment must be submitted by February 27th, 2023 at 10:00 am.
Late submissions penalties: 10% of the total value of the coursework for each hour of delay, or fraction of it.

This coursework is assessed and mandatory
and is worth 30% of your total final grade for this course.

Learning outcomes assessed

This coursework will test your understanding of some important machine learning algorithms through the writing of Matlab code that implements them.

Instructions

Identifier
Please choose a random number of 6 digits. Make sure that you keep a copy of that number as it will be used to provide the feedback (please avoid trivial numbers, such as 000000 or 123456. Also please avoid numbers starting with zero).

Submission
Compress the files of your submission into a unique zip file and rename it with your random digit number (so the zip file name becomes something like 723923.zip). Then email *your zip file as an attachment at alberto.paccanaro@fgv.br with the subject "URGENT – MSC COURSE – COURSEWORK 1 SUBMISSION".*

All the work you submit should be solely your own work.
Coursework submissions will be checked for this.

DATASET DESCRIPTION

You will be using 3 different datasets, that you will find on the Eclass page for this coursework.

1) *Car* dataset: this dataset is constituted by *1728* points in 6 dimensions. Each point refers to a car, which is described by 6 attributes, all of them (first 6 columns). Each car belongs to one of possible 4 classes (last column).

   The description of the attributes is as follows

   buying: buying price
   maint : price of the maintenance
   doors : number of doors
   persons: capacity in terms of persons to carry
   lug_boot: the size of luggage boot
   safety : estimated safety of the car

   The values of the attributes are:

   Buying: v-high, high, med, low
   Maint: v-high, high, med, low
   Doors: 2, 3, 4, 5-more
   Persons: 2, 4, more
   lug_boot: small, med, big
   safety: low, med, high

2) *Concrete dataset*: Concrete is the most important material in civil engineering. The concrete compressive strength is a highly nonlinear function of age and ingredients. These ingredients include cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate. This dataset contains the actual concrete compressive strength (MPa) for a given mixture under a specific age (days) as it was determined from laboratory. Data is in raw form (not scaled).

   The datasets contains 1030 datapoints, the first 8 columns are the features, and the one to be predicted is the last one.

3) *Raisin dataset*: different varieties of raisins are grown in Turkey and that belong to 2 classes, Kecimen and Besni raisin. For this dataset, a total of 900 raisin varieties were used. Original images were subjected to various stages of pre-processing and 7 morphological features were extracted.

   Attribute Information:

   - Area: Gives the number of pixels within the boundaries of the raisin.
   - Perimeter: It measures the environment by calculating the distance between the boundaries of the raisin and the pixels around it.
   - MajorAxisLength: Gives the length of the main axis, which is the longest line that can be drawn on the raisin.
   - MinorAxisLength: Gives the length of the small axis, which is the shortest line that can be drawn on the raisin.
   - Eccentricity: It gives a measure of the eccentricity of the ellipse, which has the same moments as raisins.
   - ConvexArea: Gives the number of pixels of the smallest convex shell of the region formed by the raisin.
   - Extent: Gives the ratio of the region formed by the raisin to the total pixels in the bounding box.

- Class: Kecimen and Besni raisin.

The datasets contains 900 datapoints, the first 7 columns are the features, and the one to be predicted is the last one.

EXERCISE 1 (value: 6 %)

Write a Matlab function "***LinearRegressionWD***" that trains a Linear Regression with weight decay regularization model using stochastic gradient descent (note: not the normal equations with weight decay).

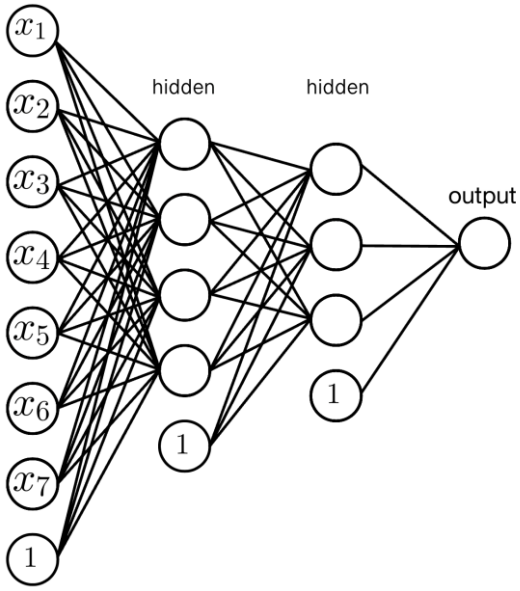
Your implementation learn to predict the Concrete compressive strength (MPa, megapascals) of the `Concrete_Data.csv` dataset, given the other 8 attributes.

The implementation will provide a plot of the training error as a function of the iteration. The algorithm will stop when the relative improvement in the error function between two successive iterations will be smaller than the value of a user-defined threshold τ.

You will also submit a script "***ScriptExercise1"*** that will run your function ***"LinearRegressionWD"*** on 90% of the dataset and will report the test error obtained on a 10% held out dataset.

EXERCISE 2 (value: 12 %)

Write a Matlab function *"**MultiLayerPerceptron**"* that trains the neural network specified in the following diagram using batch gradient descent for predicting the type of raising (Kecimen and Besni raisin) using the 7 features of each datapoint.



The implementation will plot the training error as a function of the iteration. The algorithm will stop when the relative improvement in the error function between two successive iterations will be smaller than the value of a user-defined threshold τ.

You will also submit a script "***ScriptExercise2".*** This script, will first train the model using the training set provided and your "***MultiLayerPerceptron"*** function. Your script will then assess the performance of your model on the testing dataset (also provided on EClass) by creating a figure displaying the Receiver operating characteristic (ROC) curve.

(note: in this exercise you are also required to implement a function called ***"ROCcurve"*** that will calculate and plot the ROC curve).

EXERCISE 3 (value: 12 %)

You will write a script "***ScriptExercise3"*** that builds a decision trees for the car classification problem.

Your script will use the Information Gain measure to decide the nodes for your tree. You will stop splitting the nodes when a node has less than *p* datapoints, where *p* is a variable that is set within your script).

It is not necessary to store the tree. The output of your script would be simply the set of rules represented corresponding to the nodes.

Keep in mind that that your decision trees should all be binary (for any non-binary tree there is a binary one which is equivalent)

Marking Criteria

In order to obtain full marks for each question, you must answer it correctly and completely.
Marks will be given for writing compact, vectorised code and avoiding the use of "loops" (*for* or *while* loops) for carrying out operation on matrix and vector elements.