PW4 – Exercise: Regression with an Artificial Neural Network

The objective of this exercise is to apply the principles developed in labs 3 and 4 to implement a regression model with an artificial neural network.

General instructions

1. The exercise must be handed before [date is on Moodle!].
2. You must hand a report to explain your approach (data analysis, model, results, …). You can simply add those elements within your notebook.

Topic

You have been recruited by a farmer who grows avocados. He needs you to anticipates the income he can expect using avocados sales data in the USA.

The dataset is available on Moodle. You must use it to build a model that can predict the average price of an avocado.

Before you start, a quick word about 3 attributes that may seem unclear if you are not avocados pros: columns "4046", "4225" and "4770" are avocados sizes (respectively small, big, very big). As you will see when you look at the data, all the other attributes are quite self explanatory.

Here are the overall step you must take:

1. Load the data using PandasCharger les données avec Pandas

2. Prepare the data

- Are there any missing data? If yes, take appropriate actions.
- Extract the target for your model (using the Pandas method `pop()`: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.pop.html)
- Delete columns 'Date' and 'Unnamed:0' (the first attribute is redundant with the attribute 'Year' and the second attribute is an index that is not relevant for a prediction)
- Split the data into a training and a testing dataset using Sklearn `train_test_split` (no stratified split).
- [for both train and test] extract the categorical attributes and transform them into one-hots.
- [for both train and test] standardize all cardinal data.
- [for both train and test] put back the transformed categorical attributes within the normalized dataset.

3. The model:

- Create a sequential model using TF/Keras. This model must have 2 hidden fully connected layers with, respectively, 32 and 16 units. To specify the inputs, when you create the first hidden layer, simply use the following input shape: `input_shape=X_train.shape[1:]`. The output layer is a dense layer with a single unit (*explain why you need only one unit there*).
- The cost function of your model must be a MSE. Use SGD as your optimizer.
- Train your model for 30 epochs (specify `validation_split=0.15` as the proportion of data to use for validation).
- Evaluate your model. Do a test using some data points of the test set.