TELECOM Paris **SD-TSIA204**
2023/2024

TP 2 : Linear regression

For this lab, you have to upload a **single** ipynb file. Please use the following script to format your filename (bad name will lead to a 1 point penalty):

```
# Change here using YOUR own first and last names
fn1 = "john"
ln1 = "smith"
filename = "_".join(map(lambda s: s.strip().lower(),
                        ["SD-TSIA204_lab2", ln1, fn1])) + ".ipynb"
```

You have to upload it on eCampus. The deadline is the 31/01/2024 at 23h59 in eCampus. Out of 20 points, 5 are specifically dedicated to:

- Presentation quality: writing, clarity, no typos, visual efforts for graphs, titles, legend, colorblindness, etc. (2 points).
- Coding quality: indentation, PEP8 Style, readability, adapted comments, brevity (2 points)
- No bug on the grader's machine (1 point)

Beware:

1) Labs submitted late, by email or uploaded in a wrong group folder will be graded 0/20.
2) The package `statsmodel` in NOT allowed.
3) For certain questions, answers must be accompanied by explanations. An answer without an explanation will be graded as 0.

Note: you can use https://github.com/agramfort/check_notebook to check your notebook is fine, and also use https://github.com/kenkoooo/jupyter-autopep8 to enforce `pep8` style.

---

1) For the first question, we load a standard dataset from `sklearn.datasets` named `fetch_california_housing`. This dataset has only $p = 8$ variables.
   (a) Estimate the coefficients with the expression of the normal equaitons seen in class. Code two functions to compute the MSE and the R2 coefficient and compare them with the version of `sklearn` for the train and the test sets.
   (b) Finally, give the confidence intervals at level 99% for all the coefficients coding the expression for the CI seen in session 3.
2) For the rest of the TP, we use the dataset in eCampus `data`. Load and preprocess the data:
   (a) Separate the data in train and test sets: save one fourth of the data as testing (`train_test_split from sklearn.model_selection` with the random seed set to 0 and standardize both the training and testing sets using the `fit_transform` and transform functions in `sklearn.preprocessing.StandardScaler`.
   (b) Fit a regular OLS.

Variable selection

3) Program the method of forward variable selection based on hypothesis tests for regression coefficients. This method starts from an empty set of variables S and at each iteration selects one variable relevant for predicting y and includes it in the set S. It runs until a halting condition is met. The coding process is as follows:

(a) Develop a function that, given a dataset $X \in \mathbb{R}^{n \times p}$ and y, fits p linear regression models, each using only feature X_j to predict y. For each model, conduct a test of no effect, as discussed in session 3, and compute the p-value of the test. This function should return the coefficient with the smallest p-value. Explain the significance of the p-value in this context.

(b) Apply the function iteratively. At each iteration, select the feature X_f with the smallest p-value and:

i. Include it in the set S.

ii. Remove it from X.

iii. Subtract from y the ~~residuals~~ prediction of the model fit with feature X_f. Elaborate on the reason for subtracting the residuals.

(c) Add a halting condition to the algorithm: Stop adding features to the set S when the p-value exceeds 0.05. Plot the p-values for every coefficient for the first 5 iterations (all in the same plot).

Extensions

4) Run ridge regression using scikit-learn on the training set. Run the code for 30 different values of the penalty parameter, which should be on a logarithmic scale between 10^{-1} and 10^{6}. Display two subplots at the end:

(a) The first subplot should show the evolution of the coefficients for each different value of the penalty parameter.

(b) The second subplot should display the evolution of the R-squared coefficient at each of the 30 iterations.

Since we are going to perform similar tasks for Lasso and Elastic Net, it is mandatory to write this code as an independent function that can be parameterized for each specific case.

5) Run the code for Lasso as explained in Point 4. Run the code for 30 different values of the penalty parameter, which should be on a logarithmic scale between 10^{-3} and 10^{2}.

6) Run the code for ElasticNet as explained in Point 4. Run the code for 30 different values of the penalty parameter, which should be on a logarithmic scale between 10^{-3} and 10^{2}.

PCR

7) Compute the singular value decomposition of the covariance matrix. For consistency in the notation use $U, s, V = SVD(X^{\top}X)$.

(a) Plot a heatmap of the covariance matrix.

(b) Compute the PCA for the data using the SVD.

(c) Plot the amount of variance explained by the first k components for $k \in 2..p$. How many variables do we need to explain more than 90% of the variance?

(d) Plot the projected data with $k = 2$ using as color the value of y and interpret the plot.

(e) Plot the the two first principal directions.

(f) Run OLS on the projected data (PCR) using k components for $k < 50$. Select the k that returns the best score of the OLS model and plot the evolution of the scores with k.

Comparison of the models

8) Summarize the results of the models and elaborate in their main characteristics. Plot all the training and testing errors for all the models considered and elaborate on the results.