**Practical Lab 2: Multivariate Linear Regression, Non-Parametric Models and Cross-Validation**

**The dataset**: Scikit-Learn Diabetes dataset. Scikit-Learn provides toy datasets ([list](https://scikit-learn.org/stable/datasets/toy_dataset.html)[)](https://scikit-learn.org/stable/datasets/toy_dataset.html))). Here we will use the diabetes dataset ([description](https://scikit-learn.org/stable/datasets/toy_dataset.html#diabetes-dataset)[).](https://scikit-learn.org/stable/datasets/toy_dataset.html#diabetes-dataset)).) Make sure to go over this description, before start exploring the data.

The data can be loaded into a notebook by the code below. See the [documentation](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html#sklearn.datasets.load_diabetes) to read about the `load\_diabetes` method and its optional arguments: *return\_X\_y*, *as\_frame* and *scaled*. you can add them if needed:

*from sklearn import datasets*

*datasets.load\_diabetes()*

**Objective**: build a model that can best predict the risk of diabetes progression. This will be used as a screening tool to help physicians with identifying patients at risk. The models that we look into are:

1. Univariate polynomial regression models

2. Multivariate Polynomial models

3. Decision Trees

4. kNNs

In this lab we will evaluate the models using [R-squared](https://en.wikipedia.org/wiki/Coefficient_of_determination), Mean Absolute Percentage Error ([MAPE](https://en.wikipedia.org/wiki/Mean_absolute_percentage_error)[)](https://en.wikipedia.org/wiki/Mean_absolute_percentage_error))) and Mean Absolute Error ([MAE](https://en.wikipedia.org/wiki/Mean_absolute_error)[)](https://en.wikipedia.org/wiki/Mean_absolute_error))) metrics. To do that we will run a train-validation-test split.

**Writing a good ML Report**: Notice to follow a [good notebook structure](https://github.com/CSCN8010/CSCN8010/blob/main/class_notebooks/notebook_structure/notebook_structure.ipynb), and focus on readability and clarity of the code, insights and conclusions.

The steps to be taken:

**Part 1 (5 points)**

1. Get the data.

2. Frame the Problem. Notice that the dependent variable of diabetes progression, namely "disease progression one year after baseline".

3. EDA - Describe the data, explore it, and provide insights about it. This should include at least: statistics, scatter plots histograms, a correlation matrix, and concise and relevant insights (4 point).

4. Clean the data if needed, and explain your reasoning for your reader (1 point)

5. Split the dataset to a train (75%) and validation set (10%), and test set (15%).

**Part 2 (10 points)**

6. Models: a univariate polynomial regression on the BMI feature versus the "disease progression one year after baseline" - from degree 0 to 5 (6 models) (1.5 point)

7. Compare the models using the training and validation data. Construct a table summarizing the train validation results. Each model should have a separate row in the table (3 points):

    1. R-Squared

    2. Mean Absolute Error (MAE)

    3. MAPE

8. Identify the best model based on the table above.

9. Run the chosen model on the test set and provide results (R-Squared, MAPE, MAE) (1 point).

10. Plot a graph of the train, validation and test data points, and the fit of the chosen model over each of them (1 point).

11. Write down the equation of the best model (with a two decimal digit precision, assuming it's sufficient) (1 point).

12. Calculate the expected diabetes progression for a BMI value of your choice using `model.perdict()` for that value (0.5 point).

13. How many trainable parameters are we fitting for each of the models? Explain these values. One way is to use sklearn function `get\_feature\_names\_out()`(1 point)

14. Provide a conclusion section. In particular, do a deep dive on where the model fails, and add a short section in the conclusions that describes the model limitations. This should be in addition to summarizing it performance. (1 point).

**Part 3 (5 points)**

In this part, use **all features** in the dataset, or drop some features per your discretion**based on the EDA**.

Repeat the steps in part 2 for the following models:

    1. Two polynomial models (degrees >1, of your choice)

    2. Two decision trees (e.g. consider changing the `max\_depth` hyperparameter)

    3. Two kNNs

Note: Plots of multivariate models are not required (step 10)