

## Lab 2: July 4

*Instructor: Donlapark Ponnoprat*

## Task 1: Tabular data

Upload the Boston housing dataset `housing.tgz` to Google Colab. File with one of `*.tar`, `*.tar.xz`, `*.txz`, `*.tar.gz`, `*.tgz`, `*.tar.bz2`, `*.tbz2` is a Linux archive file, so you need to extract the file using one of the following lines:

```
!tar -xvf yourfile.tar or yourfile.tar.xz or yourfile.txz
!tar -xzvf yourfile.tar.gz or yourfile.tgz
!tar -xjvf yourfile.tar.bz2 or yourfile.tbz2
```

With this dataset, we are going to train a linear regression model with

`y = median_house_value` and `X = the other features`.

But first, we need to do some data cleaning/preprocessing.

1. Encode the categorical variable with an appropriate encoder (either `Ordinal Encoder` or `OneHotEncoder`).
2. Split the data into `X` and `y`.
3. Split the data into training set and test set (choose your own proportions).
4. Impute and normalize the data. Make sure that there is no “data leakage” from the test set to the training set.
5. Train the linear regression on the training set, and report the **coefficient of determination** ( $R^2$ ) on the test set. To learn more about the `scikit-learn`’s `LinearRegression` class, see the [Documentation](#).

## Task 2: Time series data

Upload the Chiang Mai weather dataset `chiang_mai_1998-2019.csv` to Google Colab. I recommend to read the dataset as a `numpy` array, not a `pandas` dataframe, unless you are extremely good at dataframe manipulation.

With this dataset, we are going to train a linear regression model with

`y = next day Precipitation` and `X = all features in previous  $t$  days, including today`,

where  $t = 2, 3, \dots, 10$ . First, we need to preprocess the data

1. Remove the **Date** column.
2. Split the data into training set and test set (choose your own proportions).
3. Normalize the data. Make sure that there is no “data leakage” from the test set and validation set to the training set.
4. (sliding window) Define a function that takes one of the sets **and** **t** and outputs two **numpy** arrays in the following way: for  $t = 2$ , the set

|       |       |       |         |
|-------|-------|-------|---------|
| $H_1$ | $T_1$ | $P_1$ | $Pre_1$ |
| $H_2$ | $T_2$ | $P_2$ | $Pre_2$ |
| $H_3$ | $T_3$ | $P_3$ | $Pre_3$ |
| $H_4$ | $T_4$ | $P_4$ | $Pre_4$ |
| $H_5$ | $T_5$ | $P_5$ | $Pre_5$ |
| $H_6$ | $T_6$ | $P_6$ | $Pre_6$ |

is transformed into:

| X     |       |       |         |       |       |       |         | y       |
|-------|-------|-------|---------|-------|-------|-------|---------|---------|
| $H_1$ | $T_1$ | $P_1$ | $Pre_1$ | $H_2$ | $T_2$ | $P_2$ | $Pre_2$ | $Pre_3$ |
| $H_2$ | $T_2$ | $P_2$ | $Pre_2$ | $H_3$ | $T_3$ | $P_3$ | $Pre_3$ | $Pre_4$ |
| $H_3$ | $T_3$ | $P_3$ | $Pre_3$ | $H_4$ | $T_4$ | $P_4$ | $Pre_4$ | $Pre_5$ |
| $H_4$ | $T_4$ | $P_4$ | $Pre_4$ | $H_5$ | $T_5$ | $P_5$ | $Pre_5$ | $Pre_6$ |

See <https://www.cienciadedatos.net/documentos/py27-time-series-forecasting-python-scikitlearn.html> for nice visualizations of the sliding window technique.

For  $t = 2, 3, \dots, 10$ , perform step 5–6 below:

5. Apply the function in 4. to the training set and obtain  $X_{train}, y_{train}$ . Apply it to the test set and obtain  $X_{test}, y_{test}$ .
6. Train the linear regression model on  $X_{train}, y_{train}$ . With the trained model, make predictions on  $X_{test}$  and compute the RMSE against  $y_{test}$ .

Finally, we plot our results.

7. Plot the RMSE for  $t = 2, 3, \dots, 10$ . What is the best value of  $t$ ?