CS452/552 – Data Science with Python
Assignment-1

Life Expectancy by Linear Regression
Due: 14.11.2021 – 23.55

created by Furkan Kınlı, for questions → e-mail: furkan.kinli@{ozu, ozyegin}.edu.tr

1. Definition

In this assignment, you will implement a **Jupyter notebook** that solves the problem of finding the life expectancy factor in different countries by using one of the simplest regression strategies, called Linear Regression. We will **NOT** provide a baseline notebook for this problem, so you need to implement the notebook solving the life expectancy prediction problem from scratch. We expect you to pre-process the data that we provide as we expect, train different linear regression models that predict the life expectancy of the people in different countries by including or excluding some set of features given in the data, and lastly report the performance of each model **qualitatively** and **quantitatively**. The data contains 2938 unique entries with 20 different features without any partition. You are expected to split the data as training and test set.

In this assignment, you are only allowed to use the scientific computation libraries, which are introduced in the lectures (NumPy [docs], Pandas [docs], Scikit-learn [docs]), and also you are free to use any visualization library (e.g., Matplotlib [docs], Seaborn [tutorial]). However, we kindly expect you **to conduct a comprehensive experimental setup and to visualize the data and the results in your notebooks**, where each cell introduces one thing and has a comment for what it does. The details of the task are in the following sections.

In the report, you are expected to introduce the project and the aim of this project, to explain the algorithms/models employed in this project in detail. Also, it is expected to show the experimental setup (i.e., each parameter in your model or each feature in the data that you change for training), and to report the training performance in all experimental settings and the test performance of **tuned** settings for each algorithm/model. Moreover, descriptive tables, plots and figures are required **both to observe the data better and to show the results for all settings**. Please do **NOT** forget to add visualization for the images and the compressed results in your notebooks.

We strongly recommend & ask you to read about the algorithms and the problem before starting to implement your assignment.

2. Implementation Details

For this assignment, we will **NOT** provide a baseline Jupyter notebook, which means you will implement the notebook for life expectancy prediction solution from scratch. You may not use more than one notebook, which means **you need to have a single "*.ipynb" file that**

is responsible for all tasks. For each cell, a descriptive comment in the first line is **must**. Please use **the newest** version of the libraries.

The detail of this task as follows:

- Giving information about the linear regression in your report.
- Giving information about the dataset (e.g., what are the features? How many rows/columns? Numeric/categoric features? Any missing information in rows? The overall statistics of the dataset and **more**). You should extract them in notebook, then put into your report with their descriptive comments, discussion, or captions.
- Fetching the data in notebook.
- Checking out the data (e.g., head, description, info, # of missing rows, correlation, etc.). You should extract them in notebook, then put into your report with their descriptive comments, discussion, or captions.
- Visualizing the data for each dependent feature with independent feature (e.g., infant deaths vs. life expectancy, BMI vs. life expectancy etc.). You should extract them in notebook, then put into your report with their descriptive comments, discussion, or captions.
- Deciding a set of features that you think that the most compact way of representing the data before using any statistical/predictive analysis. You should discuss why you pick your set of features in detail in report.
- Feature engineering (e.g., categorical features to one-hot dummy vectors, drop/impute rows with missing info, scale the data to the range 0 to 1, extracting and visualizing the correlation between features, picking different sets including **at least 8** features in **5 different and unique settings,** split the data for training and test set (important note: use "random seed" parameter as 147) and **more**. (hints: choose dependent variable carefully, eliminate unnecessary columns, standardization to each column, do not forget to convert categorical features to one-hot format, please act all columns of one-hot vector as **one feature**)
- Training LR models for 5 different setups in notebook.
- Evaluate the models with different metrics (i.e., mean-squared error, mean-absolute error, root-mean-squared error, r2 score in notebook. You should extract them in notebook, then put into your report with their descriptive comments, discussion, or captions.
- Visualize the results of all setups for each metric. You should extract them in notebook, then put into your report with their descriptive comments, discussion, or captions.
- Printing the general formula of the model with the best performing settings. You should extract them in notebook, then put into your report with their descriptive comments, discussion, or captions.
- Discussing the model coefficients for all 5 setups in report.
- Discussing the extracted information which may be useful for, for example, WHO (World Health Organization) or Turkish Ministry of Health, that's why we try to use ML or predictive analysis on such a case. (**MOST IMPORTANT PART** in report) (e.g., development rate of a country influences the life expectancy of this country, as can be seen in Figure X, they are highly correlated. The infant death rate negatively affects the selling price as shown in Figure X. etc.)

3. Criterion

Notebook & Report: 100 Points

Linear regression and life expectancy problem are well-studied in the literature and on online resources, and thus there are many different coding examples online. We have almost all of them, at least we have a chance to compare your notebooks with our collected online resource database for this problem. Please do **NOT** try to use them directly. In any circumstances of copying online resources directly, **you will get 0 points**.

Please do **NOT** include any other 3rd-party library to your notebook (except NumPy, Pandas, Scikit-learn, Matplotlib, Seaborn), because you do not need them. Including such libraries may cause a problem for running your code in different local environments (e.g., dependencies). For any dependency on run-time that **TA can solve**, **you will get -15 points penalty**; for any dependency on run-time that **TA cannot solve, you will get -30 points penalty**.

You need to write a detailed report to explain your design and solution**. 1 PARAGRAPH CODE EXPLANATION IS NOT A REPORT, and such submissions will be simply ignored, and you will get 0 points, even if your notebook does something**. Please follow the technical report writing rules (i.e., *Introduction*, *Methodology*, *Implementation Details*, *Results*, *Conclusion*). It is **must** to add the visual contents that you extract in your notebooks to your reports.

The notebook file (*.IPYNB) and the report file **(*.PDF**) should be zipped (**.ZIP**) together. The filename of final submission file should be in the format of "**NAME_SURNAME_ID_hw1.zip**". Please follow this structure for your submission. You should **NOT** include the data (.CSV) which is provided by us to your submission. Not following this structure will be resulted as **"-10 points penalty for each" (file extension, filename, zipping, not including the data) without any excuse**.

Late submission is allowed for **1 extra day with -20 points penalty**.

You may discuss the algorithms and so forth with your friends, but this is an individual work. Therefore, you must submit your original work. **In any circumstances of plagiarism, first, you will fail the course, then the necessary actions will be taken immediately.**