

Practical Session 1 – Linear and Logistic Regression

The objective of this tutorial is to implement in Python the linear and logistic regression models studied in class. Experiments will then be conducted to evaluate the model's performance on test datasets.

Part 1 – Linear Regression

1.1 Presentation of the provided elements

A Python file (`linear_regression.py`) is provided, containing functions that need to be completed to make the program operational.

To test the developed functions, two data files are provided:

- **food_truck.txt**: contains a dataset characterized by a single predictive variable (the population size of cities, in the first column). The target variable (second column) corresponds to the profit of a food truck in that city. The problem associated with this dataset is to predict the profit a food truck could make based on the size of the city.
- **houses.txt**: contains a dataset characterized by two predictive variables (the area of a house and the number of rooms, in the first two columns). The target variable (third column) corresponds to the price of the house. The problem associated with this dataset is to predict the price of a house based on its area and the number of rooms.

1.2 Writing the program

Complete the functions in the provided file to make the program operational. It is recommended to follow the order of the functions indicated in the previous section and to test them at each step.

1.3 Experiments

Once the program is operational, conduct experiments to evaluate the model's performance and the behavior of the learning process. For example:

- Influence of the learning rate value
- Influence of the number of iterations for gradient descent
- Influence of the gradient descent strategy (batch, stochastic, mini-batch)
- Evaluate the model on other datasets (for instance, from <https://www.kaggle.com/code/rtatman/datasets-for-regression-analysis/notebook>).

Part 2 – Logistic Regression

2.1 Presentation of the provided elements

A Python file (`logistic_regression.py`) is provided, containing functions that need to be completed to make the program operational.

To test the developed functions, a data file is provided: **scores.txt**. This file contains a dataset characterized by two predictor variables (the students' scores on two exams) (the first two columns). The target variable (third column) indicates whether the student is admitted (1) or not (0) to the University. The problem associated with this dataset is to predict whether a student will be admitted to the University based on the scores obtained in the two exams.

2.2 Writing the program

Complete the functions in the provided file to make the program operational. It is recommended to follow the order of the functions indicated in the previous section and to test them at each step.

2.3 Experiments

The developed program learns and evaluates the logistic regression model on the training data. However, to properly evaluate the real performance of a predictive model, it is necessary to apply it to test data, which should be different from the training data.

Additionally, logistic regression is suited for binary classification. For a multi-class problem, one approach is to use the "one-vs-all" strategy, as discussed in class.

The tasks to be done are as follows:

1. **Splitting the data into training and test sets:**
Develop the necessary functions to split the data into two subsets: one for training and one for testing. This split should be parameterized by a real number (between 0 and 1) indicating the ratio of training data relative to test data.
2. **Adapting the program for multi-class classification:**
Modify the program to handle multi-class classification. You will apply it to a dataset of your choice (with at least 3 classes) in order to evaluate it. Numerous datasets can be found here: [UCI Machine Learning Repository](#).