Assignment 3

Fundamentos del Aprendizaje Automático
Curso 2025-2026

This third practical assignment of the course aims at expanding and clarifying the concepts introduced during the entire course (mainly, **T3** to **T7**).

You will follow this script to develop the required programs and experiments to eventually deliver a report summarizing the main insights and conclusions obtained.

Objectives

- Further understand the evaluation methodology of machine learning methods by introducing hypothesis tests.
- Apply dimensionality reduction techniques to remove redundancy in the data assortments.
- Implement basic neural schemes.
- Explain and justify the reasons for the decisions taken.
- Learn to report results and insights in a scientific manner.

Tasks

T1. Dataset: Considering the **initial** dataset developed for **Assignment 2**, obtain two alternative assortments:

1. A reduced version with **PCA** with, at least, *60% of the Cumulative Variance*. For this you may use the `PCA` method from the `decomposition` collection in the `sklearn` library.
2. A reduced version with a **neural Autoencoder** that obtains *half of the initial number of features*. For this you may use the `MLPRegressor` method from the `neural_network` collection in the `sklearn` library. Note that, while these architectures contain a number of parameters to be optimized, focus exclusively on these three:
   - `hidden_layer_sizes`: Number of neurons per hidden layer.
   - `activation`: Activation used in the hidden layer.
   - `max_iter`: Maximum number of training iterations.

T2. Gather four other assortments from your colleagues: Share with your partners the three versions—*original*, *PCA*, and *Autoencoder*—of the assortment you have prepared in **T1**. You should also get, at least, **four other collections** to complete the rest of the assignment.

T3. Classifiers: You will consider **two classifiers** for this assignment:

1. k-Nearest Neighbor classifier: The kNN rule introduced in **T4**. For the optimization of the classifier, focus exclusively on the k parameter.
2. Multi-layer Perceptron: A neural model as presented in **T5**. For this you may use the `MLPClassifier` method from the `neural_network` collection in the `sklearn` library. For the optimization of the model, focus in the three parameters studied for the *Autoencoder*: `hidden_layer_sizes`, `activation`, and `max_iter`.

Note that, for the **optimization** process, you should consider the same approach as in **Assignment 2**: considering a cross-validation scheme, obtain three disjoint data partitions—`train`, `validation`, and `test`—for then using the first two to optimize the model and the latter one to report the goodness of the classifier.

T4. Results: Keep the results obtained for each combination of **data collection**, **classifier**, and **cross-validation fold** for the final **Analysis** stage. In addition, provide the aggregated performance scores in terms of the **average** and **standard deviation** indicators for each combination of **data collection** and **classifier**.

T5. Analyze: You must consider the methods presented in the **last module** of the course. Since these concepts still have not been introduced, the precise details will be provided in the **next version** of the assignment.

Report format

The report must follow these points:

- Must use the ICML template.
- Clearly state the assignment, your name, and identification details (ID and email).
- The report must be *3 pages maximum* (double column).

Report questions

The questions will be defined in the **next version** of the assignment.

Delivery:
Submit a single ZIP file via Moodle. It must include the report and it may also contain the developed code.
Delivery date: December 23, 2025

Tips and suggestions

- Code quality is not a requirement, but the conclusions obtained.
- A Jupyter Notebook (or similar) may be adequate for the task at hand.