Group assignment
Machine Learning I

OBJECTIVE

Develop a Machine Learning (ML) system capable of solving a prediction or classification problem.

I. General Instructions:

1. **Team Formation**

This assignment must be completed in groups of **3 to 6 students**.

2. **Implementation**

For the implementation of the ML system, you should include the code developed in the Notebooks from the first part of the course.
These components must be added as **Julia modules** (i.e., Julia script files).

3. **Problem selection**

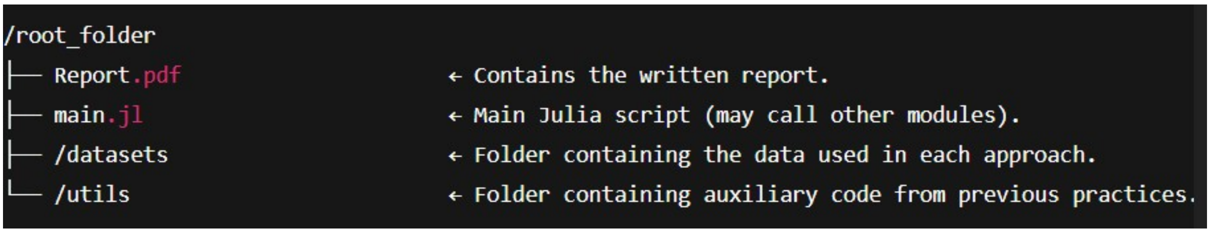
Any regression or classification problem of resonable complexity is suitable.
The selected problem must be agreed upon with the course instructor **no later than November 18th**.
The dataset used must be **publicly available** and appropriate for use with your implemented ML system.
The problem must be solved using **at least four different approaches**

4. **Submission**

The final solution must be submitted via **Moodle (Virtual Campus)** by **December 5th, 23:59**.
The deliverables must include:

- The selected dataset.
- The source code and a detailed report.
- Auxiliary code developed in previous practices.

Folder Structure (for the compressed submission file):



```
/root_folder
├── Report.pdf                ← Contains the written report.
├── main.jl                   ← Main Julia script (may call other modules).
├── /datasets                 ← Folder containing the data used in each approach.
└── /utils                    ← Folder containing auxiliary code from previous practices.
```

Report contents:

The report must clearly explain:

- The selected problem and objectives.
- The methodology and approaches followed.
- A discussion and interpretation of the results.
  (See the evaluation criteria section below for detailed requirements.)

Main.jl

The file should contain the whole code with all approaches of the project. It has to be executable from beginning to end, and it must allow to check the same results presented in the report.

Datasets:

Should contain the data used in the different approaches. Within this folder new folders can be created, and this structure is up to the decisions made by the work team.

Utils:

It should contain the code from the tutorials in the shape of one or more Julia (.jl) file(s).

5. Oral Presentation

Each group will present their project on **December 9th** (approximately **15 minutes per group, including 5 min for questions**).

II. Specific instructions and evaluation criteria

Each project will be evaluated based on three main components: **Report (50%)**, **Code (30%)**, and **Oral Presentation (20%)**.

1. Report (50% of the project mark)
   - Introduction (10% of the project mark).
     The introduction must include:
     i. A clear description of the problem to be solved.
     ii. A description and summary of the dataset used.
     iii. Justification of the evaluation metric(s).
     iv. Explanation of the code structure and organization.
     v. A short **bibliographic review** (minimum **3 scientific publications** relevant to the problem).
        1. Use a formal citation style suitable for academic publications.
        2. Websites are **not** considered scientific sources
        3. The description should connect the different works to each other, rather than being just a list of separate, unrelated paragraphs.

- Development: (30% of the project mark)
  The students are tasked with investigating various approaches for processing the dataset.
  The highest achievable score for each attempted approach will be 25% of the total value of this section.
  Each approach should include:

  i. **Dataset description:** Even if previously introduced, describe any variations used in each approach (number of samples, features, classes, etc.), supported by relevant graphs or figures.
  ii. **Data preprocessing:** Justify the normalization method and parameters used (min, max, mean, etc.) or explain why normalization was not applied.
  iii. **Experimental setup:** Specify methodology, cross-validation strategy, variable selection, dimensionality reduction, etc.
  iv. **Model experimentation:**
     Each approach must test **the four ML techniques covered in class**:
     1. **Artificial Neural Networks (ANNs):** Test at least **8 architectures** (1–2 hidden layers).
     2. **Support Vector Machines (SVMs):** Test at least **8 configurations** with different kernels and values of *C*.
     3. **Decision Trees:** Test at least **6 different maximum depths**.
     4. **k-Nearest Neighbors (kNN):** Test at least **6 different values of *k***.

     The clarity and organization of the explanations is highly valued, as well as the number of experiments.
  v. **Ensemble Method:**
     Apply at least **one ensemble technique** (e.g., majority voting, weighted voting, or stacking) combining **at least three** of the previous individual models.
  vi. **Results and discussion:**
     Report all experiments using the selected metrics.
     Include comparisons between models, supported by plots or confusion matrices or statistical significance test (ANOVA, t-test, Friedman, etc.).

- Final discussion (10% of the project mark)
  Summarize and evaluate the overall process, comparing the results across different approaches. Highlight the conclusions supported by the experimental findings.

- **Formatting:**
  Please follow the ACM Primary Article Template for the structure and formatting of your report.

2. Code (30% of the project mark)
   The code must:
   - Set a random seed to ensure reproducibility.
   - Load and preprocess the dataset.
   - Extract relevant features.
   - Split the data (training/testing)
   - On the train dataset perform a cross-validation using the provided ***modelCrossValidation*** function to choose the corresponding parameters.
   - Train the final model with the full training dataset and evaluate it on the test set.
   - Include a confusion matrix for evaluation.
     i. For ANNs, remember to use a validation split within the training data.

   Additionally, the following points will be considered in the evaluation as positive elements:
   - Use separate modules for different functions.
   - Add clear comments describing each function or module.
   - Use meaningful variable and function names.
   - Follow consistent naming conventions.

3. Oral Presentation (20% of the project mark)
   The presentation (≈15 minutes) should include:
   - A concise explanation of the problem and objectives.
   - Description of the experimental pipeline.
   - Presentation of results and comparison of approaches.
   - A short Q&A session (≈5 minutes).
   - Clarity and timing are key factors in the evaluation.

Major Penalties

Severe deductions may apply for the following:

- **Code that does not run:** −20% of the total mark.
- **Inconsistent results (code vs. report):** −10%.
- **Use of test data during training:** −20%.