SCIENTIFIC COMPUTING

Prof. Sebastián Roldán Vasco

MASTER IN AUTOMATION AND INDUSTRIAL CONTROL - 2024-2

DEADLINE: 20th November 2024

Dear student, read the following instructions carefully. Your grade will depend on their fulfillment.

GRADED WORK:

• Written report in GIT-related platform, i.e. GitLab or GitHub

• Data, scripts and functions would be self-contained.

The work must be written entirely in English

Goal:

In this assignment, students are required to use Python and Git to implement a data-driven scientific analysis. You will use your own dataset (e.g., images, signals, electrical variables, or any data relevant to control systems) and submit a Python program that demonstrates their understanding of data handling, statistical analysis, data separability, and performance evaluation.

Materials Needed:

- Computer with Python and a text editor/IDE installed.
- Git installed for version control.
- Access to a terminal or command prompt.

Activity:

1. Advanced data handling with Pandas. Use Pandas to clean, preprocess, and structure your dataset for analysis. Handle missing data, restructuring, and transformations needed to make the dataset ready for the analyses.Load your dataset and filter it for specific features or time frames

that are meaningful for analysis. Restructure the data as necessary: examples include aggregating data over time intervals, creating specific categories for variables, or adjusting formats for easier analysis (e.g., from a 3D array to 2D or vice versa if working with image data). Demonstrate handling of missing data, outliers, or inconsistencies within the dataset.

Grading Criteria:

- Proper data loading in a structured format and handling of inconsistencies (0.5 points).
- A Markdown section explaining each data handling step, with an emphasis on why these transformations were applied (0.5 points).

2. Statistical analysis of custom data and feature extraction. Conduct a statistical analysis of your dataset, focusing on identifying key properties and patterns within the data that are relevant to your dataset's context. Calculate and interpret relevant summary statistics for key variables (e.g., mean, median, variance) and distributions, and display their results visually where applicable. If working with signals, explore time-domain or frequency-domain statistics (such as RMS, spectral density). Make that the dataset includes multiple categories or classes, apply statistical tests (e.g., t-tests, ANOVA, Fisher's ratio) to determine differences across groups.Perform an analysis on how distinct or separable different groups or categories are within your dataset, both visually and numerically.

   Grading Criteria:

   - Correct calculation and extraction of features or mathematical descriptors (0.6 points).
   - Correct calculation and application of statistical measures/tests (0.8 points).
   - Clear visualizations and meaningful interpretations relevant to the dataset (0.6 points).
   - Clear interpretation of separability measurements in the context of the dataset (0.6 points).

3. Data separability. Measure performance metrics relevant to your data analysis goals. If working with labeled data, implement classification

performance metrics; if dealing with continuous or time-series data, apply metrics that assess the accuracy or relevance of your model or data transformations. For datasets with labels, calculate classification performance metrics such as accuracy, precision, recall, F1-score, or ROC-AUC for your model. For continuous or time-series datasets, define performance metrics relevant to your dataset (e.g., root-mean-square error, correlation coefficient, or spectral match) and evaluate how well data transformations or models meet these metrics.

Grading Criteria:

- Correct implementation and calculation of appropriate performance metrics (0.6 points).
- Insightful interpretation and comparison of performance results (0.8 points).

Deliverables:

GitHub repository link (or equivalent).