Scientific Programming Final Project

Alexandre Perera, Mónica Rojas
December 2025

1. Objective:

The main objective of this project is to develop a robust and accurate predictive model for use in diagnosis based on specific markers or features. To achieve this, you will work with different datasets (one per group) that are randomly assigned. Each dataset contains various features, most of which are extracted from high-dimensional data such as signals or images. Every dataset is accompanied by a short description and includes the target variable.

Please note that there may be missing values, and not all features will necessarily be required for developing your model.

Using this data, a predictive model will be trained utilizing machine learning and/or biostatistical techniques and the different tools we have been using so far.

The resulting model will be encapsulated in an API, allowing users to input markers for a given subject and receive a possible diagnosis.

2. Project Development

Before starting work on the solution, each collaborator must create a separate branch to work on. Each collaborator should be assigned specific tasks, and every team member must be responsible for at least one coding task associated with their branch. Ensure that the assigned tasks for each collaborator are clearly documented in the report.

Keep in mind that you need to coordinate and schedule your work with your teammates to complete the project on time, especially if there are dependencies in the workflow.

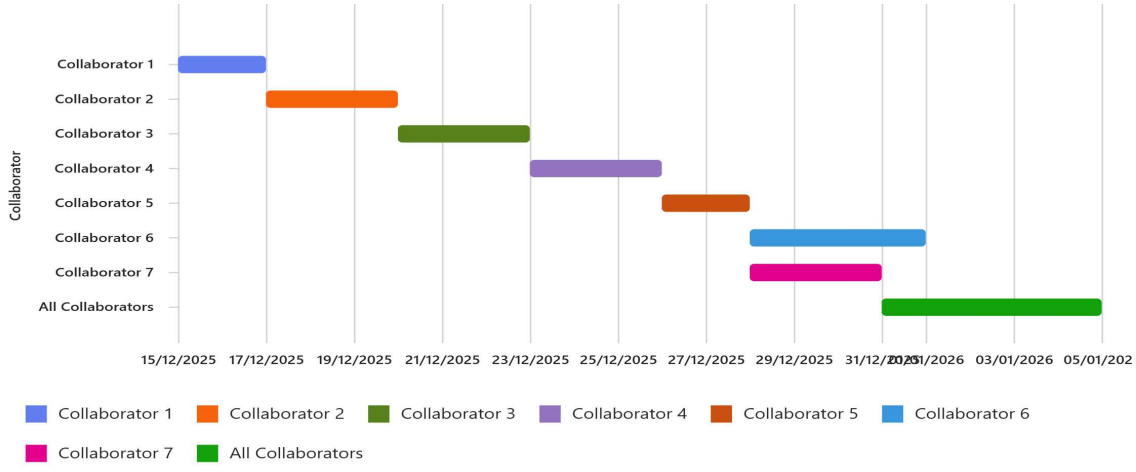
The following section provides examples on how tasks can be distributed and implemented, using Lab 4 as basis.

2.1. Example: Workflow and Distribution for achieving Early Pakinsons's Diasease

Collaborator	Tasks	Inputs / Outputs
1	Create a function to rename DataFrame columns using dict_names. Summarize cleaned data and annotate observations (e.g., outliers, variability).	**Inputs:** df, dict_names **Outputs:** renamed_df, summary notes
2	- Aggregate variables across trials using group_and_average. - Create scatterplot function for two variables grouped by category. - Generate subject_id and trial columns from name and remove name.	**Inputs:** cleaned_dataframe, var1, var2, groups **Outputs:** Aggregated DataFrame, scatterplot
3	- Normalize all variables in df using z-score or min-max scaling (justify choice). - Analyze correlations among fundamental	**Inputs:** Original DataFrame **Outputs:** Normalized DataFrame, cleaned_df

	frequency, Jitter, and Shimmer variables; keep representative ones and remove others.	
4	- Write function to aggregate variables by grouping variable using pandas groupby. - Implement classification (e.g., KNN or other model) to distinguish patients vs. controls.	**Inputs:** df, gv **Outputs:** Aggregated DataFrame, classification results
5	- Implement validation strategy (train/test split or cross-validation). - Select best model for API implementation based on validation results.	**Inputs:** Training and test sets **Outputs:** Validation metrics, selected model
6	- Design and implement API using FastAPI or Plumber. - Define endpoints for input markers and prediction output. - Integrate trained model into API and add documentation/tests.	**Inputs:** Final model, feature schema **Outputs:** Functional API with documentation
7	- Containerize API using Docker (create Dockerfile). - Deploy API on a server (cloud or local). - Test deployed API and document deployment steps.	**Inputs:** API code, Docker configuration **Outputs:** Docker image, running API on server
All	Improve and select the best predictive model, write the report and record the video	**Inputs:** Several different models, API **Output:** Report and video of your work

Tasks workflow:



3. Evaluation

For the evaluation, you must share the repository created for the project (one per group). The project should have the same number of branches as there are collaborators. The Git history will be reviewed to track each team member's contributions. Please ensure the final repository is publicly accessible.

You are required to prepare a report not exceeding 5 pages, which must include:

- The objective of the project
- The distribution of tasks
- The analysis and key results
- The conclusions of your work

Remember to add a link to your repository in the report.

Finally, create a 1-minute video demonstrating the deployment of the API.