Scientific Programming Final Project

Alexandre Perera, Mónica Rojas
January 2025

1. Objective:

The main objective of this project is to develop a robust and accurate predictive model for early diagnosis based on specific markers. To achieve this, you will work with the "ParkinsonsProject" dataset, which contains various features extracted from voice recordings that can serve as potential markers for the early diagnosis of this condition.

Using this data, a predictive model will be trained utilizing advanced machine learning and/or biostatistical techniques. 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

For this assignment, you will be working in groups of 5 so you need to identify and get in contact with your partners. The groups are already created in the virtual campus and are configured as to allow you to send messages between group members.

You need to create a private repository in GitHub. Select one team member to be the leader. The leader will be in charge of creating the main branch and allocate the necessary files on it. He will also invite other members to join the project. We encourage you to do this task together as well as to plan the tasks focusing on specific dates so you complete the work on time.

Your tasks for this assignment involve writing sections of code and/or developing various functions in Python or R. In order to track the work from each member, please implement the functions in separated files to be called from the main. Keep in mind that if you use python you may need to import some modules inside the functions so it can properly work (even when the modules are imported in the main).

Before start working on the solution, each collaborator must create a branch to work on. Each collaborator should be assigned with specific tasks and each member should be responsible for at least one coding task associated with a specific branch. Make sure you include the assigned tasks for each collaborator in the report. Keep in mind that you need to schedule your work with your team mates so you can finish the project in time.

In the following, there are some examples on how the tasks can be distributed:

Collaborator 1

- Create a function that rename variables on a given dataframe *df.* The function returns a *df* who's columns names are as detailed in *dict_names*
  Inputs:
  df: a given data frame
  dict_names: a dictionary for mapping the actual names of the columns in the dataframe (each key of the dictionary) to a given new simpler name (i.e. the values of the dictionary). You can follow the example in the notebook *numpy_pandas_sklearn.ipynb* for the Lab 4.
  Output:

renamed_df: returns the input dataframe but with the columns renamed as in *dict_names*

Please work with a version of the dataset that uses simple names for the rest of the analysis

- Summarize the data after cleaning (that is, after removing some correlations) when the data is ready. Annotate your observations. For example, how many observations do you have? Are there apparent differences between controls and patients? Is the variability comparable? If you check the minimum and maximum values are there outliers? If so, what will you do with them?

Collaborator 2

- Using the function *group_and_average, provided by* Collaborator 1 create a dataframe for aggregating each variable of the cleaned_dataframe across trials for each subject. The resulting dataframe should consist on 32 observations, one for each subject.
- Write a function that given tow variables, *var1* and *var2* and a grouping variable *groups* creates a scatterplot of the two variables, displaying the information associated with *groups* using different colors (or symbols). That is, observations belonging to a given group 1 will be displayed on a given color, observations belonging to group 2 will be displayed with a different color and so on. *var1* is displayed in the *x-axis* and *var2* in the *y-axis.* The obtained plot should contain a legend displaying the information regarding to *groups*

- Based on the variable *'name'* write some code in the main that creates two new columns in the dataframe: one displaying the subject id (S1, S2, S3…) and another displaying the trial (t1, t2, t3, …, t6 or t7 in some cases). Remove the column name from the dataframe.

Collaborator 3:

- Write a function that given a dataframe *df*, normalizes all variables according to the using a z-score or a min-max (for example). You should select the normalization method that best fit your data (remember to justify your choice in the report). The function returns a dataframe consisting on normalized variables.

- Look for correlations between the variables related to the fundamental frequency ('MDVP:Fo(Hz)', 'MDVP:Fhi(Hz)' and MDVP:Flo(Hz) in the original dataframe) using the *scat_plot* function. Use the scat_plot function created by your collaborators for this purpose. If the variables are correlated keep only a subset that are representative (one or some of them). Discard the others by removing them from the dataframe. Do the same for the variables related to Jitter ('MDVP:Jitter(Abs)', 'MDVP:RAP', 'MDVP:PPQ', 'Jitter:DDP') and Shimmer ('MDVP:Shimmer', 'MDVP:Shimmer(dB)', 'Shimmer:APQ3', 'Shimmer:APQ5', 'MDVP:APQ', 'Shimmer:DDA',). Name this dataframe as *cleaned_df* and work with it from here on.

Collaborator 4:

- Write a function that given a dataframe *df* and a grouping variable *gv,* average all variables on a given dataframe by aggregating them according to the variable *gv*. You can use the group_by operation from pandas. The function returns the averaged and aggregated dataframe.

- Implement the instructions in the main code to classify the data into patients or controls using a k-nearest neighbors or another classification model of your choice.

Collaborator 5:

- Implement a strategy for validating the prediction model. One possible solution is to divide the data set into training (70%) and test (30%) sets for training the model with the first and validate the results of the prediction model with the second. You can also implement a cross-validation approach
- Based on your validation results, select a model for implementing it in an API. The API can be implemented using for example FastAPI (python) or Plumber (R)

For the report, remember to compare the outcome of the prediction model across the following scenarios: 1) utilizing cleaned and aggregated data and 2) cleaned, aggregated, and normalized data. This will help you to select the best model

3. Evaluation

For the evaluation, you need to share the repository you created for the project (one per group). The project must have the same number of branches as there are collaborators. The history of changes in GIT will be reviewed to track the contributions of each team member. Please share the final repository with the GitHub account "monirojas" for evaluation.

The report should not exceed 5 pages and must include the distribution of tasks, the purpose of the analysis, and the key results. Remember to add the conclusions of your work. Finally, prepare a 1-minute video demonstrating the deployment of the API.