Final Project

Data Analysis in Rehabilitation

Prof. Joan F. Alonso

Master in **Neuroengineering and Rehabilitation**

2024/25

Contents

1 Introduction and Aim 1
1.1 Some context . . . 1
1.2 Key points . . . 1

2 Project Phases and Tasks 2
2.1 Proposed Planning . . . 2
2.2 Phase 1: Exploratory Data Analysis . . . 2
2.3 Phase 2: Feature Engineering and Selection/Extraction . . . 3
2.4 Phase 3: Model Training and Validation . . . 6
2.5 Phase 4: Final Model Assessment and Reporting . . . 7

3 Deliverables 8
3.1 Outputs of the Project . . . 8
3.2 Final Report . . . 8
3.2.1 Contents . . . 8
3.2.2 Delivery . . . 9
3.3 Presentation . . . 10
3.3.1 Contents . . . 10
3.3.2 Delivery . . . 10
3.4 Grading . . . 10

4 Collaboration and Expectations 10

5 Acknowledgements 11

A Description of the Database 12
A.1 Channel Information . . . 12
A.2 Experimental Protocol . . . 12
A.3 Trial Structure . . . 12

B Project Report Rubric 13

C Presentation Rubric 15

1 Introduction and Aim

1.1 Some context

Welcome to the integrative data analysis project! Over the next weeks, we will embark on a journey into the field of Brain-Computer Interfaces (BCIs). We will analyse real electroencephalography (EEG) data recorded from volunteers performing motor imagery tasks. Specifically, volunteers imagined moving either their left or their right hand, or they remained in a relaxed state (no movement).

We will start with raw data, navigate the challenges of biological signal processing, explore feature engineering techniques, and apply machine learning models to decode the brain's intentions. This project provides hands-on experience with a complete data analysis workflow in a challenging and exciting application domain.

In the *Human-Machine Interfaces (HMI)* course we have already explored the decoding of upper-limb movements using the database described in P. Ofner et al., 2017. Specifically, we now know how to:

- **Band-pass filter EEG signals** and identify bad channels and artefacts.
- **Select usable trials** using trimmed normalisation.
- **Re-reference** signals (common average, Laplacian, etc.).
- **Assess movement execution** on the EEG signals by event-related de/synchronisation (ERD/ERS) and slow cortical potentials (SCP).
- **Represent topographic maps** to check the channels where these potentials are best observed.

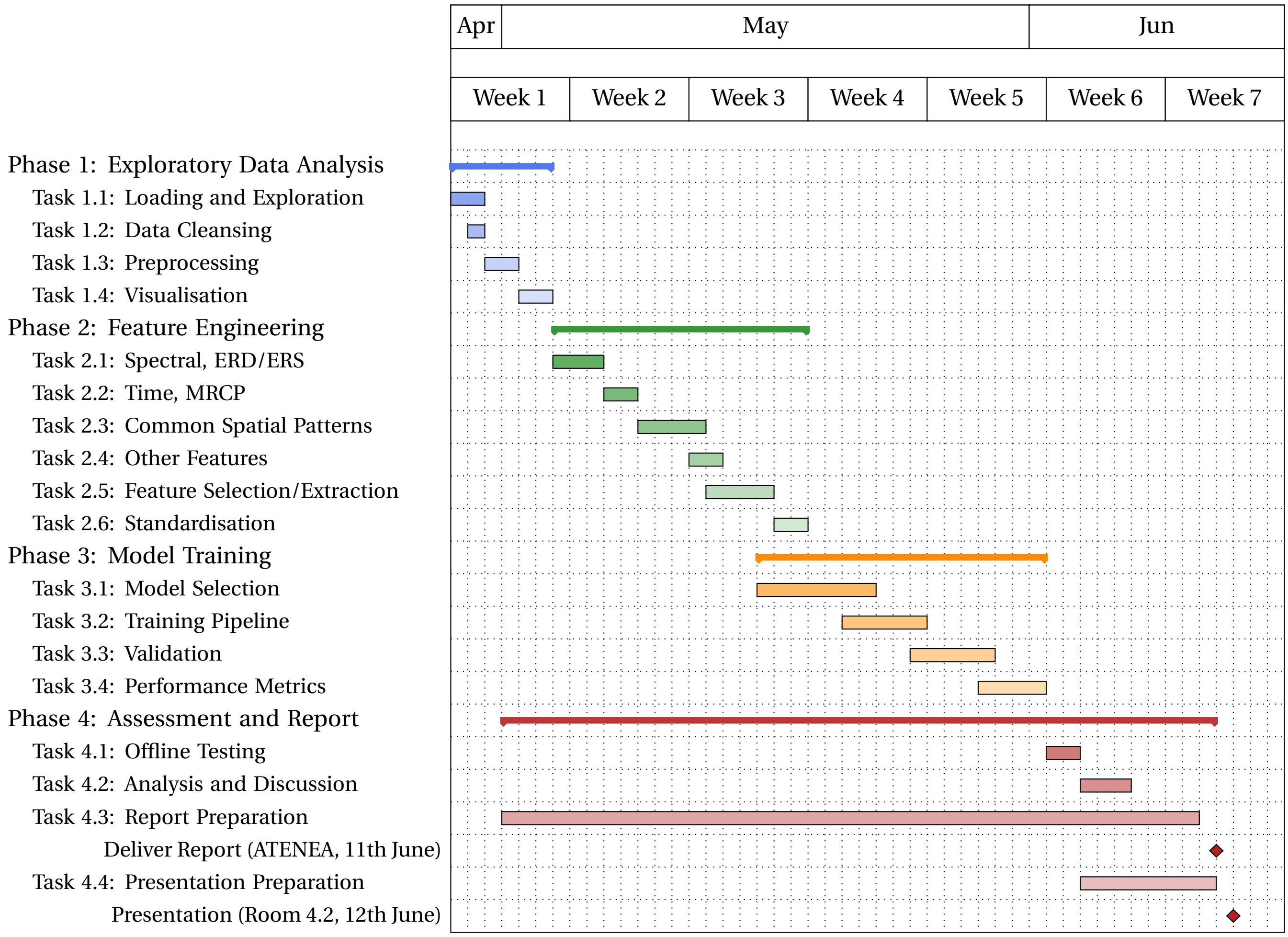
Therefore, in this project we will use this knowledge to decipher the movement imagined by a person.

1.2 Key points

- **Primary aim of the project**
  To design, implement, and evaluate a machine learning pipeline capable of accurately classifying different motor imagery states (left-hand, right-hand imagery, no imagery) based solely on the recorded EEG signals.
- **Duration**
  6 ½ weeks (from 28th April to 11th June, 2025)
- **Group Size**
  4 students
- **Grading**
  70 % Report, 30 % Presentation.
  Details on the deliverables are in Section 3.2. The rubrics used to assess the project are available in Annexe B and Annexe C.

2 Project Phases and Tasks

2.1 Proposed Planning



2.2 Phase 1: Exploratory Data Analysis

- **Objective:** Familiarise yourselves with the EEG dataset and prepare it for analysis.
- **Tasks:**
  - **Data Loading and Exploration:** Load the provided EEG datasets (in EEGLab `.mat` format). Understand the data structure, channel layout (based on the 16-channel OpenBCI setup), and event markers indicating the different imagery tasks.

    Annexe A describes the contents and format of the shared files.
  - **Data Cleansing:** Identify and handle potential issues like noise, artefacts (e.g., blinks, muscle activity), and bad channels. Decide on appropriate strategies for correction or removal.
  - **Preprocessing:** Apply necessary preprocessing steps such as filtering (e.g., bandpass filtering to isolate relevant frequency bands), referencing, and epoching (segmenting the data into trials based on task markers).
  - **Data Visualisation:** Create informative visualisations (e.g., time series plots, power spectral density plots, topographic plots) to understand the characteristics of the EEG signals during different tasks and to guide preprocessing choices.
- **Comments:** It may be easier to do this in MATLAB, reusing code developed in *HMI*.

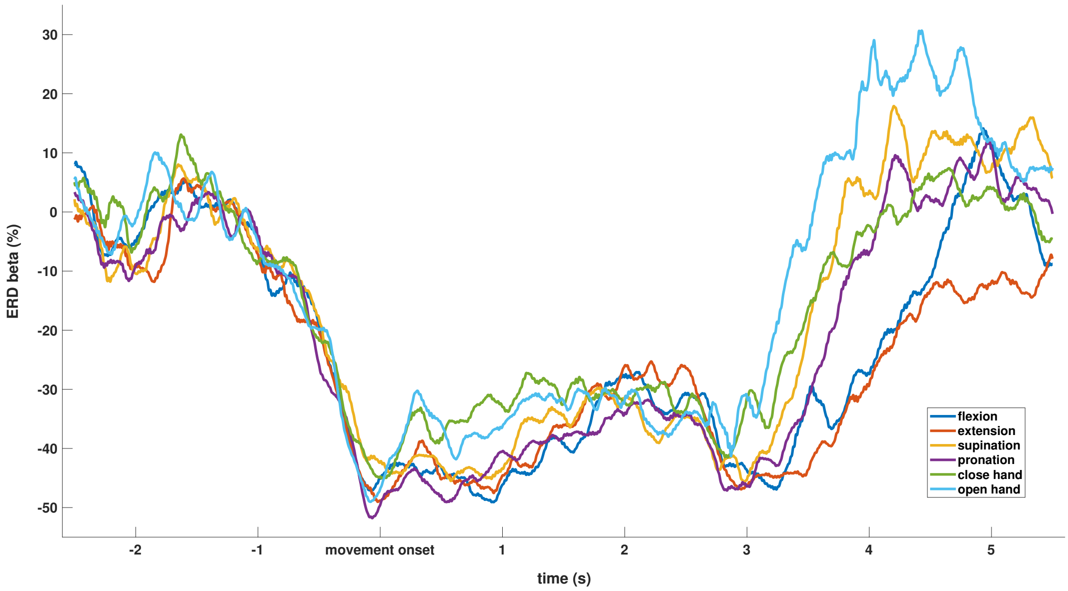


Figure 1: Grand-average ERD/ERS potentials obtained in the β band for the movements in P. Ofner et al., 2017. Image by S. Romero.

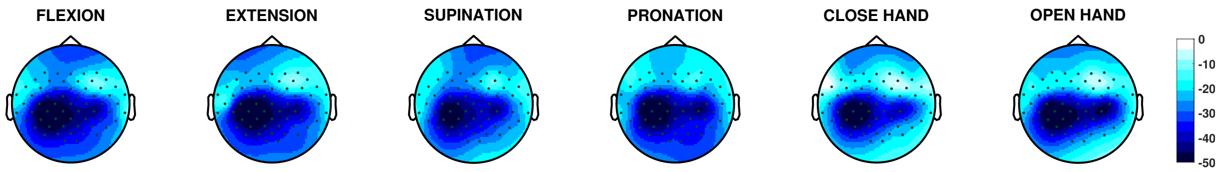


Figure 2: Grand-average ERD potentials (in µV) obtained in the β band at the movement onset, for the movements in P. Ofner et al., 2017. Image by S. Romero.

2.3 Phase 2: Feature Engineering and Selection/Extraction

- **Objective:** Create meaningful features from the preprocessed EEG data that can aid discrimination between the different motor imagery states.
- **Tasks:**
  - **Brainstorming and Feature Engineering:** Explore and implement various feature types relevant to motor imagery EEG, including:
    - **Frequency-based features:** Investigate ERD/ERS patterns, typically observed in the μ (8 Hz to 13 Hz) and β (14 Hz to 30 Hz) bands over motor cortex areas during motor imagery (See Figure 1 and Figure 2). Calculate band power or related metrics.
    - **Time-domain features:** Explore Movement-Related Cortical Potentials (MRCPs), although these are often more prominent in actual movement, investigate if relevant components exist in the imagery data. These potentials are usually best detected when the signal is low-pass filtered (e.g. from 0.3 Hz to 3 Hz). Keep in mind that slow cortical potentials are best detected around central locations (Cz and neighbouring electrodes), as seen in the maps in Figure 3.
    - **Common Spatial Patterns (CSP):** Implement and utilise CSP, a powerful technique for BCI that finds spatial filters maximising variance between classes. Experiment with the number of patterns/filters. Additional insights on CSP in this article by B. Blankertz et al. 2007.
    - **Other potential features** (e.g., statistical measures, connectivity metrics).
  - **Feature Analysis and Selection/Extraction:**
    - Analyse the discriminative power of individual features or feature sets.

    * Use techniques like training and validation curves to assess potential overfitting or underfitting with different feature sets.
    * Evaluate feature redundancy (e.g., using correlation matrices).
    * Decide whether to use feature selection methods (choosing a subset of the best features) or feature extraction methods (like PCA or potentially using CSP itself as a dimensionality reduction technique). Justify your choices.
  - **Standardisation/Normalisation:** Apply appropriate scaling techniques to ensure features are comparable.

- **Comments:**
  - The articles by D. J. McFarland et al., 2005 and by S. Vaid et al., 2015 can assist in discovering other useful features in BCI systems.
  - Feature engineering may also be easier in MATLAB, adapting *HMI* codes. CSP can also be calculated easily using MATLAB. A Python alternative (MNE-Python package), should you wish to code everything in Python, is also very useful (see example).
  - Below (see pages 4 to 6) you can find a basic example of:
    * a code structure to process signals and calculate features;
    * how to use a MATLAB table to store features;
    * how to export a table to a `.csv` file.

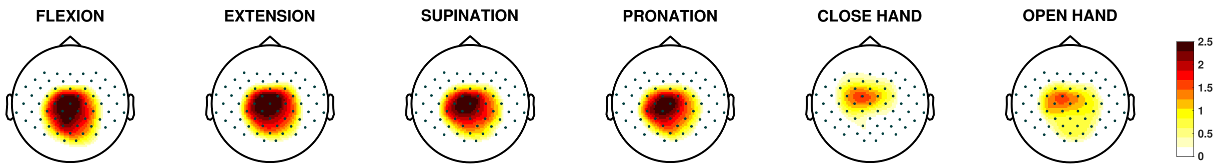


Figure 3: Slow cortical potential topographies for the movements in P. Ofner et al., 2017. Maps show the grand-average amplitude in μV of a 0.5 s segment selected just before the movement onset. Image by S. Romero.

MATLAB Code to process all trials of a subject

```
%% START & CLEANUP
clearvars; close all; clc;

%% LOAD EEG SIGNALS FROM A VOLUNTEER
filename = 'v01.mat';
load(filename);  % Load EEG variable, EEGLab format

%% DEFINITIONS
num_events = numel(EEG.event);  % 960 --> 480 pairs of fixation cross (c) followed by task
                                %          First 120 trials are motor execution
                                %          Next 360 are motor imagery. Each trial is labeled:
                                %           - b indicates resting (baseline)
                                %           - r indicates right
                                %           - l indicates left
Fs = EEG.srate;  % 125 Hz
signals = detrend(EEG.data(1:15,:)', 'constant');  % 16 columns (but ch 16 unused)
% Ignore EEG labels in the mat-file. These are the channels in order:
ch_names = {'F7','F3','Fz','F4','F8','T3','C3','Cz','C4','T4','P7','P3','Pz','P4','P8'};
first_trial = 121;


%% FILTER DEFINITIONS
% Mu+Beta filter (8 to 30 Hz)
Fstop1 = 7.9;          % First Stopband Frequency
Fpass1 = 8;            % First Passband Frequency
```

```
Fpass2 = 30;          % Second Passband Frequency
Fstop2 = 30.1;        % Second Stopband Frequency
Astop1 = 20;          % First Stopband Attenuation (dB)
Apass  = 0.1;         % Passband Ripple (dB)
Astop2 = 20;          % Second Stopband Attenuation (dB)
match  = 'passband';  % Band to match exactly
h  = fdesign.bandpass(Fstop1, Fpass1, Fpass2, Fstop2, Astop1, Apass, Astop2, Fs);
mubeta_filter = design(h, 'cheby2', 'MatchExactly', match, 'SystemObject', true);
reorder(mubeta_filter, 'up');


%% EPOCHING TRIALS AND CONCATENATION
mi_right = [];
mi_left = [];
mi_rest = [];
mi_right_std = [];
mi_left_std = [];
mi_rest_std = [];

for ev = first_trial*2 : 2 : num_events
    start_sample = EEG.event(ev).latency;
    stop_sample = EEG.event(ev).latency+2*Fs-1;

    % Display trial latency
    fprintf('Ev %3d [%s]--> %6d (%8.3f)\n', ...
        ev, upper(EEG.event(ev).type), start_sample, start_sample/Fs);

    trial = detrend(signals(start_sample:stop_sample, :), 'constant');
    pad_trial = [flipud(trial); trial; flipud(trial)];
    filt_pad_trial = filtfilt(mubeta_filter.SOSMatrix, mubeta_filter.ScaleValues, pad_trial);
    filt_trial = filt_pad_trial(size(trial,1)+1:2*size(trial,1), :);

    % Plot to check that filtered signals look OK
    if ev == first_trial*2
        figure; hold on; box on;
        plot(1/Fs:1/Fs:length(trial(:,1))/Fs, trial(:, strcmp(ch_names, 'Cz')));
        plot(1/Fs:1/Fs:length(trial(:,1))/Fs, filt_trial(:, strcmp(ch_names, 'Cz')));
        legend('Raw Cz (detrended)', 'Filtered Cz');
        ylabel('amplitude (\muV)', 'Interpreter','tex');
        xlabel('time (s)')
        title(['EEG (' filename ')'])
        axis tight;
    end

    % Calculations performed trial by trial:
    %  - Features (for example, standard deviation of channel Cz):
    %  - Concatenate trial to corresponding matrix (for example for later SCP calculation)
    switch EEG.event(ev).type
        case 'r'
            mi_right_std = [mi_right_std; std(filt_trial(:,strcmp(ch_names, 'Cz')))];
            mi_right = [mi_right; filt_trial];
        case 'l'
            mi_left_std = [mi_left_std; std(filt_trial(:,strcmp(ch_names, 'Cz')))];
            mi_left = [mi_left; filt_trial];
        case 'b'
            mi_rest_std = [mi_rest_std; std(filt_trial(:,strcmp(ch_names, 'Cz')))];
            mi_rest = [mi_rest; filt_trial];
        otherwise
            error('This should not happen!')
    end
end

%% COMBINE VECTORS INTO TABLE & EXPORT CSV
% Use only 4 values of each task (instead of 120), just for the example:
```

```
df = table([mi_rest_std(1:4); mi_left_std(1:4); mi_right_std(1:4)] , ...
            [repmat("rest",4,1); repmat("left",4,1); repmat("right",4,1)], ...
           'VariableNames', {'std_8_30', 'mi_task'});
writetable(df, 'features.csv')
```

Exported 'features.csv' file

```
std_8_30,mi_task
0.6183329,rest
0.6678268,rest
0.5968277,rest
0.5260289,rest
0.6791871,left
0.7106078,left
0.5141629,left
0.7058661,left
0.7243008,right
0.6787799,right
0.6231062,right
0.596122,right
```

2.4 Phase 3: Model Training and Validation

- **Objective:** Train and rigorously evaluate different machine learning models for the classification task.
- **Tasks:**
  - **Model Selection:** Choose at least 2 or 3 different classification algorithms suitable for this type of data (e.g., Linear Discriminant Analysis (LDA), Logistic Regression (LR), k-Nearest Neighbors (KNN), Support Vector Machines (SVM), or even ensemble methods like Random Forests or Gradient Boosting).
  - **Training Pipeline:** Develop a clear pipeline that incorporates your chosen preprocessing, feature engineering/selection, and classification steps.
  - **Validation:** Implement a robust validation strategy (e.g. hold-out, k-fold cross-validation) to evaluate model performance and tune hyperparameters reliably. Avoid testing on the left-out subject during this phase.
  - **Performance Metrics:** Select and calculate appropriate performance metrics (e.g., accuracy, precision, recall, F1-score, confusion matrix) to compare the models effectively.
- **Comments:** In this part you will be using Python (importing `.csv` files generated by MATLAB), as we have done throughout the course.

  Model selection will also depend on your feature selection/extraction strategy. For example, if you use a *wrapping* approach and change the model, you may need to run feature selection for each model. If you choose a *filtering* approach, feature selection is model-independent, but less optimised for the chosen model.

  Keep in mind that, in this project, there is no "best" or "correct" way of choosing a validation strategy. You are free to try any alternative:

  - All but one subject to train, **leaving one out** to test each time (selecting the best model as the one with best average performance),
  - Implement a **hold-out** strategy by training with some subjects and testing with the rest,
  - Training a model on each subject, leaving out some data to test each time (**stratified hold-out**), and then choosing the model also based on average performance.
  - **Or something completely different** that you may come up with (perhaps devising an original approach). The primary consideration here is to choose a strategy, interpret and understand the results obtained, and propose changes or improvements depending on them.

Also, take into account that we are dealing with a **3-class classification problem**, but it can be divided into **two 2-class classification problems**: rest vs motor imagery and then left vs right. In this case, discuss the suitability of each metric for this potentially imbalanced problem.

2.5 Phase 4: Final Model Assessment and Reporting

- **Objective:** Evaluate your best-performing model(s) on unseen data and prepare your final report and presentation.
- **Tasks:**
  - **Offline Testing:** Apply your final, trained model pipeline(s) to a separate, unseen test dataset (the left-out subject or your own recordings using the same protocol and OpenBCI headset). Report the performance on this dataset.
  - **Analysis and Discussion:** Analyse your results, discuss the strengths and weaknesses of your approach, compare the performance of different features and models, and reflect on challenges encountered.
  - **Final Report and Presentation:** Prepare a comprehensive report detailing your entire workflow, methodology, results, and conclusions. Deliver a group presentation summarising your project.
- **Comments:** More details on the deliverables in Section 3.

3 Deliverables

3.1 Outputs of the Project

This project is expected to produce three outputs:

- Well-documented code implementing the full analysis pipeline. Predictably, MATLAB and Python code (either code files or Jupyter Notebooks).
- A final written report.
- A group presentation summarising the project and findings.

These three items will be included in 2 deliverables, described in Section 3.2 and Section 3.3.

3.2 Final Report

3.2.1 Contents

Your final report should be a clear, concise, and well-structured document detailing your data analysis project. Please ensure it includes the following sections:

- **Introduction:**
  - Clearly state the problem or research question being addressed.
  - Explain the motivation and importance of the analysis.
  - Define the specific objectives of your project.
  - Briefly outline the report's structure.
- **Data Description:**
  - Identify the data source(s).
  - Provide an overview of the data (variables, size, structure).
  - Detail all data preprocessing steps (cleaning, handling missing values, transformations, feature engineering) and justify why they were necessary.
- **Methodology:**
  - Describe the analytical techniques and models used (e.g., EDA, statistical tests, classification).
  - Justify your choice of methods based on the project objectives and data characteristics.
  - Mention key software or libraries used.
- **Results:**
  - Present your main findings objectively.
  - **Crucially:** Include **meaningful figures and tables** that are clearly labelled (titles, axes, legends) and have descriptive captions explaining what they show.
  - Reference all figures and tables within the text (e.g., "Figure 1 shows...").
  - Provide **accompanying comments** that explain the key insights derived from each figure and table. Do not just present visuals without explanation.
- **Discussion:**
  - **Interpret** the results – explain what they mean in the context of your problem/question.

  - Relate your findings back to the initial objectives.
  - Discuss any limitations of your data or methodology.
  - Consider the implications or potential applications of your findings.

- **Conclusion:**
  - Summarise the primary conclusions drawn from your analysis.
  - Directly answer the initial research question based on your findings.
  - Optionally, suggest potential areas for future work.

- **References:**
  - List all cited sources (datasets, articles, tools) using a consistent citation style.

- **Annexes:**
  - Include supplementary material (e.g., code snippets, extra plots) if it supports the main report but is too detailed for the body.

Overall Expectation: The report should demonstrate a clear understanding of the problem, thoughtful application of data analysis techniques, and the ability to derive and communicate meaningful conclusions supported by well-presented evidence (figures, tables, and insightful commentary).

3.2.2 Delivery

- The report is expected to be delivered via ATENEA **before the end of 11th June**.
- The document should be **one PDF document**. Additional files or information that cannot be included as annexes (such as `.csv` files or `.ipynb` notebooks), will be added in an **additional `.zip` file**. The **maximum allowed size** for each file is **200 MB**.
- **Do not include code as screenshots or images, code must be text that can be copied and pasted**. The use of LaTeX is highly recommended. You may use this document as a reference if you are unsure where to start; the LaTeX source is available here.

Figure 4: "To the Dark Side, a Jedi turns, hmmm,
each time a code screenshot to a document you paste, yes." — Master Yoda

3.3 Presentation

3.3.1 Contents

Your presentation, similarly to the report, should be a clear, concise, and well-structured document summarising your project. It can have a similar structure to the report, but it is not compulsory to follow it.

3.3.2 Delivery

- The presentation document is expected to be delivered via ATENEA **before the end of 11th June**.
- The document should be **one PDF document**. No additional files will be accepted. The **maximum allowed size** for the file is **200 MB**.
- Each group of students will have a slot of 10 min to 15 min to present their projects **on 12th June, just after the final exam**.
- You may use a **PDF or Powerpoint** file and the computer in the room for presenting (recommended), or your own laptop if you need different software.

3.4 Grading

The final score of the project will be a weighted average:

- 70 % corresponds to the delivered documents (Project PDF, Presentation PDF).
- 30 % corresponds to the group presentation.

Check all the criteria used to assess the project in Annexe B, and to assess the presentation in Annexe C.

4 Collaboration and Expectations

This is a group project, and effective collaboration is key. Ensure clear communication, equitable distribution of tasks, and regular meetings. All group members are expected to contribute significantly to all phases of the project. We encourage you to explore, experiment, and be creative in your approach while grounding your decisions in sound data analysis principles.

5 Acknowledgements

The data in this project has been acquired and curated by Eng. Jan Pinyol, under the supervision of Prof. Dr. Sergio Romero, and with help from MSc. Eng Oziel R. Cantú. Jan and Sergio have also contributed several figures and code examples.

Profs. Dr. Alejandro Bachiller, Dr. Mónica Rojas, and Dr. Sergio Romero have also contributed with their experience in BCI and teaching in the HMI course, as well as reviewing this document.

I also want to extend my gratitude to the volunteers at the BIOART Group for their selfless contribution to our EEG recording study.

Thanks to you all!

Joan F. Alonso
Barcelona, April 2025

Annexe A Description of the Database

The EEG data for this project were acquired using a 16-channel OpenBCI system, with the EEG signals referenced to the right mastoid.

EEGLab was employed solely to read the data files generated by LabRecorder, allowing for the alignment of EEG data with temporal markers from the Psychopy experiment. No additional common referencing was performed in EEGLab (although the `.mat` files may indicate the contrary by default).

A.1 Channel Information

- The channel labels in the `.mat` data files contain the raw output from the OpenBCI system.
- Data transmission was managed via a command-line interface, which also facilitated the configuration of data transmission parameters, including amplifier gain. However, direct in-software renaming of channels is not currently supported by the OpenBCI software, nor the suppression of unused channels.
- The validated channel labels for analysis are, from 1 to 15: F7, F3, Fz, F4, F8, T3, C3, Cz, C4 , T4, P7, P3, Pz, P4, and P8.
- A 16th channel is present due to amplifier configuration and must be excluded from subsequent analysis.

A.2 Experimental Protocol

The experimental protocol involved **8 runs of 60 trials** each. Within each run, there were **20 trials of right-hand task**, **20 trials of left-hand task**, and **20 trials of rest**, which appeared in random order.

The first **two runs consisted of motor executions (ME)**, while the remaining **six runs involved motor imagery (MI)**. This resulted in a total of 480 trials (8 runs × 60 trials), distributed as 160 trials for each condition (right-hand, left-hand, rest).

Thus, for motor imagery (MI) classification, data start at the 121st trial. This subset comprises 360 MI trials (6 runs × 60 trials), with 120 trials per condition (right-hand, left-hand, rest).

A.3 Trial Structure

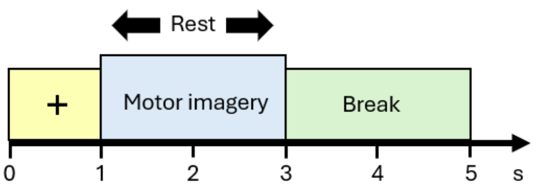


Figure 5: Paradigm used in the motor imagery trials. Image by J. Pinyol.

The experiment was programmed using PsychoPy. Each trial had an approximate duration of 5 s and followed this structure (see Figure 5):

- **Fixation Cross:** 5 s + random duration in the range of ±0.2 s.
- **Task Period:** 2 s. Here the volunteer was shown either an arrow indicating right or left (for motor imagery), or the word *Rest*.
- **Inter-Trial Interval:** 2 s + random duration in the range of ±0.2 s.

Annexe B Project Report Rubric

Table 1: Project Report Rubric

Aspect to Evaluate	Level 4 Outstanding	Level 3 Commendable	Level 2 Satisfactory	Level 1 Insufficient	Level 0 No Evidence	Weight
1 **Introduction and Problem Description**	Provides a comprehensive and insightful introduction, clearly defining the problem, strong motivation, precise objectives, and a clear report outline. Context is exceptionally well-established.	Provides a clear introduction, defining the problem, motivation, objectives, and report outline. Context is well-established.	Provides a basic introduction, outlining the problem, motivation, objectives, and report structure. Some elements may lack detail or clarity. Context may be partially missing.	Introduction is unclear, incomplete, or significantly flawed. Problem, motivation, objectives, or structure are poorly defined or missing.	No introduction or problem description provided.	10%
2 **Structure, Format, and Language**	Exceptionally well-organised report following all required sections. Professional formatting, excellent figures/tables integration. Language is precise, clear, concise, and error-free academic English.	Well-organised report adhering to structure. Good formatting and integration of visuals. Language is clear, mostly concise, and uses appropriate academic English with minimal errors.	Generally organised report, most sections present. Formatting is adequate but may have inconsistencies. Language is understandable but may lack conciseness or contain several errors.	Poorly organised, sections missing or illogical. Formatting is inconsistent or unprofessional. Language is unclear, verbose, or contains significant errors.	Report structure/ format makes it unreadable or unevaluable.	15%
3 **Methodology and Tool Usage (MATLAB, Python)**	Detailed, clear, and well-justified description of all steps (data description, preprocessing, feature engineering/ selection, model choice, validation strategy). Excellent and appropriate use of MATLAB/Python demonstrated.	Clear description and justification of most methodological steps. Appropriate use of MATLAB/Python for the required tasks demonstrated.	Basic description of methodology, some justifications may be weak or missing. Adequate use of MATLAB/Python, but perhaps not fully leveraged or with minor issues.	Methodology is poorly described, illogical, or inappropriate for the problem. Tools (MATLAB/Python) are used incorrectly or insufficiently.	No methodology described or tool usage shown.	15%
4 **Results: Metrics and Visualisations**	Presents comprehensive and insightful results using highly relevant metrics. Figures/tables are exceptionally clear, well-labelled, informative, perfectly integrated, and expertly explained in the text.	Presents clear results using appropriate metrics. Figures/tables are clear, well-labelled, adequately integrated, and well-explained in the text.	Presents basic results, some relevant metrics might be missing. Figures/tables are present but may lack clarity, labels, or sufficient explanation/ integration in the text.	Results are unclear, irrelevant, or poorly presented. Metrics are inappropriate or missing. Figures/tables are confusing, poorly labelled, or absent.	No results presented.	15%
5 **Discussion and Critical Interpretation**	Provides insightful and deep interpretation of results, critically analysing strengths/weaknesses. Effectively relates findings to objectives and BCI context. Discusses implications and limitations thoroughly.	Provides clear interpretation of results, analysing strengths/weaknesses. Relates findings to objectives and context well. Discusses implications and limitations adequately.	Provides basic interpretation of results. Some analysis of strengths/weaknesses present. Connection to objectives/context is made but may be superficial. Basic discussion of limitations.	Interpretation is minimal, incorrect, or superficial. Little to no analysis of strengths/weaknesses or connection to objectives/context.	No discussion or interpretation provided.	15%
6 **Conclusions and Originality**	Draws strong, well-supported conclusions directly addressing all objectives. Offers significant original insights, novel approaches, or critical reflections beyond the baseline requirements.	Draws clear conclusions addressing the main objectives. May offer some original insights or thoughtful reflections on the process or results.	Draws basic conclusions related to objectives, but may lack depth or full support from results. Limited evidence of original thought or reflection.	Conclusions are weak, unsupported, irrelevant, or missing. No evidence of originality or critical reflection.	No conclusions provided.	15%

Continued on next page

Table 1: Project Report Rubric (Continued)

Aspect to Evaluate	Level 4 Outstanding	Level 3 Commendable	Level 2 Satisfactory	Level 1 Insufficient	Level 0 No Evidence	Weight
7 **Code: Clarity, Comments, and Functionality**	Code is exceptionally clear, well-structured, efficiently written, and thoroughly documented (comments, README). It runs flawlessly and reproduces the results described. Best practices are followed.	Code is clear, reasonably structured, and adequately documented. It runs correctly and reproduces the main results. Good practices are generally followed.	Code is generally understandable but may lack structure or sufficient documentation. It runs but may require minor fixes or clarifications to reproduce results. Basic coding practices followed.	Code is difficult to understand, poorly structured, or lacks documentation. It fails to run correctly or reproduce results without significant effort.	No code submitted, or code is completely non-functional.	15%

Annexe C Presentation Rubric

Table 2: Presentation Rubric

Aspect to Evaluate	**Level 4 Outstanding**	**Level 3 Commendable**	**Level 2 Satisfactory**	**Level 1 Insufficient**	**Level 0 No Evidence**	**Weight**
1 **Organisation and Structure**	Exceptionally clear and logical flow. Introduction perfectly sets the stage (context, objectives), and conclusion provides a strong, insightful summary. Transitions are seamless.	Clear and logical flow. Introduction effectively presents context and objectives. Conclusion summarises key points well. Transitions are smooth.	Generally logical flow, but some parts may be slightly disorganised. Introduction and conclusion cover basic points but could be clearer or more impactful. Transitions are adequate.	Disorganised flow, difficult to follow. Introduction or conclusion are weak, missing key elements, or unclear. Transitions are abrupt or confusing.	No discernible organisation or structure.	15%
2 **Content Clarity and Depth**	Explanations of problem, methods, results, and significance are exceptionally clear, accurate, and demonstrate deep understanding. Insightful connections made.	Explanations are clear, accurate, and demonstrate a good understanding of the project concepts and details.	Explanations are generally understandable but may lack some clarity or depth. Basic understanding demonstrated.	Explanations are unclear, inaccurate, or superficial. Demonstrates poor understanding of the project content.	Content is irrelevant or incompre-hensible.	25%
3 **Visual Aids and Delivery**	Visual aids are highly professional, informative, and perfectly support the talk. Delivery is engaging, clear, well-paced, and demonstrates confidence. Excellent use of language.	Visual aids are clear, relevant, and effectively used. Delivery is clear, well-paced, and confident. Good use of language.	Visual aids are adequate but may have minor issues (e.g., clutter, relevance). Delivery is generally clear but may lack engagement or have minor flaws (pace, clarity).	Visual aids are poor quality, confusing, or used ineffectively. Delivery is unclear, hesitant, poorly paced, or lacks engagement.	No visual aids used, or delivery prevents un-derstanding.	20%
4 **Teamwork and Time Management**	Excellent coordination, seamless transitions, and clearly balanced participation. Presentation perfectly fits the allocated time.	Good coordination, smooth transitions, and balanced participation among group members. Presentation adheres well to the time limit.	Adequate coordination and participation, though some imbalance or minor issues with transitions may exist. Generally adheres to time limit, may be slightly over/under.	Poor coordination, awkward transitions, or significantly unbalanced participation. Poor adherence to time limits (significantly over/under).	No evidence of teamwork, or time manage-ment hinders pre-sentation.	10%
5 **Answering Questions**	Demonstrates mastery by providing exceptionally clear, accurate, concise, and insightful answers to all questions. Handles challenging questions effectively.	Provides clear, accurate, and thoughtful answers to questions, demonstrating a solid understanding of the project. Handles most questions well.	Provides generally adequate answers, but some may lack clarity, depth, or accuracy. Demonstrates basic understanding but struggles with complex questions.	Unable to answer most questions accurately or clearly. Responses are confused, incorrect, or demonstrate a lack of understanding.	Does not attempt to answer questions or responses are irrelevant.	30%