AI for Medicine – Project Report

Prof. Stefano Diciotti – University of Bologna

Academic Year: 2024/2025

Student Name: Alessandro Frullo (m. 1132704)

Degree Program: Electronics for Intelligent Systems, IoT & Big Data

Submission Date: 24/07/2025

Summary

1. Project Title

EEG-based Motor Imagery Classification for Brain-Computer Interface (BCI) Applications

2. Problem Statement

This project addresses the classification of EEG signals related to motor imagery versus rest conditions, a crucial problem in Brain-Computer Interface (BCI) research. Accurate detection of motor imagery states from EEG can enable assistive technologies for disabled patients and enhance neurorehabilitation techniques. The challenge lies in handling noisy EEG signals, variability across subjects, and class imbalance, while building robust machine learning models to decode brain states.

3. Objective of the Study

The study aims to develop and evaluate a machine learning pipeline to classify EEG epochs as either motor imagery or rest. The goal is to extract meaningful features from multichannel EEG data, avoiding data leakage, and use interpretable models to assess feature importance, with the goal of improving BCI performance.

4. Dataset Description

Name and Source

- Name: *EEG Motor Movement/Imagery Dataset v1.0.0*
- Source: PhysioNet

Number of Samples

- The dataset contains over 1500 EEG recordings from 109 volunteers; each recording lasts about one to two minutes. For computational power purposes, only data from 50 volunteers has been used.

Data Type

- Format: EDF+ files with 64 EEG channels recorded at 160 Hz sampling rate.
- Channels: EEG signals are recorded using the 10-10 international system (excluding some auxiliary channels).
- Annotations: Each recording has event markers indicating the start of specific tasks.

Available Labels

- *T0*: Rest periods.
- *T1*: Start of real or imagined movement of the left hand or both hands.
- *T2*: Start of real or imagined movement of the right hand or feet.

Challenges

- Class imbalance: There are fewer T1 and T2 events compared to T0, which can affect model training.
- Noise: EEG signals can be noisy and contain artifacts.
- Inter-subject variability: EEG patterns differ among individuals, making generalization difficult.

5. Data Preprocessing

Data preprocessing is the first step, and all function used for these purposes are included in *utils/step1_tools.py*

- EDF files were loaded recursively from a dataset folder using the function *load_edf_dataset()*, extracting signal arrays, channels, and event annotations.
- EEG channels were mapped to brain areas following the 10-10 system using *list_eeg_channels()*.

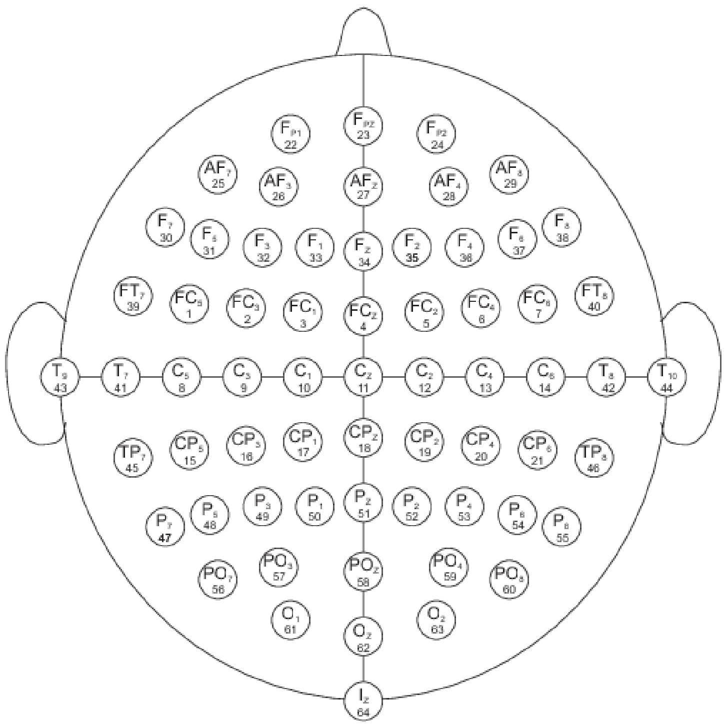

Figure 1: Electrodes position for the international 10-10 system

- Epoch extraction around motor imagery/rest events was performed using the function *extract_epochs_imagery_vs_rest()*, selecting default channels, and extracting epochs from -0.5s to +2.0s around event onset at 160 Hz sampling.
- Epochs with incomplete signal lengths or missing channels were discarded.
- Extracted epochs were concatenated channel-wise and saved as compressed *.npz* and *.pkl* datasets for reproducibility.
- DataFrames were created with *mean signal values* per channel for further feature representation.

Table 1: Portion of table with mean signal values for each electrode (subject 0)

FC5	FC3	FC1	FCZ	FC2	FC4	FC6	C5	C3
-9.38	-234.125	-20.32	-25.91	-20.275	-6.935	-156.675	-315.725	-22.505
74.975	9.695	-5.135	7.625	-5.175	-9.225	-4.05	8.005	-1.09
-107.025	0.53	13.635	125.525	140.425	71.325	20.86	-10.08	0.945
36.345	22.185	13.53	111.725	70.875	10.4	90.575	358.175	224.675
-32.64	-93.375	-8.485	-115.375	-20.325	-47.025	-22.345	-43.65	-3.505

6. Avoiding Data Leakage

design or tuning

To make sure the model learns correctly and does not get "cheating" information, we took special care to avoid data leakage, which happens when information from the test data accidentally influences the training process (it can lead to overly optimistic results that don't work well in real life).

The approach was:

- Training and test data were carefully separated so that no data from the test set was used during training or model tuning.
- Training and test sets were split respecting subject identifiers to avoid mixing data from the same subject across sets, preserving true generalization.
- All feature engineering steps, such as normalization or feature selection, were performed within the training folds only, never using information from the test set.
- When splitting the data, we made sure to keep the distribution of subjects balanced, so the model learns patterns that generalize well and is tested on data it has never seen before.
- The evaluation pipeline strictly avoided "peeking" at test labels during model tuning and training.

7. Machine Learning Pipeline

Here's a step-by-step overview of the machine learning pipeline implemented in the project:

- **Model choice:** We used a **Random Forest classifier**, which is a popular and effective algorithm. It builds many decision trees and combines their results to improve accuracy and reduce overfitting.

- **Data splitting and validation:** To evaluate how well our model works, we split the data into training and testing sets carefully. We used **stratified splitting by subjects**, meaning that data from the same person did not appear in both training and test sets. This way, the model is tested on data from people it has never seen before, which better reflects real-world performance.

- **Feature extraction:** From the EEG epochs, we calculated **mean signal values per channel**, which served as input features to the model. These features summarize the brain activity during each event window.

- **Hyperparameter tuning:** We adjusted key parameters of the Random Forest, like the number of trees and depth, to find the best performing settings using only the training data.

- **Evaluation metrics:** We measured the model's performance using several metrics:
    - **Accuracy:** The overall percentage of correct predictions.
    - **Precision-Recall curve and Average Precision (AP):** Useful for understanding the trade-off between correctly identifying motor imagery and avoiding false alarms.
    - **ROC curve and Area Under the Curve (AUC):** Show the model's ability to discriminate between classes across different thresholds.
    - **Confusion matrix:** Shows the number of true positives, false positives, true negatives, and false negatives.

- **Matthews Correlation Coefficient (MCC):** A balanced metric that accounts for true and false positives and negatives, especially useful for imbalanced classes.
- **Balanced Accuracy:** Average of recall obtained on each class, mitigating the effects of class imbalance.

- **Model interpretability:** To understand which features and brain areas were most important for the classification, we used **SHAP** (SHapley Additive exPlanations). This technique assigns importance values to each feature, helping us see which EEG channels contribute most to the model's decisions. Data extracted from this step, showing extra informations about the most relevant features, are saved into *top_features_shap* subfolder in a *.json* format, for any further analysis and comparison.

- **Visualization and logging:** We used custom plotting functions (*plot_class_distribution()*, *plot_and_save_confusion_matrix()*, *plot_precision_recall_curve()*, *plot_and_save_roc_curve()*) and logging utilities to generate graphs and save performance summaries. These were implemented in separate helper scripts (*step2_tools.py* to *step4_tools.py*) for modularity and clarity. All log files including metrics are in the *log* subfolder, there is one for every run of the pipeline with different number of analyzed subjects.

This pipeline combined data preprocessing, model training, evaluation, and interpretation to build a robust and explainable EEG motor imagery classifier.

8. Results

The following metrics are related to the analysis obtained analyzing 50 subjects, using the Random Forest Classifier.

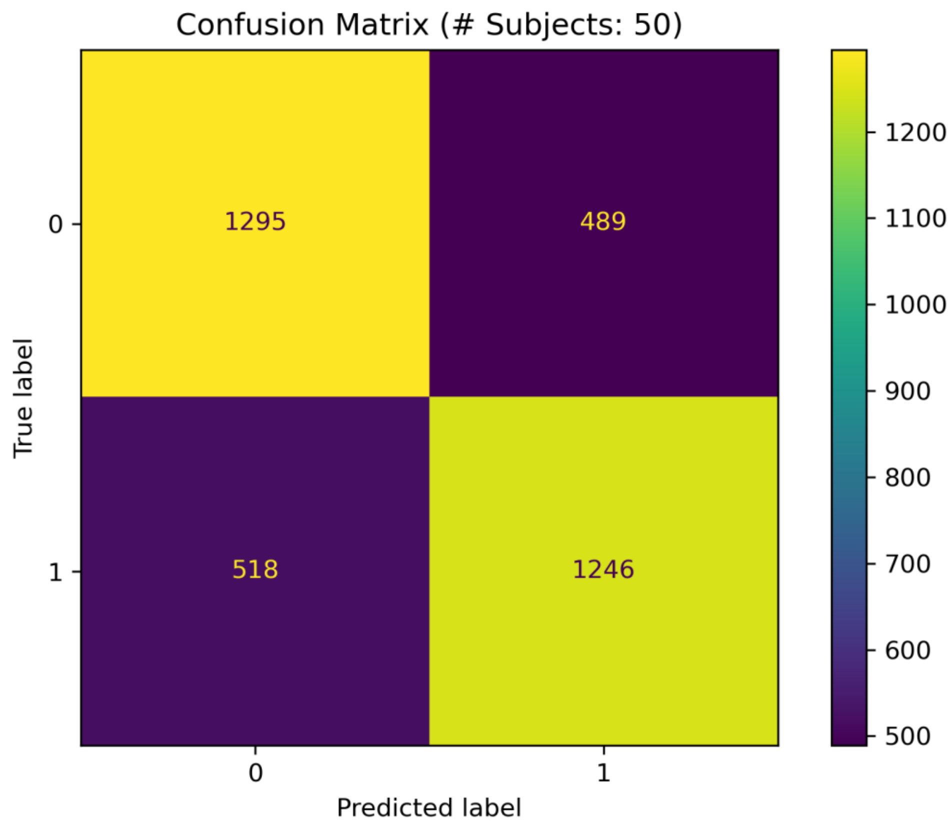

Figure 1: Confusion Matrix

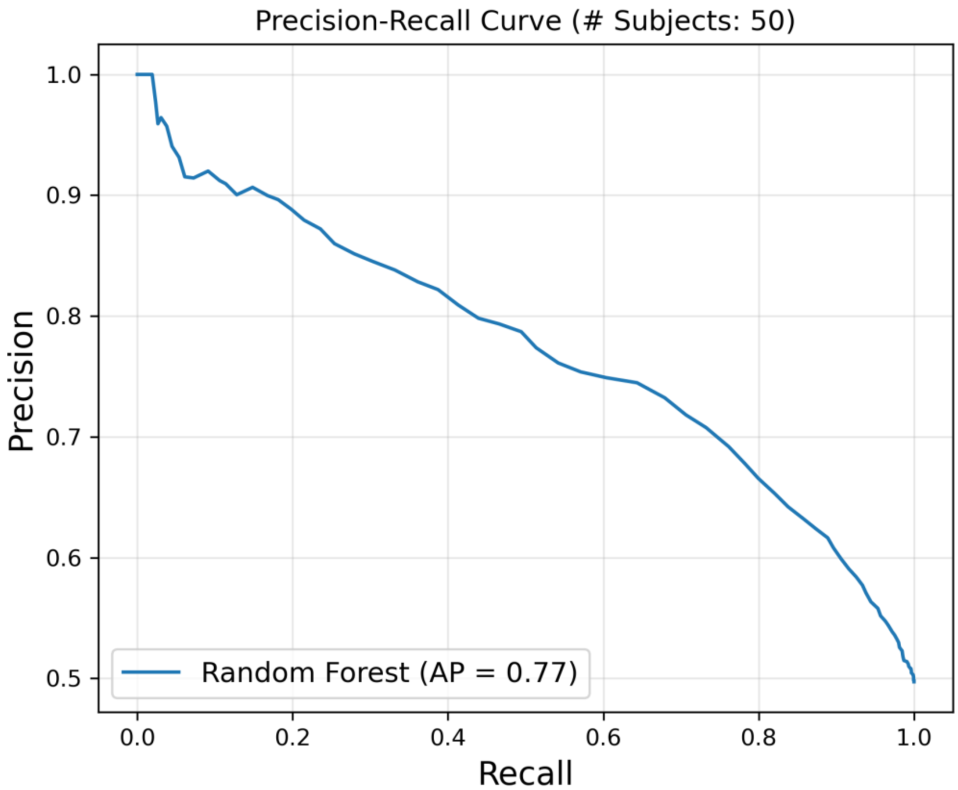

Figure 2: Precision-Recall Curve

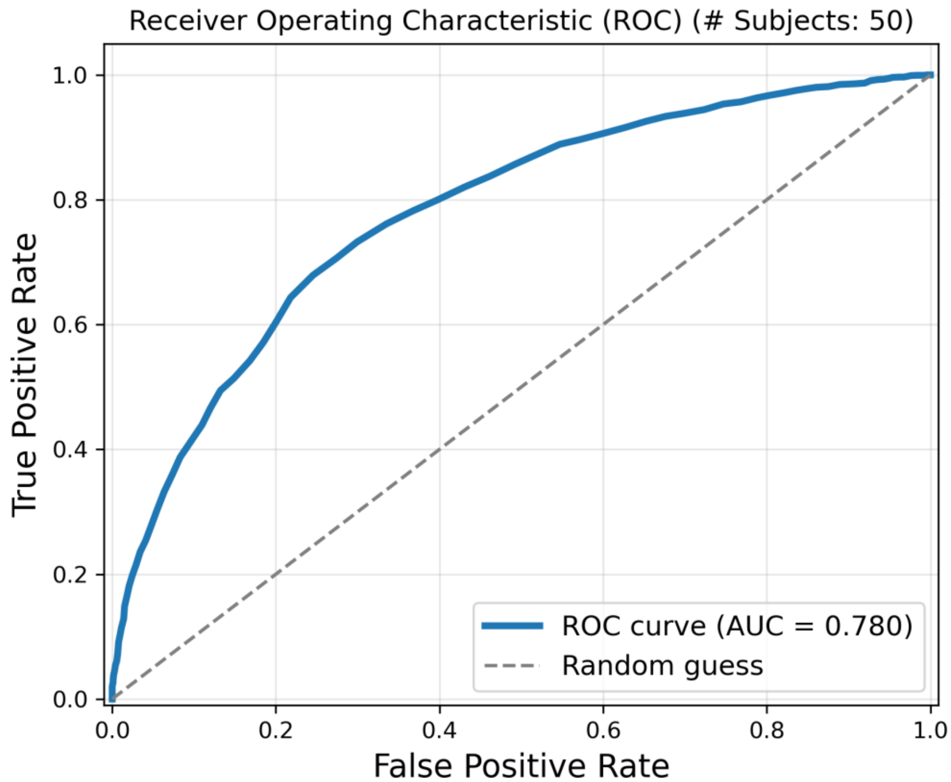

Figure 3: ROC Curve

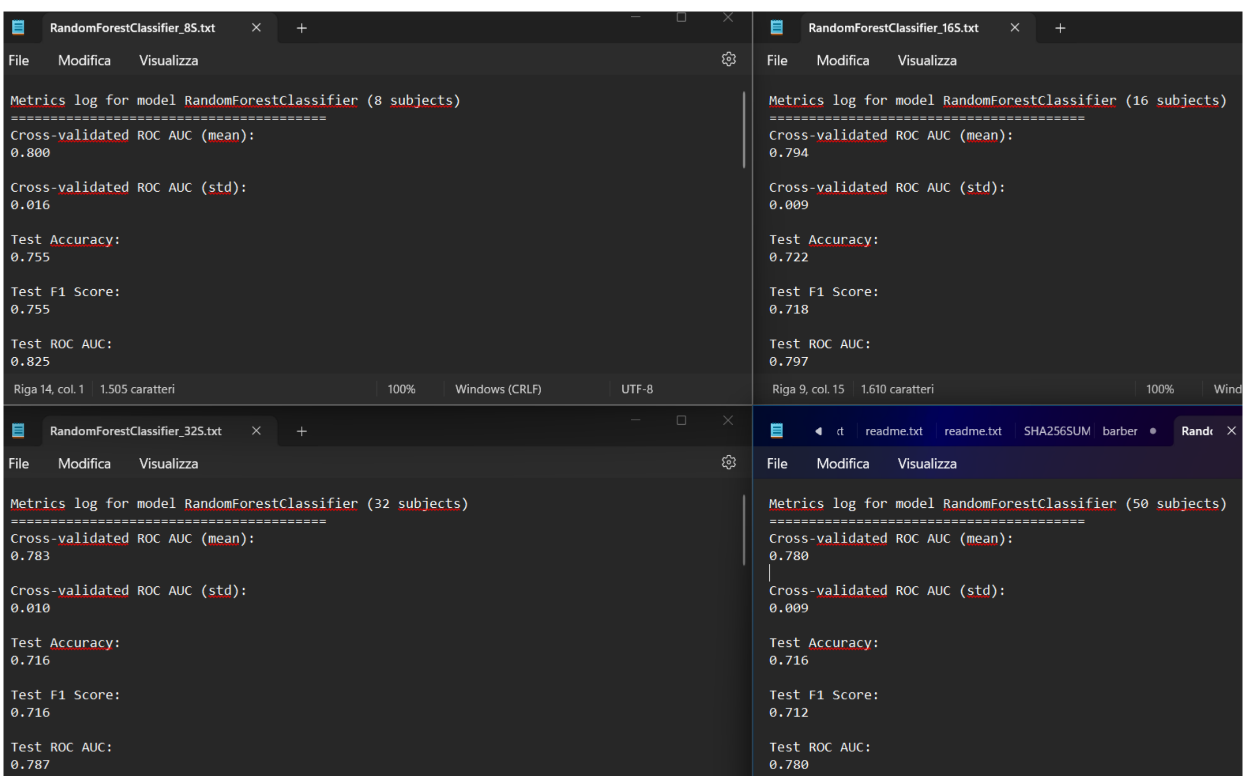

Figure 4: Performance scores with different numbers of analyzed subjects

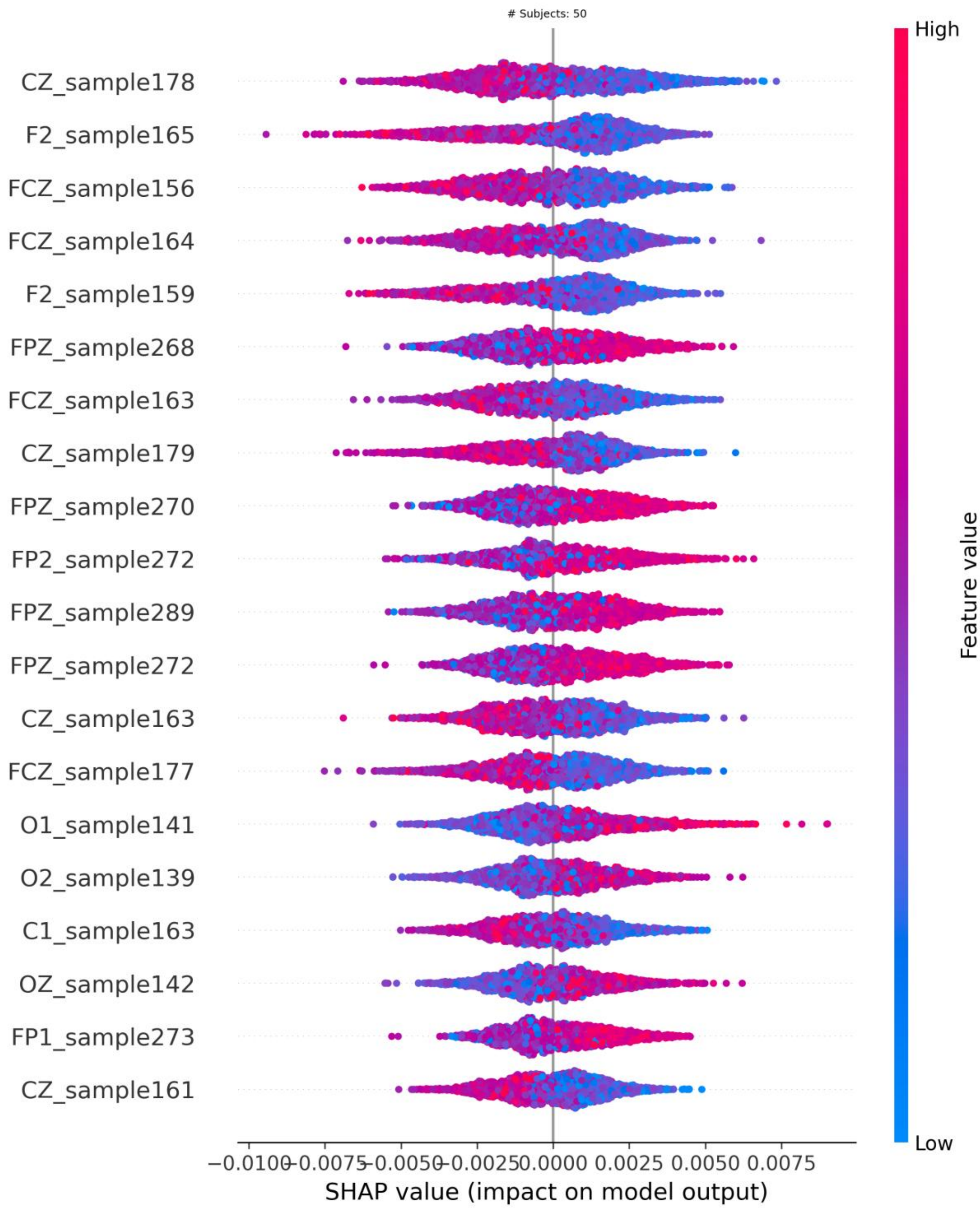

Figure 5: SHAP Analysis

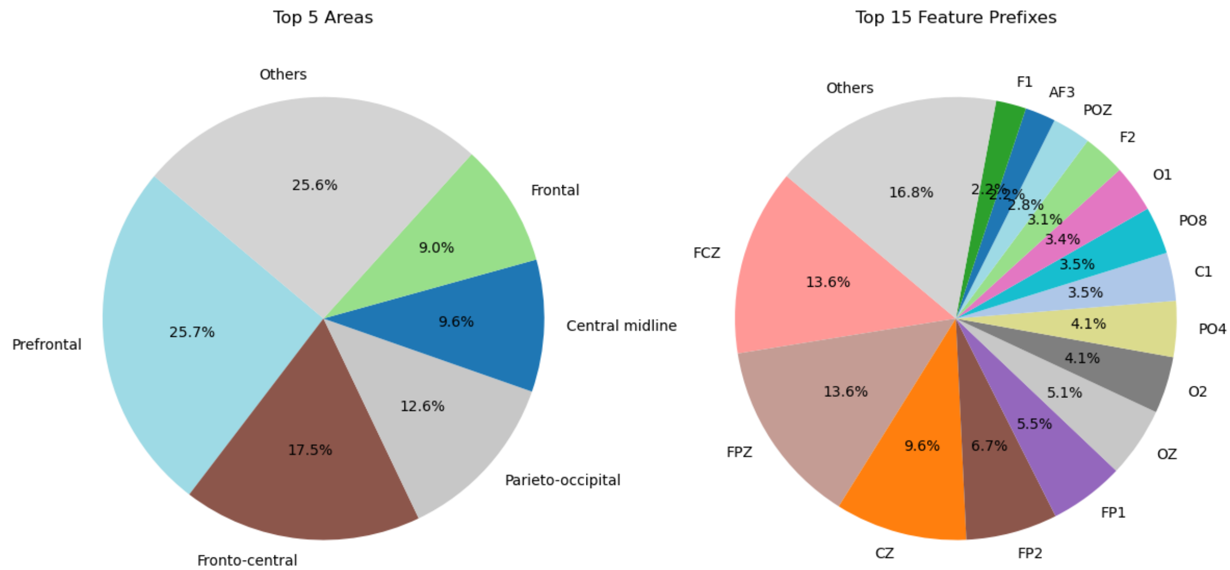

Figure 6: SHAP-based global interpretability

9. Discussion

The *Random Forest classifier* achieved moderate discrimination between motor imagery and rest states across the 50-subject pool:

- The **confusion matrix** (Figure 1) shows that rest epochs were correctly identified in about 73% of cases (1295/1784), while imagery epochs were correctly identified in about 71% (1246/1764), for an overall accuracy of ≈72%. Most errors arose when low-amplitude imagery trials were labeled as rest or vice versa.
- The **precision–recall curve** (Figure 2) gave an average precision (AP) of 0.77. As shown in the PR plot, precision remains above 0.90 for recall up to 0.2, and stays above 0.80 until recall reaches 0.4, demonstrating the model maintains high confidence on a substantial portion of imagery trials even as recall increases.
- The **ROC curve** (Figure 3) yielded an AUC of 0.78, indicating fair separability between classes.
- In Figure 4, there's a **comparison between different runs**, more specifically it was executed the same pipeline with 8, 16, 32, and 50 subjects. The cross-validated AUC was 0.80 with 8 subjects, then 0.79 (16), 0.78 (32), and 0.78 (50). Classification performance stabilizes quickly and adding more subjects introduces extra heterogeneity that pulls the average AUC down to a plateau around 0.78.
- The **SHAP summary** violins (Figure 5) reinforce this: for top samples like *CZ_sample178* and *F2_sample165*, high feature values (red) push the model toward imagery, while low values (blue) push it toward rest. The width of each violin indicates that these time-channel points vary substantially across subjects, yet consistently align with known *μ/β dynamics* in motor imagery (power decreases during imagined movement and rebounds afterward).
- **SHAP-based global interpretability** (Figure 6, left) shows that *prefrontal* (25.7%) and *fronto-central* (17.5%) areas are the ones most relevant to capture attentional and working-memory engagement. At the **single electrode level** (Figure 6, right), *FCZ* and *FPZ* lead (13.6% each), reflecting

a blend of cognitive and motor-planning signals; *CZ* follows (9.6%), consistent with primary motor cortex involvement; then *FP2* (6.7%) and *FP1* (5.5%), which likely index lateralized attentional shifts.

Some limitations, that can lead to further future developments, are:

- It was considered just the average power in each band because it's fast to compute. This means we might have missed quick bursts or complex patterns that richer features catch.
- Not all 109 subjects were analyzed due to computational power limitations (necessity of restructuring the EDF loading format).
- Available data could be used also for testing left vs. right imagery, leading to a multiclass analysis.

10. Conclusions and Future Work

Possible **future improvements** to the project could involve the following aspects:

- Extension to multiclass classification: distinguishing not only between motor imagery and rest, but also between left/right side movements (labels available in the dataset).
- More accurate analysis by considering the available data from all subjects.
- Optimized analysis by focusing only on the most relevant features identified during the current project.
- Comparison of results obtained using different models.

11. Ethics and Data Privacy

The *EEG Motor Movement/Imagery Dataset* used in this project is publicly available from the *PhysioNet repository* (https://physionet.org/content/eegmmidb/1.0.0/). This dataset was collected with appropriate ethical approvals and informed consent from the participants, ensuring compliance with standards for human subject research.

Key points regarding ethics and privacy:

- The dataset is openly accessible for academic and research purposes, with no personal identifying information included.
- Subject anonymity is preserved, as the data is anonymized and only contains EEG signal recordings and event markers.
- The project uses this dataset strictly for non-commercial research aligned with the original data usage terms provided by PhysioNet.
- Any third-party software libraries used in the project are open source or properly licensed for academic use.

12. Code and Reproducibility

The project has been developed using *Visual Studio Code* (v. 1.102.0.0). Runs also in *Google Colab* (necessity to import working folder).

- **Python notebook**: AIMed_Project_Frullo.ipynb

- **Key dependencies**:

python	*3.11.9*
numpy	*2.02*
pandas	*2.2.3*
matplotlib	*3.10.0*
scikit-learn	*1.6.1*
shap	*0.48.0*
pyedflib	*0.1.40*
tqdm	*4.67.1*

- **Utilities encapsulated in modular files**: *step1_tools.py, step2_tools.py, step3_tools.py, step4_tools.py*, all located into *utils* subfolder.
- Results reproducible by running notebook sequentially with dataset in expected folder structure.
- Plots and logs saved automatically respectively into *output_img* and *log* output directories.

13. References

- **PhysioNet EEG Motor Movement/Imagery Dataset**
  Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., & Wolpaw, J. R. (2004). *BCI2000: A General-Purpose Brain-Computer Interface (BCI) System*. *Clinical Neurophysiology*, 115(3), 755–763.
  Available at: https://physionet.org/content/eegmmidb/1.0.0/
- **PhysioNet Standard Citation**
  Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). *PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals*. *Circulation* [Online], 101(23), e215–e220. RRID:SCR_007345.
- **BCI2000 Official Project Page**
  Wadsworth Center, New York State Department of Health.
  Available at: https://www.bci2000.org/mediawiki/index.php/Main_Page
- **ChatGPT**
  OpenAI. Large Language Model used for project support, explanation, and code suggestions.
  Available at: https://openai.com/chatgpt