Mid-Eval Report

EEG P300 Speller

Sarthak Gupta

240939

1. Project Goal

The goal of this project is to develop a real-time P300-based EEG speller system capable of detecting user-intended characters from brain signals. The project focuses on extending the traditional P300 speller approach by incorporating customizable machine learning models for event-related potential classification. Both classical machine learning techniques, such as LDA and SVM, and deep learning models like EEGNet are explored and evaluated for P300 detection. The objective is to build a flexible and modular classification pipeline that supports real-time inference while enabling comparative analysis of model performance in terms of accuracy, latency, and robustness.

2. Progress Made

In the project we have been steadily moving forward towards our goal and have done many assignments and lectures. Below is assignment wise breakdown of progress.

Assignment 0: Basic Python and MATLAB Excercises

In this assignment, we learned basic python and MATLAB syntax and techniques, how to access dataset elements, how to plot results, and so on. Ideas learned here have been used extensively in all the other assignments.

Assignment 1: Supervised Learning Fundamentals

This assignment focused on supervised learning and the ideas behind it. Key topics include overfitting and generalization, bias-variance tradeoff and sources of prediction error. Ensemble learning methods such as bagging,boosting and random forests were studied. We learned how combining multiple models can reduce variance and improve robustness. It also covered risk minimization, loss functions such as hinge loss, squared loss, and regularization techniques such as LASSO and Ridge regression.

Distance-based learning using KNN was discussed, including the effect of the choice of distance metric and the value of k. Decision tree learning was covered along with impurity measures such as Gini index, optimal predictions at leaf nodes, and methods to prevent overfitting.

Assignment 2: EEG Data Processing Pipeline for P300 Detection

In this assignment we finally began working on EEG related concepts. This assignment focused on developing a complete EEG preprocessing and ERP analysis pipeline for P300 detection using real EEG dataset. The aim was to clean raw EEG recordings and align data suitably for analyzing P300 responses.

Raw EEG data was loaded and reshaped from epoch based format to a continous signal with multiple channels. Marker channels corresponding to flashing events and stimulus codes were processed and aligned with the EEG signal to ensure accurate correlation between stimuli and neural responses.

Independent Component Analysis (ICA) was done (figure 1) to visualise and remove artifact related components such as eye blinks and noise. After artifact removal the cleaned EEG Data (figure 4 for visualisation) was segmented into epochs. Averaged evoked responses were computed separately for target and non-target stimuli, and these responses were visualised at electrodes Cz and Pz (figures 3 and 4).

The presence of a P300 signal (although slightly delayed in in my implementation of assignment) in target trials confirmed that the preprocessing and event extraction pipeline was functioning correctly and producing meaningful ERP features.

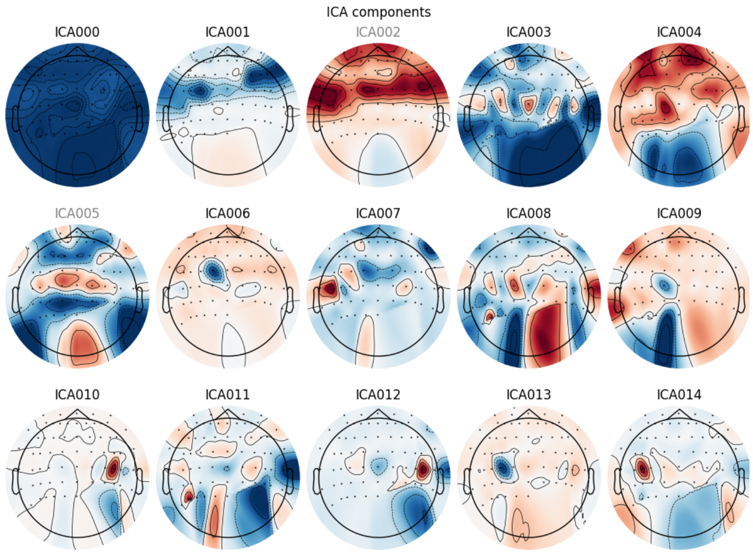

Figure 1: Independent Component Analysis (ICA) components used for artifact identification

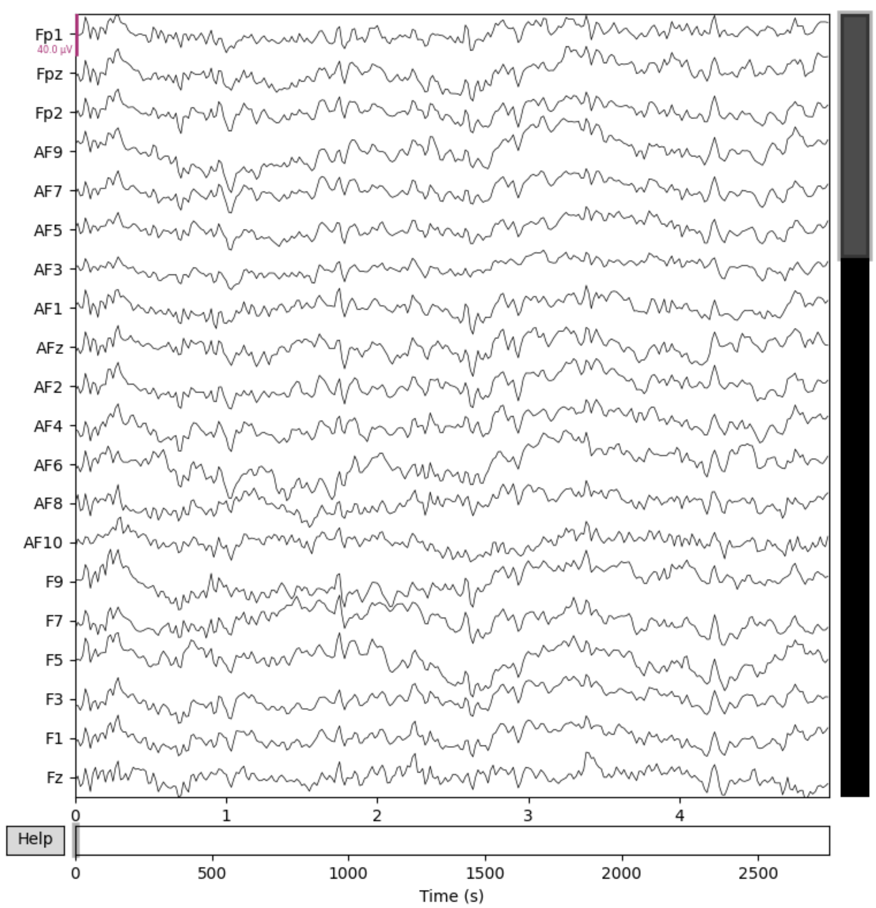

Figure 2: Cleaned EEG data after artifact removal using ICA

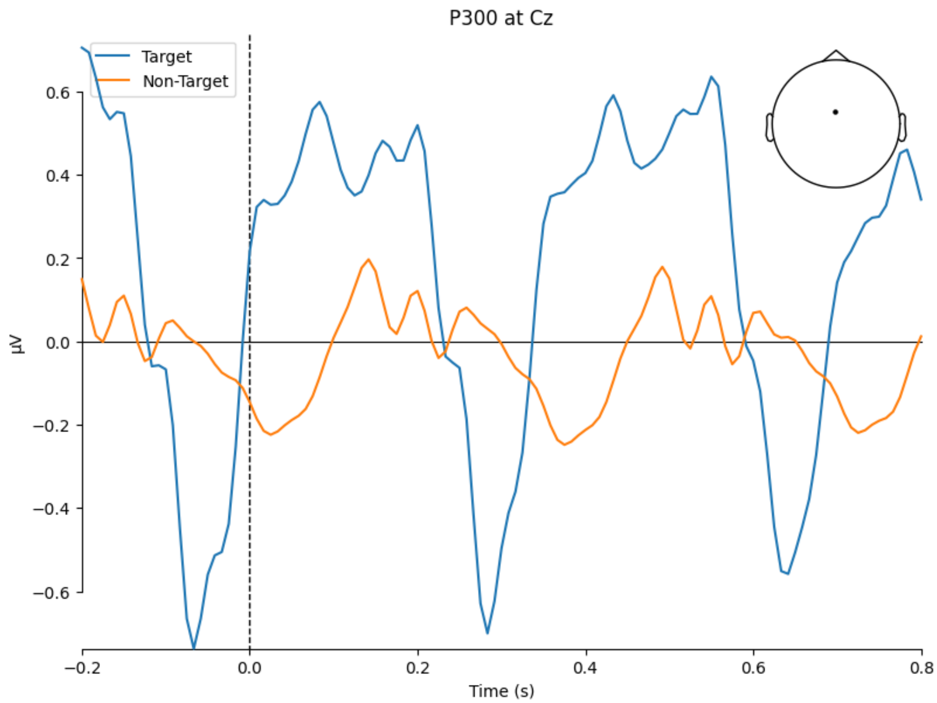

Figure 3: Averaged evoked responses for target and non-target trials at Cz

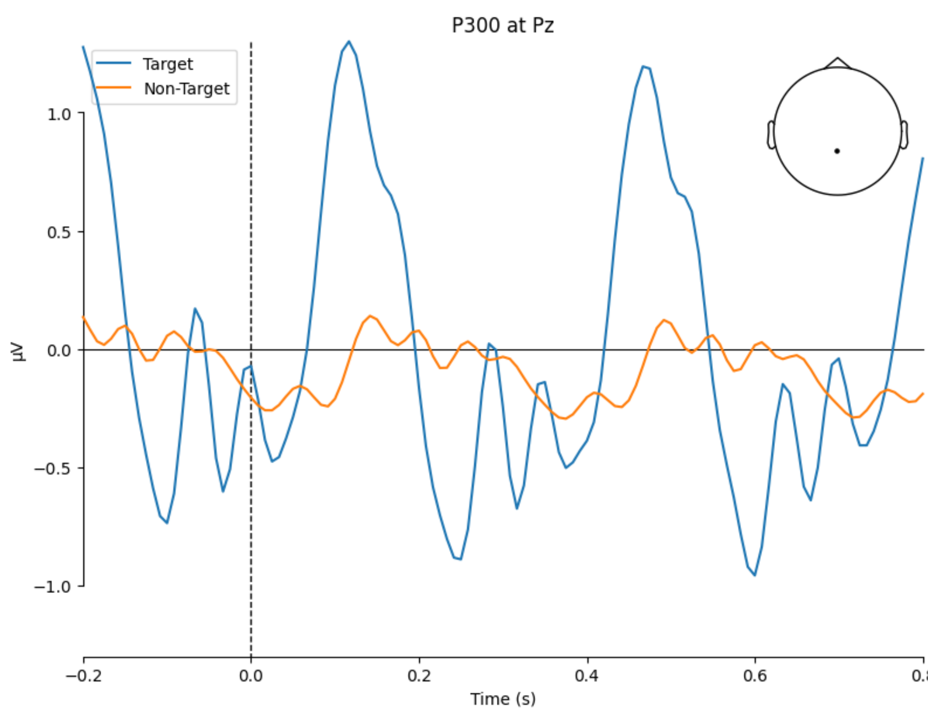

Figure 4: Averaged evoked responses for target and non-target trials at Pz

Assignment 3: Machine Learning Pipeline for P300 Classification

This assignments main focus was on building and then evaluating a machine learning pipeline for classifying P300 EEG responses. Using the preprocessed and epoch-aligned EEG data from earlier stages, the goal here was to extract meaningful features and compare different classifiers for distinguishing target and non-target stimuli.

We explored multiple feature extraction methods including raw domain features, principal component analysis (PCA) with different numbers of components, and Common Spatial Patterns (CSP). These representations were evaluated using balanced baseline classifiers such as Linear Discriminant Analysis (LCA) and logistic regression. The best performing feature set was selected based on validation performance.

Afterwards several classical ML models including Support Vector Machines (SVM), Random forests, and Gradient Boosting were trained and compared with each other. Class imbalance was handled using training strategies. Performance was evaluated using F1-score. The selected feature method was applied to the subjects.

Finally, trained models along with their associated preprocessing componenets were saved and exported for later use.

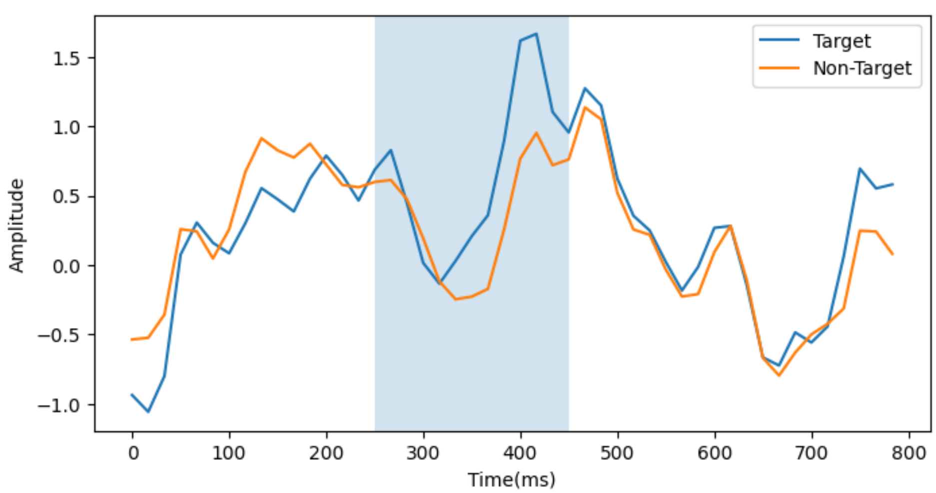

Figure 5: Visualising ERP responses

```
LDA Results
              precision    recall  f1-score   support

         0.0       0.81      0.58      0.68       657
         1.0       0.14      0.34      0.19       131

    accuracy                           0.54       788
   macro avg       0.48      0.46      0.44       788
weighted avg       0.70      0.54      0.60       788

Confusion Matrix:
[[380 277]
 [ 87  44]]

Logistic Regression Results
              precision    recall  f1-score   support

         0.0       0.83      0.76      0.80       657
         1.0       0.16      0.23      0.19       131

    accuracy                           0.68       788
   macro avg       0.50      0.50      0.49       788
weighted avg       0.72      0.68      0.70       788

Confusion Matrix:
[[502 155]
 [101  30]]
```

Figure 6: Model results

3. Code Links

- Assignment 0 Notebook: `https://github.com/Sartha-k-g/EEG-Speller-Assignm` `blob/main/240939_Sarthak_Gupta.ipynb`
- Assignment 1 Notebook: `https://github.com/Sartha-k-g/EEG-Speller-Assignm` `blob/main/240939_SarthakGupta.pdf`
- Assignment 2 Notebook: `https://github.com/Sartha-k-g/EEG-Speller-Assignm` `blob/main/240939_SarthakGupta_Assignment_2.ipynb`
- Assignment 3 Notebook: `https://github.com/Sartha-k-g/EEG-Speller-Assignm` `blob/main/EEG_assignment3_SarthakGupta_240939.ipynb`

4. What Did and Didn't Work

What Worked:

- EEG preprocessing steps such as filtering, baseline correction, and epoch extraction produced stable and stimulus-aligned data.
- Averaged ERP responses at electrodes Cz and Pz showed a clear separation between target and non-target trials, confirming the presence of a P300-related component.
- Feature extraction using PCA and CSP improved class separability compared to raw time-domain features.
- SVM and ensemble-based classifiers performed better than simple linear models when handling EEG data.

What Did Not Work Well:

- The P300 peak was observed at a slightly delayed latency, making precise temporal localization difficult.
- Classification using raw EEG features was sensitive to noise and high dimensionality.
- Class imbalance continued to limit target-class recall despite reasonable overall accuracy.
- Model performance was sensitive to preprocessing and feature selection choices.

5. Challenges Faced

One main challenge I encountered was that the peaks in the Cz and Pz plot (figure 3 and 4) isn't at 300ms exactly, as one would expect from P300 type response, but slightly delayed. Although a clear distinction was observed between target and non target responses, the delay made it less straightforward to define a window for feature extraction. After researching I think this delay is likely due to preprocessing choices such as downsampling and epoch alignment.