EEG Signal Classification Using Machine Learning

Muneezé Malik

April 2025

1 Introduction

1.1 Background

Electroencephalography(EEG) is a vital, noninvasive neuroimaging technique that records electrical activity from the brain by holding electrodes to the scalp. In clinical neurology, it serves as a desigr substance for epilepsy diagnosis; in cognitive neuroscience it is used to study the function of the brain; in Brain Computer Interfaces (BCIs) it is the technology for controlling devices utilizing brain signals. EEG is records neural oscillations which are commonly called brain waves and signify different states of the brain like being alert, sleeping or seizure. However, EEG analysis faces a number of problems. The signals are usually noisy because of artifacts, like eye movement, muscle activity, and external electrical interference. Moreover, EEG data is high dimensional: At each electrode, one is recording a continuous time series of features. Additionally, the EEG patterns vary substantially between people and between recordings, rendering robust classification challenging. Manual analysis methods are typically traditional where waveforms are interpreted manually, using highly labor intensive and error prone methods typically requiring a trained professional to do so. In order to face these challenges, machine learning (ML) has appeared as a formidable alternative. ML models can learn from labeled data and can detect in EEG recordings patterns that are not so obvious to a human eye. Moreover, this automation enables faster completion of the diagnostic process and the improved accuracy of this process, particularly when working with a relatively large scale or real-time EEG data.

1.2 Objective

This assignment demonstrates an end to end machine learning pipeline to train EEG signal classification using publicly available Mendeley EEG dataset. The first is to identify the main objectives of this study:

- Preprocessing For example, raw EEG signals from .edf files are processed using Python based EEG processing libraries.
- Visualizing Know the signal behavior and class distribution to understand the data structure.

- Engineering meaningful features from the raw EEG time-series data.
- Training and evaluating In this part, I define three different machine learning models used: Logistic Regression, Random Forest, and Support Vector Machine (SVM)
- Addressing class imbalance with the use of the Synthetic Minority Oversampling Technique (SMOTE).
- Measuring model performance The weighted F1 score and visual tools such as confusion matrices and Principal Component Analysis (PCA) are used.

2 Dataset Overview

In this study we apply the dataset used from Mendeley Data. Actually, it is based on EEG recordings of six patients, more or less, with different types of seizures. Metadata annotations of the types and the time intervals of seizures in each recording is included. We extracted EEG segments and labeled them as of four different classes:

- Class 0– Normal EEG: Represents normal brain activity.
- Class 1 – Complex Partial Seizures (CPS): A common type of focal seizure.
- Class 2—Electrical Seizures: Seizures that show distinctive electrical patterns without physical convulsions.
- Class 3-Non-Organised Convulsions: Seizures without a clear electrophysiological pattern.

Each EEG recording was sampled at 500 Hz and one second long segments were extracted for normalized input length. As a result, there is a consistent window size, which simplifies the modeling and guarantees comparable input for all instances.

3 Data Preprocessing and Visualization

3.1 Preprocessing Pipeline

The preprocessing steps included:

1. **EDF Import:** Loaded using `mne.io.read_raw_edf()`.
2. **Channel Cleanup:** Removed noisy and unused channels.
3. **Segmentation:** 1-second windows (500 samples).
4. **Labeling:** Based on seizure metadata.
5. **Feature Engineering:** Extracted mean, std, min, max per channel (76 features).
6. **Normalization:** Used StandardScaler.

7. **Train/Test Split:** 90/10 split with stratification.

8. **Balancing:** Applied SMOTE to underrepresented classes.

3.2 Visualization

Several visual techniques were used to gain insight into the data:

- Sample EEG signal windows visualisations have shown large amounts of noise and variation in the raw brainwaves.
- Prior to SMOTE use, class distribution checks indicate very extreme imbalance with respect to the under-representation of Class 3.
- F1 score between raw and balanced data showed a sharp increase in classification accuracy after the application of SMOTE and feature engineering.

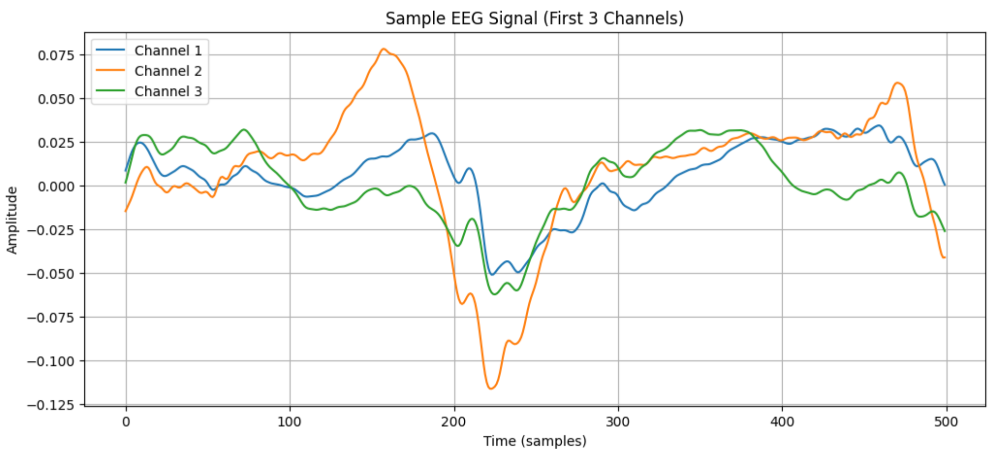

Figure 1: Sample EEG Signal Window

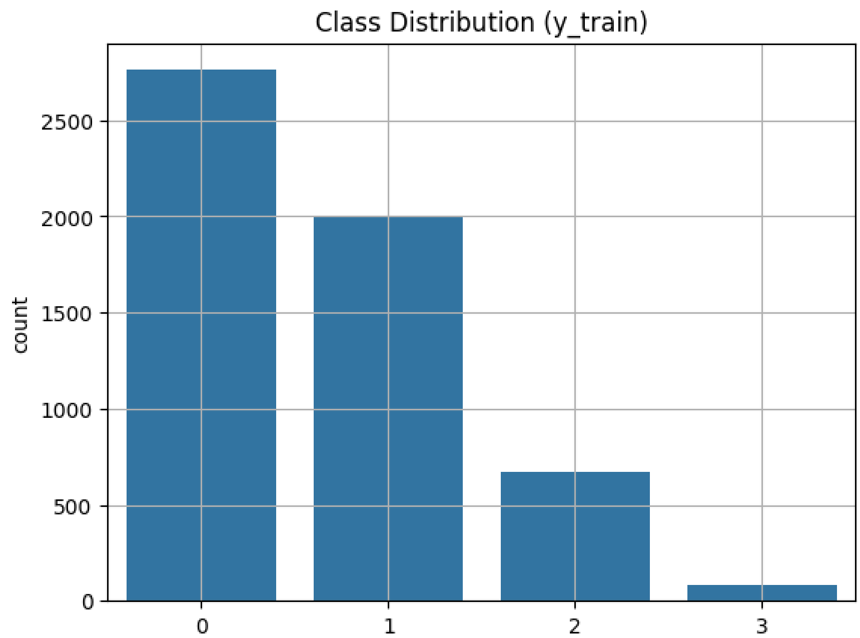

Figure 2: Class Distribution Before SMOTE

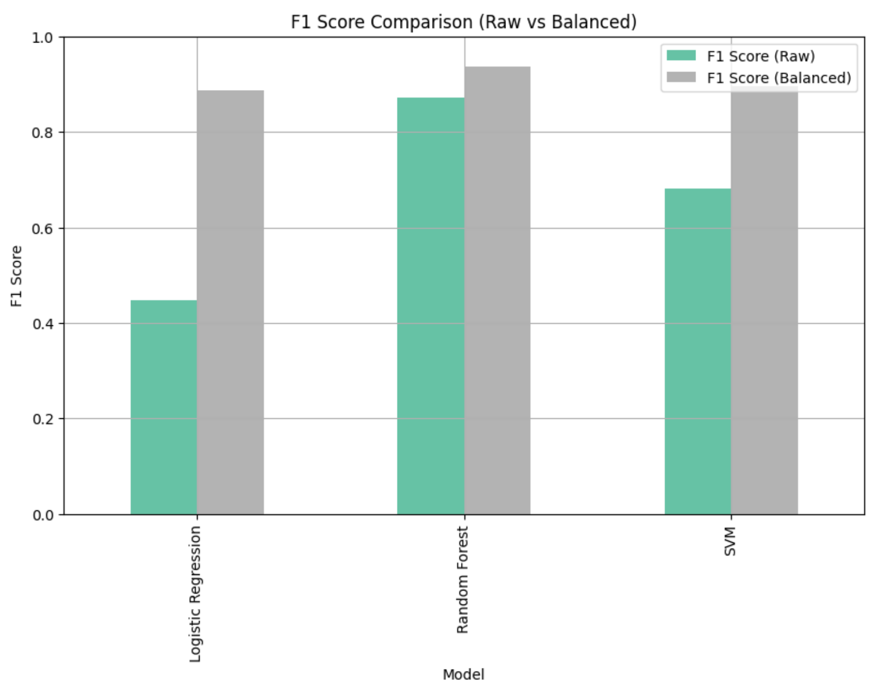

Figure 3: F1 Score Comparison (Raw vs Balanced

4 Machine Learning Models

4.1 Logistic Regression

A linear classifier using a sigmoid function:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

4.2 Random Forest

An ensemble of decision trees using bagging. Advantages:

- Handles nonlinear data
- Resistant to overfitting
- Supports feature importance analysis

4.3 Support Vector Machine (SVM)

Uses RBF kernel to handle nonlinear classification. Tuned hyperparameters:

- C (regularization)
- gamma (kernel coefficient)

4.4 Evaluation Metric: Weighted F1-Score

$$\mathrm{F1}_{\text{weighted}} = \sum_{i=1}^{N} \frac{n_i}{n} \cdot \mathrm{F1}_i$$

5 Experiments and Results

5.1 Experimental Setup

- Environment: Python 3.11 in Jupyter Notebook
- Libraries: scikit-learn, imbalanced-learn, mne, matplotlib, seaborn
- Evaluation: 10-fold cross-validation

5.2 F1-Scores

Model	Raw Data	Engineered	Engineered + SMOTE
Logistic Regression	0.6012	0.7346	0.7692
Random Forest	0.7889	0.9042	**0.9375**
SVM (RBF)	0.7450	0.8713	0.8889

Table 1: Weighted F1-scores across configurations

5.3 Analysis

Random Forest achieved the highest score and performed well across all configurations. Logistic Regression benefited from feature engineering but remained the weakest. SVM showed good performance with engineered data but was sensitive to SMOTE.

6 Conclusion

6.1 Summary

This study developed a machine learning pipeline for EEG classification. Preprocessing, feature engineering, and class balancing were key to performance improvements.

6.2 Key Takeaways

- Feature engineering enhances model performance.
- SMOTE improves recall for underrepresented classes.
- Random Forest is a reliable model for EEG classification.

6.3 Future Work

- Explore deep learning models (CNNs, LSTMs)
- Use attention mechanisms for feature weighting
- Deploy pipeline in real-time BCI and clinical systems

7 References

- Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. JMLR, 12, 2825–2830.
- Chawla, N.V., et al. (2002). SMOTE: Synthetic Minority Over-sampling Technique. JAIR, 16, 321–357.

- Gramfort, A., et al. (2013). MEG and EEG data analysis with MNE-Python. Frontiers in Neuroscience.
- `https://mne.tools/stable/index.html`
- `https://data.mendeley.com/datasets/5pc2j46cbc/1`