EEG data classification

Piotr Pawłowski, Filip Juza

Introduction

EEG (Electroencephalography) is a technique for exploring human brain activity. It is a non-invasive neurophysiological technique used to measure and record the electrical activity of the brain. It involves placing electrodes on the scalp to detect and amplify the electrical signals generated by the firing of neurons (nerve cells) in the brain.

Neurons communicate with each other through electrical impulses, and these electrical activities create patterns that can be detected on the scalp. EEG captures these patterns in the form of brainwaves, which are categorized into different frequency bands such as delta, theta, alpha, beta, and gamma waves. Each of these waves is associated with specific mental states or activities.

By analyzing EEG recordings, researchers and clinicians can gain insights into brain function, study various cognitive processes, and diagnose or monitor neurological conditions such as epilepsy, sleep disorders, and brain injuries. EEG is valuable for studying brain activity in real-time and is widely used in both research and clinical settings.

The primary objective of this work is to develop a machine learning model with the capability to categorize and distinguish human activities based on the analysis of recorded electroencephalography signals. In essence, we aim to harness the power of advanced computational algorithms to discern patterns and features within the EEG data, enabling the model to accurately identify and classify different states or actions exhibited by the human subject.

Dataset

The Amsterdam Open MRI Collection[1] (AOMIC) is a collection of three multimodal MRI data including structural, diffusion weighted and functional BOLD MRI datasets. This work uses only one of them(PIOP2).

Dataset description

Original dataset contains 896 samples. Every element of dataset is associated with activity performed by a subject, activity distribution among our dataset is shown at Figure 2:

[1] https://nilab-uva.github.io/AOMIC.github.io/

- Emotion matching
- Resting state
- Working memory
- Stop signal

Each sample consists of 200 time series of varying length. Each series represents recorded EEG signals of a particular brain area. Some parts of EEG signals are shown at Fiigure 1).

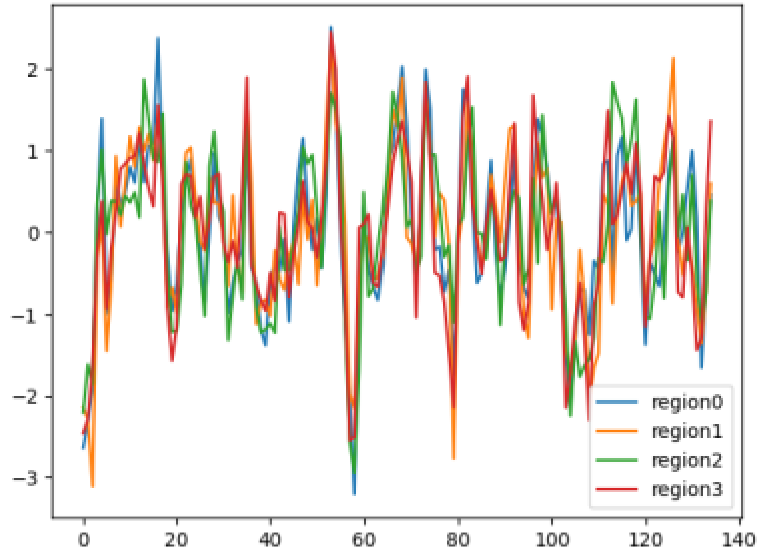

Figure 1) Example signals from one sample for 4 regions after denoising.
No clear trend or pattern is visible in data.

Dataset preparation

We employed two distinct methodologies for the generation of training data: the first entailed feature extraction techniques, while the second applied Principal Component Analysis (PCA) directly to the raw time series data.

We automatically generate input features by using python library called *tsfresh*.[2]
It is a powerful tool designed for automatic extraction of relevant features from time series data. It simplifies the process of feature engineering by automatically computing a wide range of descriptive statistics and characteristics from time series, enabling efficient analysis and modeling of temporal patterns. For one timeseries it is possible to calculate hundreds of features, including most common ones, like standard deviation, mean, skewness, kurtosis, minimum, maximum and others.

For PCA, columns with missing values were removed and data was scaled using StandardScaler[3].

[2] https://tsfresh.readthedocs.io/en/latest/
[3] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

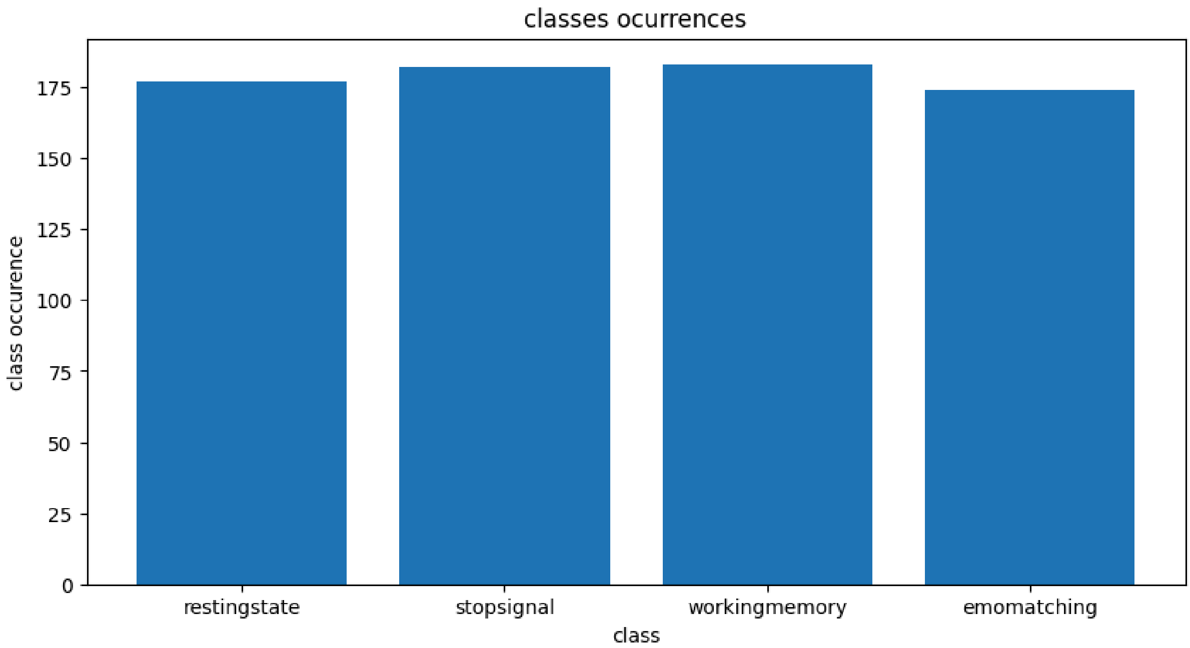

Figure 2) Distribution of classes in dataset. Classes are almost equally numerous.

Methods

In this work we decided to use 2 different approaches/models for classification. The first one is using LightGBM[4], which is an improved version of gradient boosting tree. On numerous occasions, LightGBM has demonstrated[5][6] a remarkable precision in contrast to alternative models, substantiating its efficacy and superiority in predictive accuracy.
The second method acts as a baseline - it uses a Naive Bayes Classifier.

Light GBM

LightGBM is a gradient boosting tree technique, improved with *Gradient-based One-Side Sampling* (GOSS) and *Exclusive Feature Bundling* (EFB). It is used for classification tasks. It is easy to learn, as it does not use deep learning, and compared to other similar solutions achieves better performance. Benchmark for this method is shown on Figure 3).

Table 3: Overall accuracy comparison on test datasets. Use AUC for classification task and NDCG@10 for ranking task. SGB is lgb_baseline with Stochastic Gradient Boosting, and its sampling ratio is the same as LightGBM.

	xgb_exa	xgb_his	lgb_baseline	SGB	**LightGBM**
Allstate	0.6070	0.6089	0.6093	0.6064 $\pm$ 7e-4	**0.6093 $\pm$ 9e-5**
Flight Delay	0.7601	0.7840	0.7847	0.7780 $\pm$ 8e-4	**0.7846 $\pm$ 4e-5**
LETOR	0.4977	0.4982	0.5277	0.5239 $\pm$ 6e-4	**0.5275 $\pm$ 5e-4**
KDD10	0.7796	OOM	0.78735	0.7759 $\pm$ 3e-4	**0.78732 $\pm$ 1e-4**
KDD12	0.7029	OOM	0.7049	0.6989 $\pm$ 8e-4	**0.7051 $\pm$ 5e-5**

Figure 3) Method results comparison[7]

[4] https://dl.acm.org/doi/10.5555/3294996.3295074
[5] https://arxiv.org/abs/2310.04700
[6] https://www.researchgate.net/publication/342683052_LightGBM_Algorithm_for_Malware_Detection
[7] https://dl.acm.org/doi/10.5555/3294996.3295074

Naive Bayes Classifier with PCA

The Naive Bayes Classifier[8], a fundamental tool for classification tasks, operates on Bayes' rule. To mitigate issues related to overfitting and the curse of dimensionality, we incorporate Principal Component Analysis (PCA) on the extracted features. The selection of the ideal number of components is based on insights drawn from the scree plot, contributing to improved robustness and generalization in the classifier's performance.

Results

The conducted experiments involved the utilization of the provided dataset, where 80% of the data was allocated for training purposes, while the remaining 20% constituted the test set for model evaluation. It is noteworthy that both methodologies used are time efficient, allowing for fast model development and validation of obtained results.

Metrics comparison

We assess the performance of the classifiers we utilized by calculating three metrics: f1-score, accuracy, and precision. This helps us gauge how well the classifiers are doing from different perspectives. Collected metrics are presented in Figure 4)

	F1 score[9]				Precision[10]				Accuracy[11]
class	1	2	3	4	1	2	3	4	
NB classifier with PCA	0.57	0.42	0.88	0.85	0.49	0.45	0.94	0.92	0.67
LightGBM	0.7	0.67	0.9	1	0.68	0.67	0.92	1	0.82

Figure 4) Results achieved by two compared methods. We can see that LightGBM outperforms NBC in every metric

Discussion

We can see that LightGBM significantly outperforms Naive Bayes Classifier. This was expected, as NBC is a baseline method. Further improvements may be acquired by performing PCA on LightGBM input data. Other methods based on DeepLearning or other Gradient Boosting methods may yield more accurate results, while not being so efficient in terms of training.

It is worth pointing out that results differ greatly for each of the predicted classes. Such behavior may be the result of feature extraction, and may be corrected by using raw time

[8] https://arxiv.org/abs/1404.0933
[9] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
[10] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html
[11] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html

series as input to the method. It is not desired for methods to have such a big span in results. Further work to reduce such differences is required.