Harmful Brain Activity Classification - Executive Summary

Project on detecting and classifying seizure-like brain activity based on the HMS - Harmful Brain Activity Classification Kaggle project. (Erdos Institute, Spring 2024)

Group: Evgeniya Lagoda, Kshitiz Parihar, Souparna Purohit, and Jianing Yang.

Acknowledgement

We would like to thank our mentor Rongqing Ye for his many helpful suggestions, and always keeping us on track. We would also like to thank The Erdös Institute Data Science Boot Camp, and in particular, Roman Holowinsky, Steven Gubkin, and Alec Clott for giving us this opportunity.

Background

About EEG

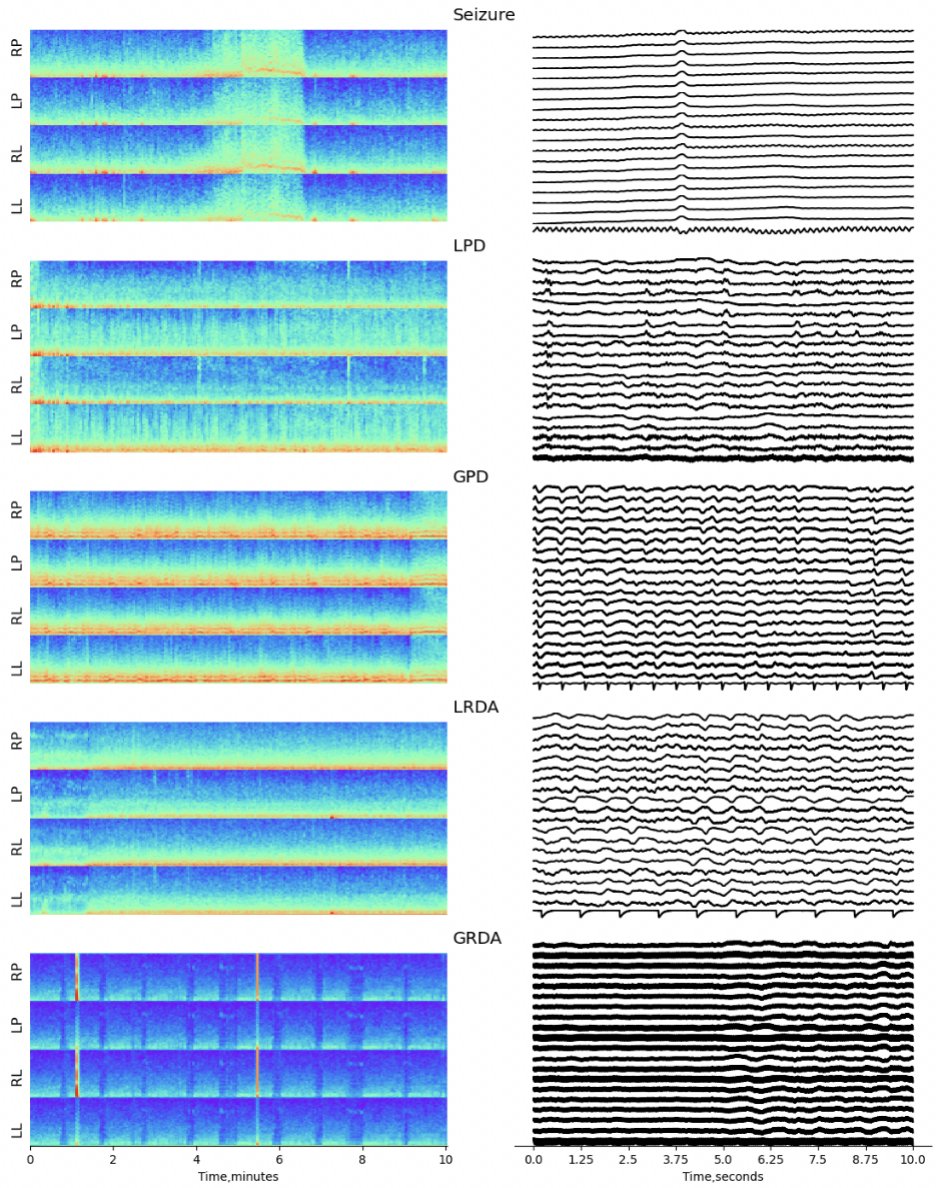
Electroencephalography (EEG) is a method to record the spontaneous electrical activity of the brain. 19 electrodes are placed on the scalp to detect electrical signals from four regions of the brain: LL (left lateral), RL (right lateral), LP (left parasagittal), and RP (right parasagittal).

Goal

The goal of this project is to detect and classify the following types of harmful brain activities:
Seizure
Generalized Periodic Discharges (GPD)
Lateralized Periodic Discharges (LPD)
Lateralized Rhythmic Delta Activity (LRDA)
Generalized Rhythmic Delta Activity (GRDA)
“Other”

We will do this by training models on EEG signals recorded from hospital patients exhibiting such brain activities. More specifically, given 50 seconds of EEG signal, our model will output a probability distribution for the six classes [‘Seizure’, ‘LPD’, ‘GPD’, ‘LRDA’, ‘GRDA’, ‘Other’].

Shown below are examples of EEG data (on the right), along with its corresponding spectrogram data (on the left), for each of the 5 classes of interest.



Stakeholders

Hospitals, labs, and brain researchers: Automating EEG analysis that can alleviate the labor-intensive, time consuming, and fatigue-related error prone manual analysis by specialized personnel, enabling detection of seizures and other types of brain activity that can cause brain damage ensuring quick and accurate treatment.

Data

Starting dataset

The dataset for the project is publicly available at Kaggle Competition HMS – Brain Activity Classification. The data from Kaggle provided includes:

1. train.csv from Kaggle competition, which has 106800 data points (i.e. rows of data).
2. EEG data associated with each 'eeg_id' in train.csv (typically 50 secs of EEG data, but longer on occasion). The EEG data provided from Kaggle consist of columns with readings from the 19 electrodes mentioned above (plus EKG): ['Fp1', 'F3', 'C3', 'P3', 'F7', 'T3', 'T5', 'O1', 'Fz', 'Cz', 'Pz', 'Fp2', 'F4', 'C4', 'P4', 'F8', 'T4', 'T6', 'O2', 'EKG'].

The data consists of readings over 50 seconds (and sometimes longer), sampled at a rate of 200 samples per second. For EEG data longer than 50 seconds, an EEG offset is provided in the train dataset (as the column 'eeg_label_offset_seconds'). An offset of k seconds indicates that the EEG window from seconds k to k+50 is used for the final predictions (we note that the final classification into the seizure-like activity is made from the middle 10 seconds of the 50 seconds window of EEG data).

3. Spectrogram data associated with each 'spectrogram_id', which typically consists of 10 minutes of data.

Data Preprocessing

In **scripts/data_preprocessing/eeg_preprocessing.py**:

1. From the Kaggle provided EEG data, we obtain the "relative signals"
   LL: Fp1 - F7, F7 - T3, T3 - T5, T5 - O1
   LP: Fp1 - F3, F3 - C3, C3 - P3, P3 - O1
   RP: Fp2 - F4, F4 - C4, C4 - P4, P4 - O2
   RL: Fp2 - F8, F8 - T4, T4 - T6, T6 - O2
   for each 50 seconds of EEG data by taking the appropriate differences between the original EEG signals. The LL, LP, RP, RR denote the four different regions of the brain mentioned in the Background section.

2. We then process these relative signals to remove frequencies below 0.5 Hz and above 40 Hz, and also apply a notch filter at 60 Hz. These signal preprocessing steps are based on framework generally applied in EEG studies [2]. We save the middle 10 seconds of data for future use.

In **scripts/data_preprocessing/extract_time_frequency_univariate_features.py**:

3. Using the python library "eeglib", we extract various features from the (middle 10 seconds of the filtered) relative signals that have been previously used in literature for EEG classification tasks [1]. Some such features are: relative band powers, spectral edge frequncy, and the Hjorth parameters. We also calculate standard statistics for the relative signals (mean, standard deviation, skewness, and kurtosis). This yields 448 new features per row of the original dataset.

In **scripts/data_preprocessing/feature_extr_kaggle_spec.ipynb**:

4. From the Kaggle provided spectrograms, we extract features such as total power and powers for various frequency ranges (bands) for each of the four regions of the brain mentioned above, as well as statistics such as mean, min, and max for each column of the provided spectrogram over a 10 minute window, and a 20 second window. The idea here is to capture information about the long term behavior of the EEG signal that is missing from looking at just the 50 second EEG data. This yields 2424 new features per row of the original dataset.

In **scripts/data_preprocessing/make_mel_spec_from_EEG.ipynb**:

5. From the 50 second window of the Kaggle provided EEG data, after applying preprocessing steps 1 and 2, we create our own Mel spectrograms using the librosa python library. Then we extracted statistics such as mean, min, max, and standard deviation for the middle 10 seconds of each frequency in the Mel spectrogram for each of the four regions of the brain. These spectrograms give us much more granular spectral information about the short term EEG signals, compared to the lower

resolution, longer term spectrogram data from step 4 above. This yields 2048 new features per row of the original dataset.

In **scripts/data_preprocessing/merge_spectrogram_features_n_train_test_split.ipynb**:

6. We first filter out rows from the original dataset where preprocessing step 2 yielded NaNs. Then, in order to prevent over-representation of EEG data points with multiple time offsets close to one another and since the features we consider are less sensitive to moderate changes in amplitudes and frequencies of the EEG signals that can happen in short periods of time, we filter out EEG offsets that are less than 10 seconds apart (since the final predictions are made on 10 seconds of EEG data). Furthermore, since the predictions for different offsets are similar (as they are ultimately based on different time windows of the same EEG signal), we retain the votes of the dropped EEG offsets, and merge them with the remaining ones.

This results in dropping the number of data points from 106,800 (in the original training data set) to around 32,500 data points.

7. Then we use (one iteration of) StratifiedGroupKFold to create a train/test split, stratifying by the predicted class of seizure-like activity (i.e. “expert_consensus”), and grouping by the patient_ids to ensure there is no overlap of patient ids in the train and test sets.

There are several data points that have only one or two experts whose votes determined the predicted class, making these predictions less reliable compared to others with more votes. Hence, we ensure that these data points are put in the train set, and not the test (while training, we give such data points less weight).

The resulting **training** data set has around **29,500** data points, each with **4920** features. The **test** set has around **3,000** data points, also with **4920** features each.

Models

KPI / Evaluation metric

Since our models predict probability distributions, a natural choice for our KPI / evaluation metric is the Kullback–Leibler (KL) divergence between two probability distributions p = [p_0, … , p_n], and q = [q_0, … , q_n] given by p_0 * log(p_0/q_0) + … + p_n * log(p_n/q_n), where p is the “true” distribution, and q is the predicted distribution. This metric is also used in the Kaggle competition.

Models

Since we are ultimately solving a classification problem, we considered the following natural classification models: Naive Bayes, Logistic regression, Random forest, XGBoost classifier, and CatBoost classifier.

But first, a natural baseline model is to assign to each class simply the proportion of the data points in the training set with that class as the expert consensus. Namely, the baseline assigns to each data point the constant distribution [p0, p1, p2, p3, p4, p5] where p0 = percentage of Seizure votes in the training set, etc.

For each of the other models, we trained them on the combinations of the features extracted from data preprocessing steps 3 and 4, and 4 and 5. Out of these, the models generally performed better on the

features extracted from steps 4 and 5 (i.e., on the features extracted from the Kaggle spectrograms and the features extracted from the Mel spectrograms, respectively).

Furthermore, as an attempt to denoise our data (since our combined training data set has 4920 features), we performed a feature selection step using CatBoost (with defualt parameters). A 5-fold cross validation on the train set is used to compute the average feature importance scores, whereby feature importance across each fold was obtained on the validation set and then averaged. Top features up to 90% of cumulative feature importance were then selected. This subset, which we will call SF (for selected features), consists of a truncated set of 1998 features.

Average KL divergence across 5-fold CV

We reran the 5 models mentioned above on the subset SF of the features, and they yielded better average KL divergence across 5-fold cross validation. We record the KL divergence of each of these models trained on SF (except for Baseline) - after some hyperparameter tuning.

1. **Baseline:** 1.36
2. **Naive Bayes:** 16.72
3. **Logistic Regression (multiclass):** 1.17
4. **Random Forest:** 0.81
5. **XGBoost:** 0.78
6. **Catboost:** 0.79

Test set performance

Our best performing model is the XGBoost classifier with optimal parameters (found using the hyperopt python library):

{'colsample_bytree': 0.6376657493489575, 'gamma': 0.17012927852299328, 'learning_rate': 0.13440876516049854, 'max_depth': 3, 'n_estimators': 470, 'reg_lambda': 2.1752761618923397, 'subsample': 0.7988428380922181}

On the test set, this model yields:

KL divergence: 0.71
Accuracy: 0.53
Weighted Precision: 0.58
Weighted Recall: 0.57

On Test Set

KL Divergence: 0.71 Balanced Accuracy: 0.53

Predicted \ True	Seizure	LPD	GPD	LRDA	GRDA	Other
Seizure	433	57	97	32	35	117
LPD	8	276	16	41	7	60
GPD	36	13	248	2	2	26
LRDA	3	10	0	41	8	26
GRDA	30	18	26	55	285	66
Other	125	152	48	56	105	415

Confusion matrix

	Precision	Recall
Seizure	0.56	0.68
LPD	0.68	0.52
GPD	0.76	0.57
LRDA	0.47	0.18
GRDA	0.59	0.64
Other	0.46	0.58

Weighted Precision: 0.58
Weighted Recall: 0.57

Weighted (# of true instances of each class) average of individual precision and recall values to account for class imbalance

This model, along with its performance on the test set, is saved in scripts/models/best_model_and_performance.

Strengths and weaknesses of our model, and future research

As apparent from the confusion matrix, our best model performs reasonably well for classifying Seizures, LPD, GPD, and GRDA, but performs poorly at detecting LRDA. For future research, we hope to improve our model's performance in detecting LRDA, as well as improving the precision and recall for the other classes by extracting more sophisticated features following state-of-the-art EEG literatures. We also hope to run various deep learning models, such as CNNs, on such features.

References

[1] Leal, Adriana, et al. "Unsupervised EEG preictal interval identification in patients with drug-resistant epilepsy." Scientific Reports 13.1 (2023): 784. [2] Jing, Jin, et al. "Development of expert-level classification of seizures and rhythmic and periodic patterns during eeg interpretation." Neurology 100.17 (2023): e1750-e1762.