# Predicting Brain Spectrogram Diagnosis

Kamron Mojabe, Matthew Mollerus, Jailynne Estevez, Michelle Sinani

W207 Section 6

Berkeley
UNIVERSITY OF CALIFORNIA

# Dataset

**Introduction:** The HMS - Harmful Brain Activity Classification competition, hosted on Kaggle, challenges participants to use machine learning for a crucial healthcare application. By analyzing EEG data through spectrograms, the goal is to identify harmful brain activity, which can have significant implications for diagnosing and treating neurological disorders.

**Goal:** The main objective is to harness the power of spectrograms in classifying harmful brain activity. Spectrograms, as visual representations of the spectrum of frequencies of a signal as it varies with time, provide a unique approach to understanding brain activity.

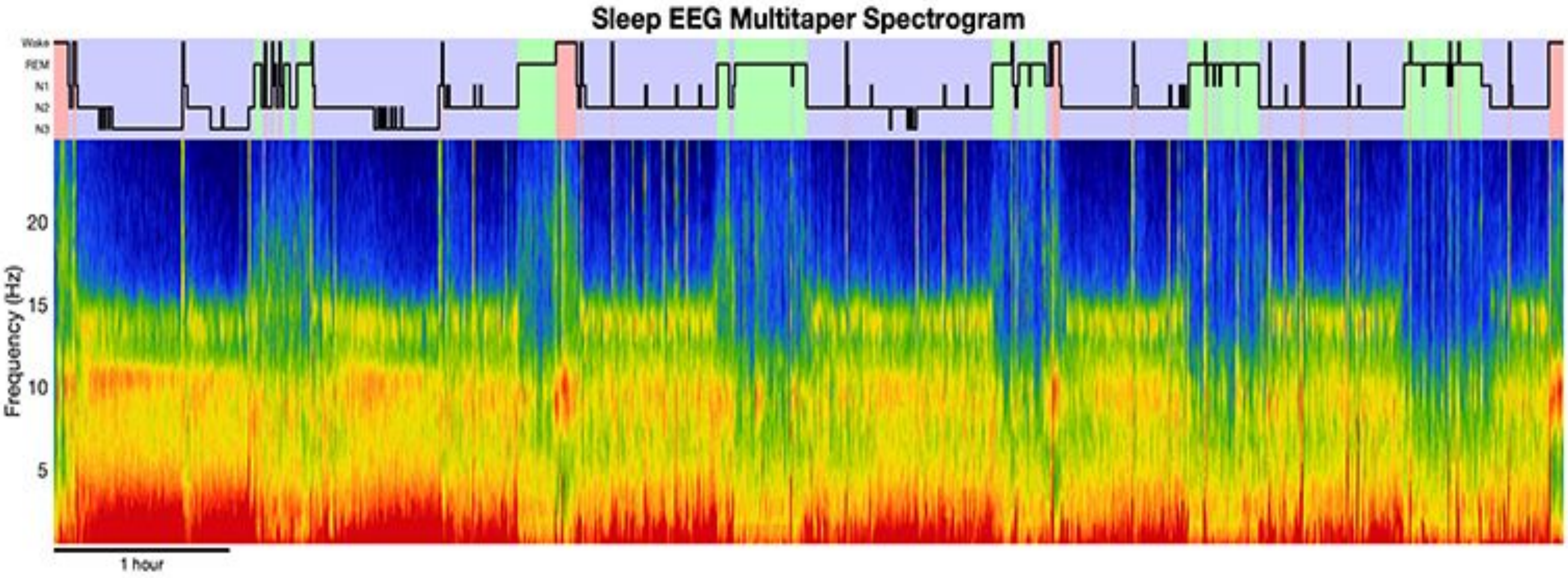
**Source of Data:** Specialized dataset provided on Kaggle, under the title "HMS - Harmful Brain Activity Classification."

**Purpose:** The core aim is to classify harmful brain activity accurately, which could potentially lead to advancements in medical diagnostics and treatment strategies.

**Dataset Structure:**

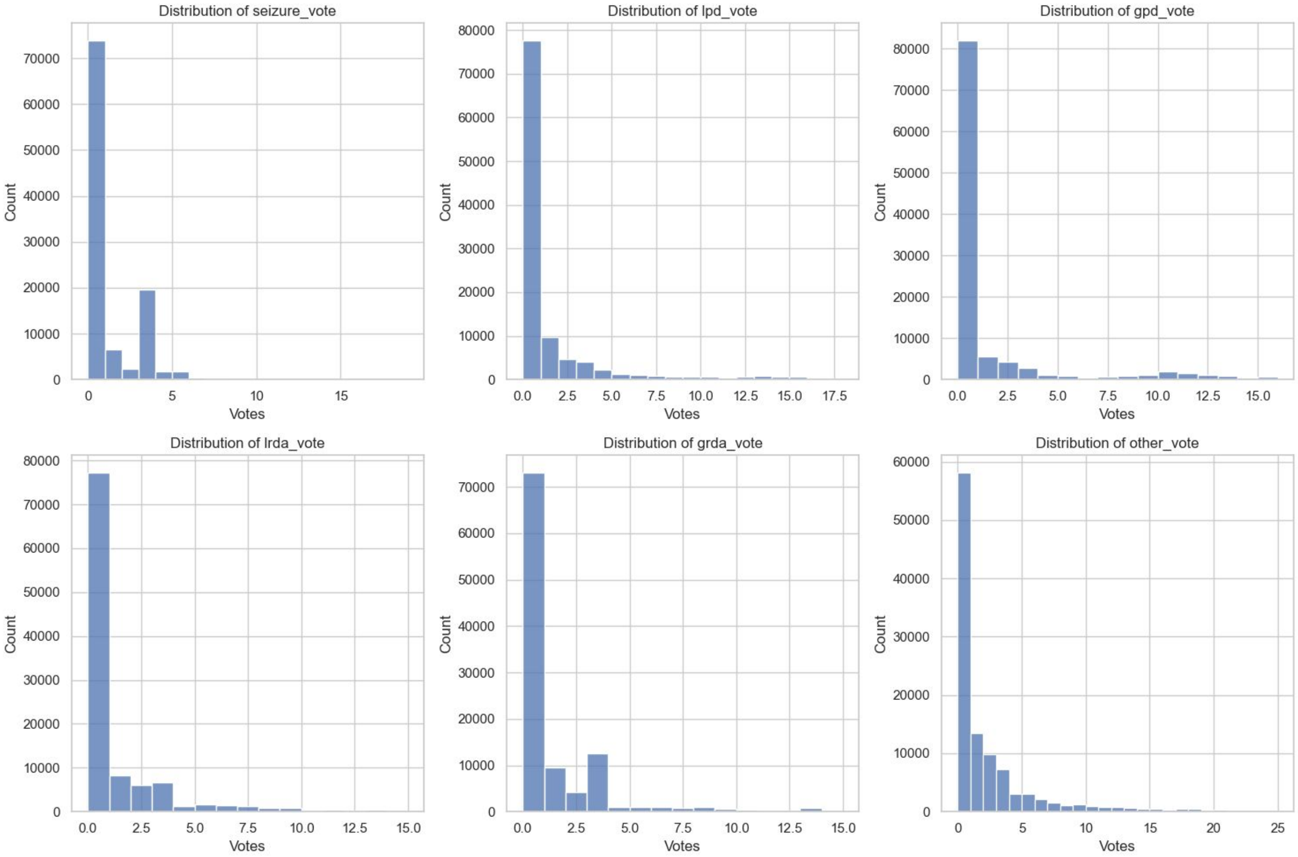
- **specs.npy File Overview:** The dataset is encapsulated in a file named **specs.npy**, a compact collection of spectrograms
- **Data Organization:**
  - The file is structured as a Python dictionary, where each key corresponds to a **spectrogram_id**
  - The values are NumPy arrays with a shape of (time, 400), where each row represents a 2-second interval, making each spectrogram vary in length
  - **Selection Criteria:** To determine relevant segments of the spectrograms for classification, the **eeg_label_offset_seconds** column from the train.csv file is used. This method ensures that the analysis is focused on specific intervals of brain activity of interest

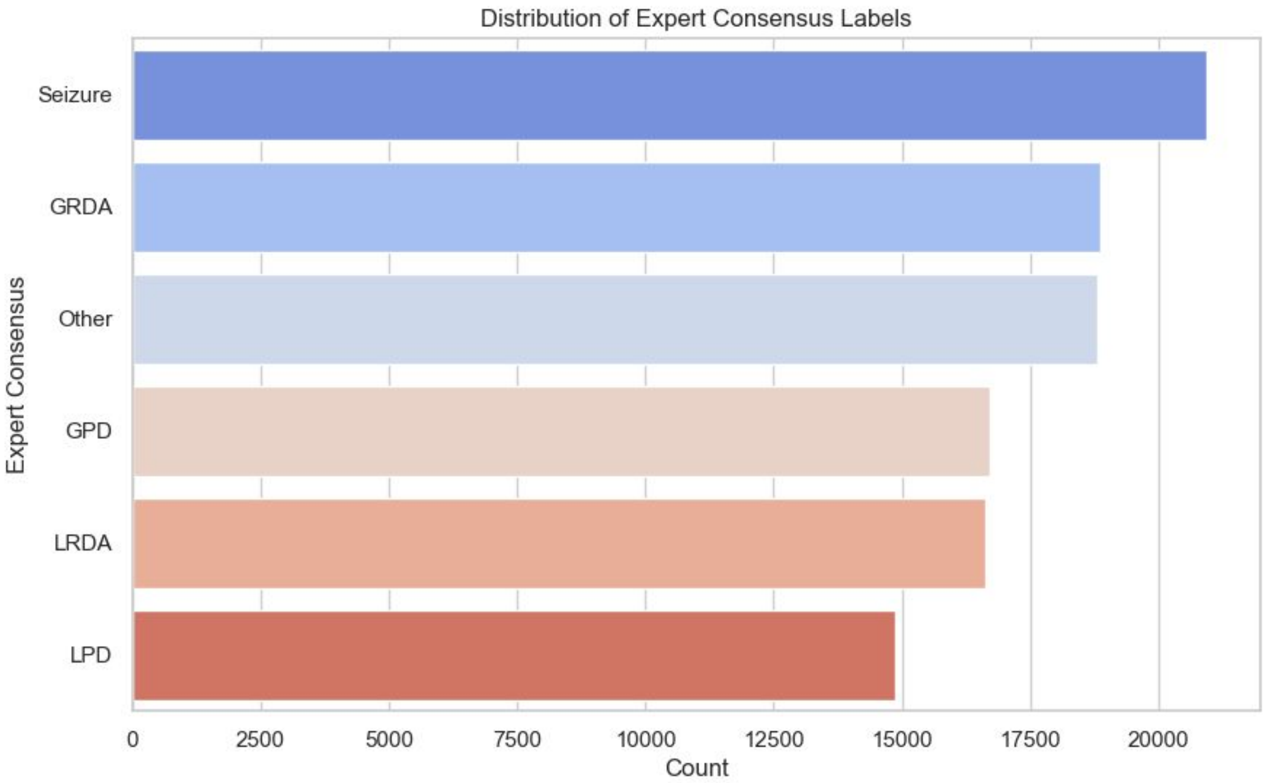
# Example Spectrogram



# Exploratory Data Analysis

## Distribution Expert Votes by Diagnosis





| Diagnosis | Count |
|---|---|
| Seizure | 20,933 |
| Non Seizure | 85,867 |

**Main Takeaway:**

Seizures make up a larger consensus, backing our focus on seizures for our models

# Data Pre-Processing

- **Objective:** Prepare EEG spectrogram data for machine learning modeling.
- **Tools and Libraries:** Utilized pandas for data manipulation, TensorFlow and Keras for deep learning operations, and Numpy for numerical computing.

## Steps Involved:

### Data Loading & Handling:

- Loaded training and testing datasets from specified paths.
- Fill missing values 'NaN' with '0' to ensure data completeness.
- **Apply binary label classification** to the expert consensus for seizure diagnosis.

### Data Transformation:

- Remove time index from spectrogram data to simplify data structure for modeling.
- Data Type Conversion: Convert data to 'float32' for efficient numerical operations.

### Parallel Processing & Dataset Construction:

- Utilized parallel processing to expedite the reading and transformation of spectrogram data.
- Constructed structured training and validation datasets with appropriate preprocessing for model input.

### Spectrogram Processing:

- Implemented a function to read spectrogram data from .parquet files, fill missing values, and convert to numpy arrays.
- Processed spectrograms to ensure a consistent shape and format for model input.

### Dataset Creation for Modeling:

- Defined functions to decode spectrogram paths into tensors suitable for TensorFlow processing.
- Built TensorFlow datasets with appropriate preprocessing, caching, and batching for efficient training.

# Approach & Models

# Baseline Model

- Aimed to establish a benchmark using the dominant class prediction.
- Demonstrated the necessity for a more sophisticated modeling approach due to the baseline simplistic nature.

## Model Development:

### Logistic Regression:

- A simple logistic regression model was implemented as an initial attempt to classify the processed EEG spectrograms.
- Utilized a basic architecture with a flatten layer followed by a dense layer with a sigmoid activation function.
- The model was compiled with the binary cross-entropy loss function and optimized using SGD.

### Training and Validation:

- The logistic regression model was trained on the preprocessed dataset over 5 epochs.
- Employed batch processing and validation steps to monitor performance and overfitting.

### Performance Evaluation:

- The model's performance was evaluated on a validation set, considering accuracy and F1 score as primary metrics.
- Despite the simplicity of the logistic regression model, it served as a foundational step towards more complex neural network architectures.

# Model

## Iteration from baseline logistic regression model

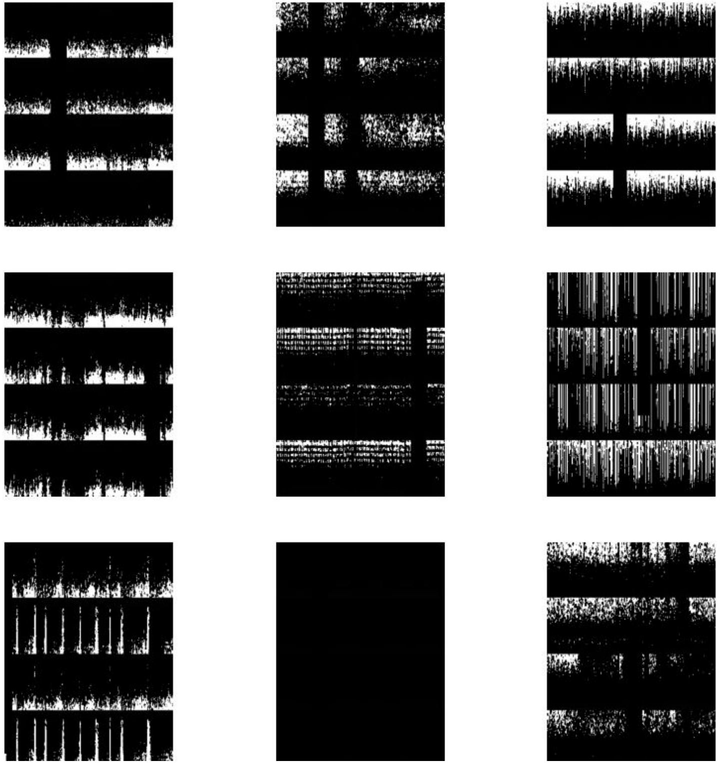
### Architecture Overview:

- **Design**: Sequential CNN with input adaptation, three convolutional layers (16, 32, 64 filters), max pooling, and batch normalization.
- **Feature Processing**: 2D to 1D conversion through flattening.
- **Decision Making**: 128-unit dense layer with dropout; sigmoid neuron for binary output.
- **Optimization**: Adam optimizer, binary cross-entropy loss, focus on accuracy.

### Enhancements:

- **Data Augmentation**: On-the-fly adjustments (flips, brightness, contrast, MixUp, random noise, random cutout) to increase robustness
- **Class Balance**: Computed weights to offset class imbalances.
- **Smart Training**: Employed EarlyStopping and ModelCheckpoint for optimization during augmented dataset training.

*Note: Architectural details and augmentation strategies work in concert to combat overfitting and ensure a generalizable and efficient learning model.*



# Tuning and Improvement

## Decoding

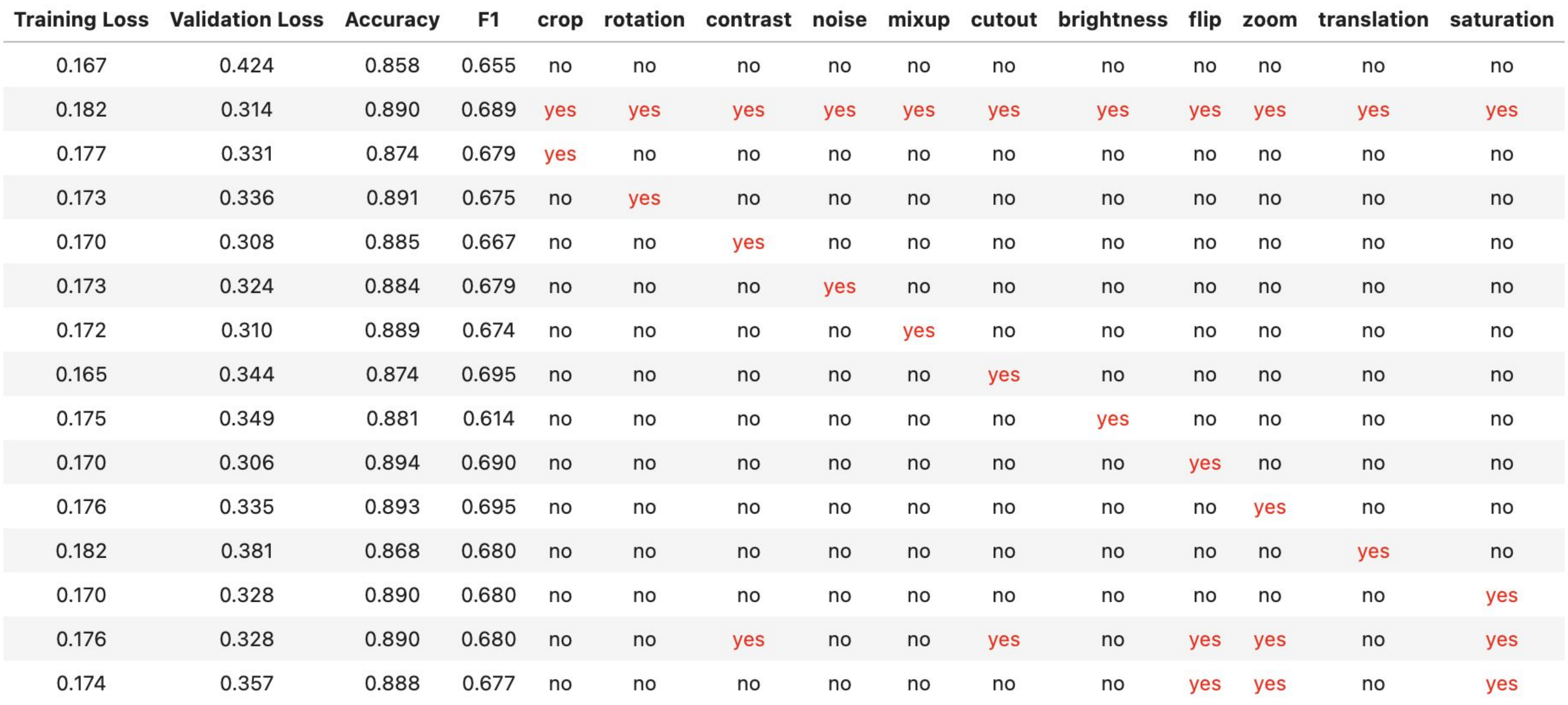
### Signal Preprocessing:

- **Data Reshaping**: Transformed incoming data into .npy format for uniformity and ease of processing.
- **Header**: Removed header tags to focus on the raw signal data.
- **Subsample Extraction**: If an offset is specified, extract labeled subsamples from the signals to analyze distinct events.
- **Spectrogram Standardization**: Applied padding to the spectrogram data to ensure a consistent shape across the dataset.

### Label Processing:

- **One-Hot Encoding**: Categorical labels converted into a one-hot encoded tensor - facilitating clear-cut classification.
- **Signal-Label Pairing**: Combined the processed signals with their corresponding labels to map each spectrogram path to its offset, preparing the data for model input.

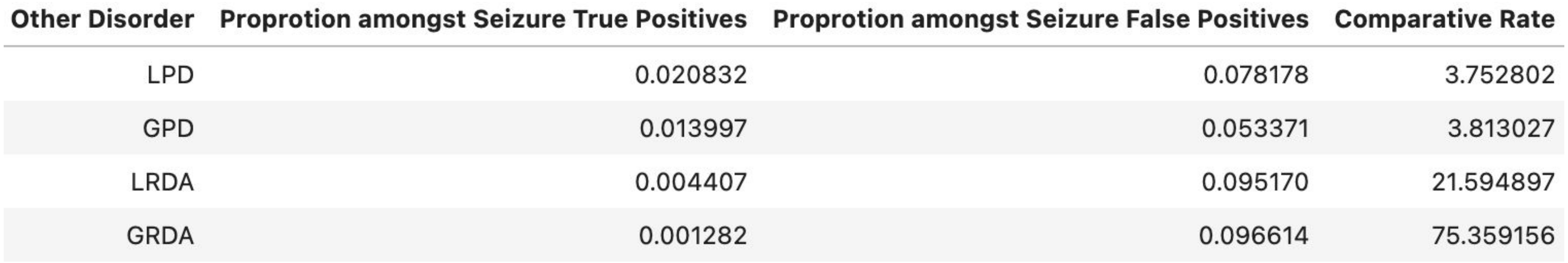
# Tuning & Improvements



| Training Loss | Validation Loss | Accuracy | F1 | crop | rotation | contrast | noise | mixup | cutout | brightness | flip | zoom | translation | saturation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.167 | 0.424 | 0.858 | 0.655 | no | no | no | no | no | no | no | no | no | no | no |
| 0.182 | 0.314 | 0.890 | 0.689 | yes | yes | yes | yes | yes | yes | yes | yes | yes | yes | yes |
| 0.177 | 0.331 | 0.874 | 0.679 | yes | no | no | no | no | no | no | no | no | no | no |
| 0.173 | 0.336 | 0.891 | 0.675 | no | yes | no | no | no | no | no | no | no | no | no |
| 0.170 | 0.308 | 0.885 | 0.667 | no | no | yes | no | no | no | no | no | no | no | no |
| 0.173 | 0.324 | 0.884 | 0.679 | no | no | no | yes | no | no | no | no | no | no | no |
| 0.172 | 0.310 | 0.889 | 0.674 | no | no | no | no | yes | no | no | no | no | no | no |
| 0.165 | 0.344 | 0.874 | 0.695 | no | no | no | no | no | yes | no | no | no | no | no |
| 0.175 | 0.349 | 0.881 | 0.614 | no | no | no | no | no | no | yes | no | no | no | no |
| 0.170 | 0.306 | 0.894 | 0.690 | no | no | no | no | no | no | no | yes | no | no | no |
| 0.176 | 0.335 | 0.893 | 0.695 | no | no | no | no | no | no | no | no | yes | no | no |
| 0.182 | 0.381 | 0.868 | 0.680 | no | no | no | no | no | no | no | no | no | yes | no |
| 0.170 | 0.328 | 0.890 | 0.680 | no | no | no | no | no | no | no | no | no | no | yes |
| 0.176 | 0.328 | 0.890 | 0.680 | no | no | yes | no | no | yes | no | yes | yes | no | yes |
| 0.174 | 0.357 | 0.888 | 0.677 | no | no | no | no | no | no | no | yes | yes | no | yes |

# What Our Model Misses

- **Probabilistic Classification:** Can use disagreements in expert voting to better model; use loss metrics such as KL divergence
- **Seizure Detection Focus**: Currently, our model excels at distinguishing seizure events but struggles with other neurological activities.
- **False Positives Noted**: Non-seizure activity is often misclassified as seizures, indicating a potential overfitting to seizure-like features.
- **Classification Challenges**: The complexity of brain activity patterns calls for more sophisticated discrimination capabilities within our model.
- **Future Direction**: Aim to enhance feature extraction and incorporate multi-label classification strategies for improved accuracy across diverse neurological conditions.



| Other Disorder | Proprotion amongst Seizure True Positives | Proprotion amongst Seizure False Positives | Comparative Rate |
|---:|---:|---:|---:|
| LPD | 0.020832 | 0.078178 | 3.752802 |
| GPD | 0.013997 | 0.053371 | 3.813027 |
| LRDA | 0.004407 | 0.095170 | 21.594897 |
| GRDA | 0.001282 | 0.096614 | 75.359156 |

# Conclusions

## Limitations:

- **Dataset Complexity:** Handling a vast 26GB dataset with non-standard .parquet and .npyc file formats posed challenges.
- **Diverse Classifications:** The need to differentiate between multiple disorder types complicated the modeling process.
- **Operational Hurdles:** Issues with data operationalization impacting the efficiency of our workflow.

## Future Research:

- **Expand Classification Scope:** Move beyond binary seizure classification to include multiple classes and continuous probability outputs.
- **Temporal Dynamics:** Leverage time indices within the dataset for more nuanced temporal feature extraction.
- **Advanced Models:** Explore the efficacy of models like ImageNet-trained CNNs and Light Gradient-Boosting Machine (LGBM) for higher accuracy.

# Ethical Considerations in Dataset Handling

## NeurIPS Checklist Alignment

### Patient Privacy and Confidentiality

- Ensure all data used for training and evaluation are anonymized to comply with privacy regulations.
- Implement robust security measures to prevent unauthorized access and disclosure of personal clinical data.

### Informed Consent

- Inform patients about using and sharing their data, emphasizing the research purposes to ensure informed consent compliance.

### Bias and Fairness

- Identify and mitigate potential biases in the dataset to prevent disproportionate impacts on specific patient populations.
- Develop fair and unbiased models across various demographic groups and medical conditions.

### Accountability and Transparency

- Maintain transparency regarding model limitations and uncertainties.
- Implement stringent documentation and labeling practices to ensure accountability in case of errors.

### Continuous Improvement and Safety Monitoring

- Regularly update and improve models based on clinician feedback and new data points to enhance performance and safety.

### Selection and Randomization in Clinical Decision-Making

- Address potential biases in selecting doctors for EEG and spectrogram data analysis.
- Ensure randomization in doctor selection to avoid biases and enhance the integrity of clinical decisions.

# References

- Kaggle Dataset and Informational
  - https://www.kaggle.com/competitions/hms-harmful-brain-activity-classification/overview
- Discussion Board Resources
  - https://www.kaggle.com/code/awsaf49/hms-hbac-kerascv-starter-notebook
  - https://www.kaggle.com/solution-write-up-documentation
- Neurips
  - https://neurips.cc/Conferences/2021/PaperInformation/PaperChecklist
- Augmentation Resources
  - https://sh-tsang.medium.com/brief-review-specaugment-a-simple-data-augmentation-method-for-automatic-speech-recognition-1ceddfe24e2d
  - https://keras.io/examples/vision/mixup/
  - https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-022-01942-2

# Team Contributions

Our project was a collaborative effort, seamlessly integrating individual contributions into a cohesive whole. Throughout the project, we utilized asynchronous communication via Slack, complemented by strategic meetings to align our goals and progress. Each team member not only focused on their specific areas of expertise but also flexibly contributed where needed to enhance our overall project.

- **Kamron Mojabe**: Focused on model augmentation and loss function adjustments and contributed to the presentation slides.
- **Matthew Mollerus**: Took charge of the initial setup on Kaggle, including data preparation and establishing a baseline model with logistic regression, and made related presentation slides.
- **Jailynne Estevez**: Worked on model augmentation, created EDA graphs, and helped in slide development.
- **Michelle Sinani**: Served as our meeting coordinator, ensuring effective team synchronization, and contributed to model enhancements and slide preparation.

This integrated approach ensured that our project was well-rounded and robust.