Detecting Seizures with Binary Classification Algorithms

Addison Mirliani, Ryan Ghayour, Sophia Wang
CS135
Tufts University
Final Project

Abstract

Electroencephalography (EEG) examinations require the labor of both a specialist/technician and the patient. As technology continues to develop, there is a shift towards automation in the medical field. EEG exams are considered to be a hot spot, given the potential application of machine learning within the field. In this study, binary classification models are utilized to predict the incidence of a seizure in an EEG scan. Specifically, a support vector machine (SVM) and logistic regression (LR) models are utilized to analyze the CHB-MIT Scalp EEG Database [5]. There were 4 SVM models used, each with different kernels: RBF, polynomial, linear, and sigmoid. For all SVM models, there was an accuracy rate of 75%. For the logistic regression model, the accuracy rate was also 75%. However, these rates were generated based on a relatively small data sample – there were issues with running all of the data due to time and space complexity issues on Google Colab.

1 Problem Statement and Goals

Electroencephalography (EEG) is an essential part of a neurology assessment. EEG records spontaneous extracellular electrical activity generated by the brain, dependent on the action potentials of the neurons. Thus, the goal of this project is to create a Machine Learning model that can accurately detect whether a patient is having a seizure by looking at their EEG data.

With our technology, we can solve the issues that are present in the current method of diagnosis. The typical EEG technology requires a large amount of monitoring and manual checking by a trained EEG technician. Our EEG model, however, will do this checking without any need for monitoring by a technician. Our algorithm can be augmented in the future to perform real-time monitoring of the EEG signal, so that we will alert a technician if there are any issues to address. This will save the technician time and energy that would have been spent monitoring the EEG. Instead, the technician will be called in to help only when there is an abnormality. This is also significant because, as previously stated, there are many care centers that don't have access to these trained technicians. With our technology, these care centers will have access to the same level of accuracy as if they had hired an expert themselves.

In the same way a person needs specific training to be able to read an EEG signal, our model will also receive a large amount of training on these EEGs. Our algorithm can recognize patterns very easily and is able to identify signals despite the effect noise may have on readings. The more data we use to train our algorithm, the more accurately our algorithm will detect any abnormalities in real-time. With enough data, our algorithm will be as accurate as a trained professional, and will become even more accurate as we continue to acquire more and more data. This algorithm will be instrumental in the improvement of care for EEG monitoring.

2 Literature Review

Given the broad nature of this project, we divided up our literature review into two (2) parts:

1. Similar Studies
2. EEG Data

The first portion of the literature review was an exploration of current studies to see what was present in this field. While there were similar studies, each of them had a different goal.

The first study we investigated focused on EEG applications to canines [1]. Since canine epilepsy are isomorphic to human epilepsy, researchers sought to forecast seizures using a support machine vector – a method that would parallel ours. The study began with a direct data collection; that is, 16 electrodes were placed onto canine scalps – 8 dogs in total – and the EEG data was taken. From there, the data was translated and bandpassed into 11 frequency bands. Scientists then used an open-source SVM algorithm to classify the data. Prediction performance was measured, and it was concluded that SVM was a better forecaster as compared to a random Poisson forecast of the same data. The main takeaway from this study was the ability to see how SVM was used in the context of an EEG. In other words, seeing the approach in terms of data and model allowed us to gain insight as to how we should proceed forward with our SVM as a predictor.

The next study was an analysis of EEGs using logistic regression and neural networks [2]. The traditional statistical method of logistic regression would be used as the first method, and a multi-layer perceptron neural network (MLPNN) would be the implementation of the artificial neural network. Whereas our method focused on preprocessing the data as a standard EEG reading, this study opted to instead, analyze the power spectral densities – energy per unit time – of the EEG data. The PSDs would be used as inputs to the classifiers, instead of the standard EEG data. This was a major point of distinction. The output, like ours, would be: seizure or no seizure. In terms of performance, the results were striking: MLPNN identified all epileptic and normal cases. Out of these 200 cases, LR misclassified 11 cases (94.5% accuracy rate) and MLPNN misclassified 7 cases (96.5% accuracy rate). We can see from here that MLPNN classified more accurately compared to LR. However, given the relatively high accuracy rates of the classifiers, these could potentially be very useful tools for prediction in the future, as technology continues to develop.

Last, we looked at a study that would compare different modalities of classification [3]. Initially, we had planned to try a neural network, but due to constraints in our data, we weren't able to model with it. Nonetheless, this study still provided numerous insight as to how NNs could be used in classification scenarios. Moreover, the way in which input data was processed also differed – we chose to focus strictly on the type of NN that was used. In this study, 3 neural networks were implemented: fully connected neural network (FCNN), recurrent neural network (RNN), and convolutional neural network (CNN). Although the test data was initially from mice, the researchers later switched to human data as a means for validation. They found that the CNN was the most accurate classifier out of the neural networks, and that it has strong potential to be a real-time classifier given its ability to extract features from the data.

In addition to general EEG + Machine Learning papers, we consulted resources about EEG data. Perhaps the most critical aspect of this project was translating the dataset into usable pieces for our purposes. The main library used was the MNE Library in Python, whose purpose is to analyze human neurophysiological data [4]. By studying the documentation for this library, we learned how to effectively work with our EEG data. This was crucial during the preprocessing step, in which EEG data files had to be read in. Moreover, it was essential when splitting the data into training and testing files – this library granted us the necessary and proper background to be able to work with our EEG data.

3 Model Design

In the most basic terms, this algorithm receives Electroencephalography files accompanied with information on which of said files have seizures, and it uses that information to accurately predict the existence of seizures in future EEG scans. In order to accomplish this, our model is comprised of two main sections:

1. Data Preprocessing
2. Training

3.1 Data Preprocessing

The data that is read in by our model is an EEG scan, in the form of a .edf file. In order to transform this data into a format usable by our machine learning algorithms, we needed to manipulate the data in a few key ways.

The data set used for this model consisted of EEG files and text files, separated into folders. Each folder contained a set of a few dozen EEG scans accompanied with a single text file that contained data for that grouping of Electroencephalography files. This text file would identify which EEG scans contained a seizure, as well as the duration of the seizure if applicable. Due to the size of the EEG files and the limited resources involved, we only were able to read in about a folder or two of data at a time. There were four main end goals for the data preprocessing portion of this model as follows.

1. Convert each EEG file into a numpy array of data. This way, each Electroencephalography scan could be read in by our machine learning algorithms as a single sample appended to the data matrix. We accomplished this using the pyedflib and mne libraries, which both contain necessary tools for extracting data from edf (European Data Format) files.
2. Decrease the size of each numpy array so that we can store the data and pass it into various models without running out of RAM. To do this, we tried downsampling our data. We accomplished this by resampling at a lower frequency and displaying the images of each signal. We chose the signal with the smallest size that still looked similar to and included the most remarkable features that were present in the original graph.
3. Convert all text files with information on samples in our data matrix into one output array, with 1s and 0s corresponding to whether or not each sample contained a seizure. This proved to be a slightly less complicated task, as the formation of the "y" matrix followed closely to the structure of each text file. Each text file was separated by data per EEG scan, which contained a section labelled "Number of seizures in file" followed by a 1 if there was a seizure, 0 otherwise. Evidently, the format of the text files corresponded to our ideal output array. Therefore, our model read in each "number of seizures in file" value, and stored them in order in an array. As will be seen, our model stores each EEG scan in order in the data matrix, so this array served as a sufficient labels array.
4. Split the data into training and testing. We decided to use an 80-20 split so that 80% of the data is training and 20% is testing. We initially tried doing this randomly, but due to the small number of seizures in our data, we needed to intervene to ensure that both our training and testing data included examples of files containing seizures. We ended up using the train_test_split function to create a split that had seizure and non-seizure files in both our training and testing datasets.

3.2 Training

After some deliberation, we decided on the following two (2) binary classification algorithms to train our model to accurately predict the existence of seizures in EEG scans:

1. Support Vector Machine
2. Logistic Regression

3.2.1 Support Vector Machine

An SVM is a classification algorithm that is able to use both linear and nonlinear kernel functions to best separate the data into classes. For the data that our model is working with, each sample consists of signal readings from multiple channels flattened into one row of a numpy array, therefore the theoretic visualization of said data was a less desirable approach to understanding the data. Thus, a support vector machine was ideal, as it allowed us to test various types of kernel functions on our data to see what resulted in the most accurate classification. In our model, we trained our data on multiple different SVM kernel functions in order to determine what fits our processed EEG data best. The different types of kernel functions used were as follows:

1. Radial Basis Kernel Function
2. Linear Kernel Function
3. Polynomial Kernel function
4. Sigmoid Kernel function

3.2.2 Logistic Regression

Logistic Regression is a type of classification algorithm that excels at binary classification. Not only is this powerful classification algorithm quick to implement, it is also less inclined towards overfitting and is very efficient. Considering that our model is attempting to classify samples of large size into binary classifiers, we found that Logistic Regression would make a useful secondary algorithm to train our data with. However, Logistic Regression has a drawback. Logistic Regression assumes by default that the data is linearly separable. For data as complicated as Electroencephalography, it is not a guarantee that a linear classification of the data will suffice. However, given the ease of implementation, we decided that it was certainly worth it to include it in our algorithm. If the model's linear kernel SVM performs well, then using Logistic Regression would be a spectacular way to even more accurately and efficiently classify the data.

4 Model Computation

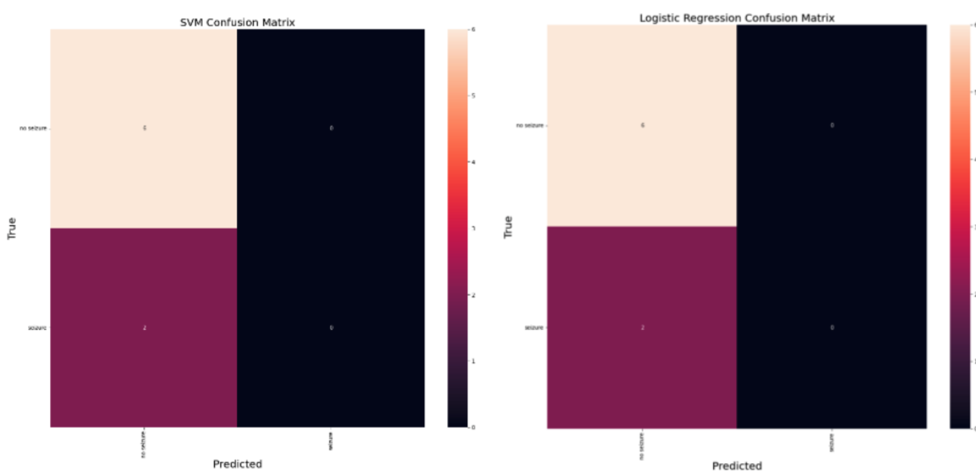

Figure 1: SVM and Logistic Regression Confusion Matrices

5 Error Analysis

As illustrated in the previous section, the Support Vector Machine algorithm (using each of the four kernel functions) as well as the Logistic Regression algorithm all resulted in a 75 percent success rate in correctly classifying EEG scans with seizures vs scans without. Using a confusion matrix (shown above) we arrive at the conclusion that every testing data sample was classified by our model as a non-seizure. The 75 percent success rate corresponds exactly to the 75:25 ratio of non-seizure samples to seizure samples. Despite attempting a random selection of data samples as well as a sorting of data by previously defined data folders within the data set, our model classified all the testing data into the same class.

While unsatisfactory, this result is not unpredictable. EEG files, when converted into csv files, are in the form of matrices with tens of millions of data points. Therefore, given our limited access to storage and RAM, our sample sizes for training and testing never exceeded 30 to 40 samples. During the creation of this model, we estimated being able to use hundreds, if not thousands or tens of thousands of samples while training our model. In the realm of binary classification algorithms, having a sample size of a few dozen is simply not enough to properly train a model. Thus, we conclude that the inaccuracies in classification stem from a lack of training data. Fortunately, Professor Chin was kind enough to allow us to explain this shortcoming in our video and write-up as our computers have shown to be unable to handle running our model on larger data sets.

If we had been given access to more resources, such as unlimited storage and access to more files containing EEG scans with seizures, we would have been able to create a model with more success in predicting these seizures. If we had more RAM, we would be able to load in a lot more of the edf data and we could limit our downsampling, so that our models could be trained using a higher amount of data. In addition, we would be able to create much larger training and testing datasets. Given more access to files containing seizures, we could make training and testing data with a 50-50 split between seizure-containing and non-seizure-containing edf files. In conclusion, our primary limiter and the root cause of error in our model is a lack of RAM, storage, and seizure data.

6 Application

In the future, we plan to use this same technology to improve readings for infant EEG signals as well. This data is a lot more difficult to acquire, but is worth training separately because infant EEG signals contain a lot more noise and an even more highly-trained specialist to read. We would look into this area after proving our model could be successful on adult EEG signals first.

There are many problems with the current neonatal EEG solutions that can result in inaccurate readings and missed diagnoses in infants. In the NICU, EEG readings are essential to early diagnosis of a hemorrhage, brain tumor, cerebral infarction, seizure, and other major health concerns. Because the stakes are so high, the technology used to detect these problems must be extremely accurate and minimize the risk of error as much as possible. The technology used for these infants must also be able to accurately detect these abnormalities in readings early on, in order to call a doctor to action as soon as possible. In the cases of these serious health issues, time is of the essence, and the technology is an integral part of increasing these infants' chances at survival.

Another future direction we plan on taking would be expanding our model to detect different kinds of abnormalities. We could do this with binary classification, and detect whether or not an abnormality occurred. This could be useful because any kind of abnormality in an EEG reading could indicate a serious health issue, so we could detect issues past seizures. Eventually, if this was successful, we could look at multi-class classification that deals with specific issues such as hemorrhage, brain tumor, cerebral infarction, and seizure. The ability to diagnose each of these from an EEG reading could be extremely valuable for patient care and could help physicians act earlier, preventing patients from having long-term complications as a result of any of these conditions being treated too late.

7 References

[1] Brinkmann BH, Patterson EE, Vite C, Vasoli VM, Crepeau D, Stead M, et al. (2015) Forecasting Seizures Using Intracranial EEG Measures and SVM in Naturally Occurring Canine Epilepsy. PLoS ONE 10(8): e0133900. https://doi.org/10.1371/journal.pone.0133900

[2] Ahmet Alkan, Etem Koklukaya, Abdulhamit Subasi, Automatic seizure detection in EEG using logistic regression and artificial neural network, Journal of Neuroscience Methods, Volume 148, Issue 2, 2005, Pages 167-176, ISSN 0165-0270, https://doi.org/10.1016/j.jneumeth.2005.04.009.

[3] Cho, KO., Jang, HJ. Comparison of different input modalities and network structures for deep learning-based seizure detection. Sci Rep 10, 122 (2020). https://doi.org/10.1038/s41598-019-56958-y

[4] https://mne.tools/stable/index.html

[5] https://physionet.org/content/chbmit/1.0.0/