Arshay Rao

Vedparkash Singh

Ivan Cano

Steven Liu

Edmundo Zamora

Cogs 189 De Sa

The Effectiveness of Seizure Detection

Introduction / Motivation

Disorders that affect the Central Nervous System, such as epilepsy, are incredibly debilitating conditions that have significant detriments to a person's cognitive capabilities. These disorders are known to produce seizures, or sudden electrical bursts can cause temporary or long-term abnormalities in muscle movements or brain functions. Being able to anticipate or correctly diagnose seizures as they are happening is essential to treating patients who suffer from CNS disorders and ensure that their conditions do not worsen. To achieve this, we are interested in examining the accuracy and recall of EEG data. Accuracy is needed to ascertain whether subjects are being labeled as having had a seizure while recall tells us what percentage of these people are being *correctly* labeled as positive. The dataset that we have chosen features 4097 data points corresponding to brain activity across a roughly 23.5 second period for each of 500 patients. By looking at the number of patients who suffered a seizure during this time, we can determine whether there are instances of patients who are not being correctly identified as having had a seizure. We will use a variety of supervised classification algorithms to accomplish this, such as, K-Nearest-Neighbor, Logistic Regression, Stochastic Gradient Descent, Gaussian

Naive-Bayes, and Random Forest as well as an Artificial Neural Net and the unsupervised algorithm Principal Component Analysis.

Related Works

We did not find any class material that focused specifically on seizure detection and many of the algorithms that we have chosen to use for this project were not discussed in class. We chose to use these algorithms because we were comfortable with their implementation and wanted to see how they would perform in the context of EEG data. The exception to this is PCA, which was discussed during class lectures. PCA is an extremely helpful method of reducing dimensionality which we thought would help make our data easier to read. Overall, we believe that our research on this project is important in the context of the entire class because it allows us to take a deeper look at the types of EEG data that is available and apply knowledge of machine learning to it.

We were able to find a multitude of studies online that investigated similar concerns with seizure detection in EEG data using machine learning. One article, written by Siddiqui, Morales-Menendez, Huang, and Hussain (2020), analyzes the use of multiple machine learning algorithms in order to determine which one is most useful for further studies for researchers with data science backgrounds. Their dataset contained EEG data for patients along a fixed duration, much like our own database. The writers of the article analyzed multiple types of seizures using the classifiers ANN, KNN, SVM, and a single Decision Tree. They decided to look at the “line length” and “relative power” of plots of various patients’ EEG activity in order to determine the strength of seizures. Another interesting application of their research was using machine learning for the purpose of localizing seizures. In their own words, “machine learning classifiers have not

been extensively applied for seizure localization." They were not able to gain much insight into this application since they could not identify the location of seizures within different lobes of the brain. Through their research, they found that Decision Trees provided the most predictive accuracy when applied to EEG data. Our project focuses not only on accuracy, but also recall to verify whether the accuracy of our models is legitimate.

Methods

After successfully importing our dataset, our first step is to clean the data and ensure that no values are missing. Initially we were given 5 different data classes where 2-5 meant that there wasn't a seizure and 1 meant that they did record the seizure activity. Once we have confirmed this, we take the final column of the table and treat it as our label. We then converted the y column to be zeros or 1. This is because the value of this column contains a binary value which represents whether a patient experienced a seizure during the course of the experiment. Statistics taken from our data showed us that only around 20% of observed patients were experiencing seizures which makes sense because only 1 out of the 5 files had collected seizure activity. We decided to test out multiple different models and see which one best fits and predicts if someone has had a seizure or not. For each of these models, we use functions that are built into the sklearn package in Python. Functions that allowed us to predict scores were also extremely valuable.

The PCA allows us to reduce the dimensionality of our data. We performed PCA on the data with the intent of making our data easier to read and interpret. As stated earlier only 20% of patients experienced a seizure, and this was corroborated by our PCA plot. At this point, we were ready to split our data into our training, validation, and testing sets. We decided that we could split our 11500 data points into our training and testing sets by using a 60-20-20 split as a ratio of

training:cross-validation:testing. After performing brief tests on our datasets to check for bad data, we started building our supervised models.

For K-Nearest-Neighbors (KNN) our first model uses the K-Nearest-Neighbors (or KNN) algorithm. We thought this would be an excellent first method because of its easy implementation. KNN is a classification algorithm that uses various surrounding points to determine classification of a singular point. In order to run KNN on our data, we had to choose different numbers of neighbors (1-10, 20, 30, 40, 50, 70, 100, 150, 200, 250 300). We were then able to iterate over this list and predict accuracy and recall scores for our training and validation sets. We then plotted the amount of neighbors and our recall/accuracy scores on a plot as our x and y axes respectively.

Our second model was created using Logistic Regression (LR) . LR works by attempting to predict how probable a target variable is, which would be the odds of a seizure occurring in our case. LR is a useful algorithm for us because it addresses the problem of overfitting. In order to make sure that our model does not overfit our data, we make use of regularization to reduce bias and variance. Regularization is represented by the parameter *C*. In our LR plot, the values of *C* are varied between between 0.05 and 1.05 with increments of 0.05. We then predicted the scores for our training and validation sets before graphing them against *C*.

Next, we decided to use the SGD method. With SGD, we iterate through our training and validation sets multiple times according to different alpha values (.000001, .00001, .0001, .001, .01) and select a random sample on which we perform Gradient Descent. We predict and store the scores for recall and accuracy across each iteration and plot them.

Gaussian Naive-Bayes is another predictive modeling algorithm that we used to predict the probability of our binary variable. In order to perform GNB, we had to choose a list of small

decimal parameters containing the values [1e-7, 1e-8, 1e-9, 1e-10, 1e-11]. We can then perform GNB for every iteration of this list using the iterated value as a parameter and predict recall/accuracy for our training and validation sets like with previous algorithms.

We also built a model using the Random Forest algorithm, which functions similarly to KNN in that it is reliant on building tree-like structures between points. We chose RF because it creates decision trees for each data point and then selects the best result. This also means that RF is more beneficial than using a single decision tree, as taking the average of multiple trees helps to reduce overfitting. We iterated over the list of numbers [2, 3, 5, 7, 8, 10] and used each one as the depth parameter for sklearn's built-in Random Forest algorithm. From there, the process was the same as with previous models in that we were able to predict recall and accuracy scores from our training and validation sets.

Our last model was created by an artificial neural network using keras. The optimizer and loss that we used were adam and sparse categorical cross entropy, respectively. After testing three times, we noticed that our ANN was having overfitting issues and it wasn't as efficient as KNN or the Random Forest models.

At the end of each model using the aforementioned classification algorithms, we utilize GridSearchCV in order to collect our testing sets from each model and compare their effectiveness. We performed GridSearch by iterating over each of the models in our pipeline for our testing set and recorded recall and accuracy features. We then appended the best result for each model to a list which was then visualized as a barplot, showing the best accuracy and recall of each model.

Results

We optimized our models by running them through a GridSearch in a pipeline and then recorded the best metrics and hyper parameters for each model. The KNN model performed really well on this dataset and through GridSearch performed the best when only looking at the single closest neighbor. It also had the highest recall which means that it didn't incorrectly classify the wrong data as seizure data. This suggests that the data was well clustered between seizures and non-seizure recordings and that differentiating the data wouldn't have been that complicated.

Next we observed the Logistic Regression model which also performed well, it had it's optimized parameter at C = 0.95, this shows us that the model's complexity wasn't too high. C is inversely related to the lamda regularizer and a 0.95 C seems to show that there didn't need to be too much regularization. This shows that the data was linearly separable, which means that the model would be able to classify between seizure and not seizure clearly.

Next, we tested the Stochastic Gradient Descent model where alpha was really small, 0.001. Alpha represents the rate that the cost function decreases when you run gradient descent. Showing that the best hyperparameter was very small shows that the data must have had a lot of variance in that a too big of a growth rate would have made the model miss a local minimum. This shows that the data had a lot of peaks and valleys in it. This model had the worst performance when it came to recall which means that it had a lot of false positives, with a score of 48%, it seems like its almost a fifty-fifty chance of data being classified as seizure data.

The Gaussian Naive-Bayes model also performed really well, it had a really really small smoothing alpha value, the closer the alpha is to 1, the more likely the data is uniformly distributed. Our best performing apha is 1*e^-0.7 which means that the data was very far away

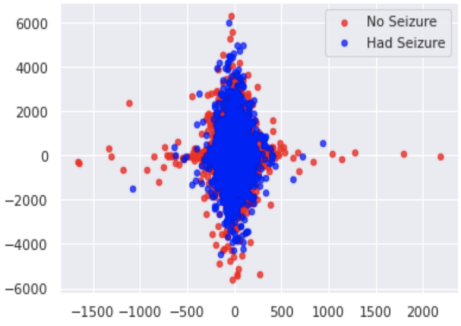
from a uniform distribution which reveals to us that seizures aren't that prevalent in our data, something we had initially already confirmed.

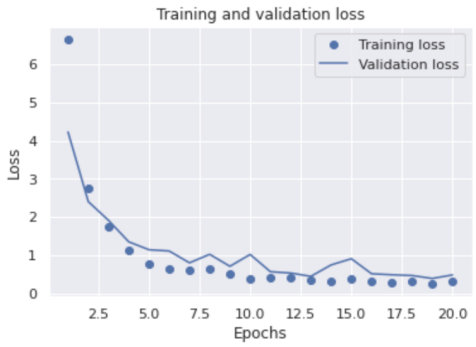
Lastly we checked the Random Forests model which performed the second best, it's hyper parameter was a max-depth of 10, this is a relatively small number which shows that the algorithm was able to quickly and efficiently classify the whole data. This also makes sense because we have determined that our data only had 20 percent data where the participant was having a seizure. Furthemore this also shows us that there wasn't a lot of variability in the data which can be a little concerning because it becomes easier to overfit.

Model	**Recall**	**Accuracy**	**Optimized Hyper Parameters**
KNN	**0.99819**	**0.9867**	**# Neighbors = 1**
Logistic Regression	**0.9728**	**0.7488**	**C = 0.95**
Stochastic Gradient Descent	**0.4881**	**0.774**	**Alpha = 0.001**
Gaussian Naive-Bayes	**0.9368**	**0.9519**	**Smoothing = 1*e^(-07)**
Random Forest	**0.9916**	**0.9767**	**Max Depth = 10**

Discussion

After finishing all of the rankings and completely testing out our models, the outcome kind of makes sense. We personally expected a lot more from the neural network but we had issues with optimizing it in that it kept on overfitting the data. This also motivated us to not include the Artificial neural network in the rankings because even though it had an accuracy of 90 percent, we believe that with further data and testing that it would fail, as you can see in the training and validation loss graph, the validation was lost a lot more than in the training. The other interesting thing that we discovered was that the clustering algorithms worked better than the classification algorithms. Initially this was kind





of confusing but after further inspection of the data, we realized that clustering algorithms would thrive with this data in that the majority of the data was non-seizure data so the 20 percent that was, would have been closely near the same areas. We are also able to confirm this claim of ours through the visualization of the PCA results that showed seizure data points to mostly follow the x = 0 line with some variance which can be seen in the top left image of this page. The classification algorithms did ok, we wonder if manipulating the data in a different way would have improved their performance. If we had more time, we would have improved our Artificial Neural Network more to where it wasn't overfitting and also manipulated the data more so that our models would classify better and more efficiently. We would also have implemented a 5-fold cross validation on our data just to mix our data better and really make the models work, this way we would have had a better representation of how our models performed against each other. Another place for improvement would be on our metrics that we used to judge the models with, we should have used the ROC and AUC curves as well as precision and the f1 metrics.

Works Cited

De Sa, V. (2021). “Cogs 189: Brain Computer Interfaces”. University of California San Diego. Class Lecture.

Siddiqui, Morales-Menendez, Huang, and Hussain (2020). “A review of epileptic seizure detection using machine learning classifiers”. US National Library of Medicine. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7248143/

Temko, A., et al. “EEG-Based Neonatal Seizure Detection with Support Vector Machines.” *Clinical Neurophysiology*, Elsevier, 14 Aug. 2010, www.sciencedirect.com/science/article/pii/S1388245710006036.

UR. Acharya, SV. Sree, et al. “EEG Seizure Detection and Prediction Algorithms: a Survey.” *EURASIP Journal on Advances in Signal Processing*, SpringerOpen, 1 Jan. 1970, link.springer.com/article/10.1186/1687-6180-2014-183.