Nicolas Wagner

CSC 597 Statistical Learning Project: EEG Mental State Classification

Introduction

Generally, seizures are interruptions in normal bodily functioning caused by a sudden, simultaneous discharge of neurons in the brain [1]. Epileptic seizures are those which are unprovoked – meaning not caused by isolated events such as stress or fever – and originate, specifically in the brain [1]. In the long-term, epilepsy refers to such recurrent seizures [1]. Epilepsy is among the most frequently occurring neurologic afflictions, effecting as much as 1% of the population, and to this day there is not a complete understanding of the detailed underlying causes for the condition [1]. Despite this, seizures are readily discernable into several categories: atonic, generalized tonic-clonic, absence, and myoclonic [1]. Each category has defining associated physical cues varying from a lack of response to stimulus, to uncontrollable muscle convulsions.

To diagnose seizures, several tests are presently employed including the electroencephalogram (EEG) as well as computed tomography (CT) and magnetic resonance imaging (MRI) scans. Of primary interest is the former, however in practice the latter two are useful ancillary tests to provide further clarity for the diagnosis. The EEG is a device appears as a network of electrodes placed on the surface of the head and usually concentrated on the scalp. The tool is used to measure the electrical activity of the brain and detects abnormal brain waves. The monitoring sessions vary in length but can be administered after an epileptic seizure to attempt to gauge the cause or focal point for the seizure [2].

As can be expected, there is a significant demand for understanding and predicting the incidence of seizure in the brain [2]. Some of the work at the forefront of the field was that of Shoeb and Guttag which yielded a classifier with a 96% accuracy in detecting seizure occurrence on a curated dataset [3]. The model that performed best was a support vector machine (SVM) approach [3]. This work was only a portion of Shoeb's larger doctoral thesis which sought to develop patient-specific classifiers for seizure occurrence, yielding a classifier with 95% accuracy but having a much wider and more dynamic approach [4]. Subasi, Kevric, and Canbaz built on this work and leveraged a hybrid approach enlisting the support of a distinctly artificial intelligence approach, Particle Swarm Optimization (PSO), coupled with the SVM to form the novel PSO-SVM. This model was able to reach 99% accuracy on their own curated dataset [5]. Finally, Müller et al. branched out into a more general approach to predict the general mental state of patients achieving a classifier with 98% accuracy for this purpose [6].

This last work inspired the project to approach the challenge of classification for patients as a multi-class problem rather than the standard binary approach of seizure detection. There are distinct benefits to being able to properly determine the mental states of a patient such as determining sleeping stages, tiredness, or monitoring emotions [6]. The benefit is having a more flexible model that will work in more complicated scenarios, not just determining if a seizure has occurred. Thus, the goal is to develop a model which can accurately distinguish between an individual who is experiencing a seizure and one who is in another, healthy state based on EEG recorded values.

The other classifications are patients who had their eyes open (group 5), patients who had their eyes closed (group 4), patients having tumors in their brain with brain activity recorded from a healthy region of the brain (group 3), patients having tumors with brain activity from the tumor region (group 2), and patients having an active seizure (group 1).

Methods

The data was split 80/20 into training and testing sets and for the most part 10-fold cross validation was employed as a means to validate the training of the models. Where models trained prohibitively long 5-fold cross validation was leveraged to speed up convergence.

Dataset:

The selected dataset is from the UCI Machine Learning repository originally although it had been removed so a copy had to be acquired from Kaggle. The data was originally gathered as EEG readings taken from 500 individuals over the course of 23.5 seconds. During this time-period, each individual had 4097 data points sampled from the EEG. The data was restructured by the authors so that each individual's data points would be split into 23 groupings of 178 sampled EEG data points. The 4097 observations per patient were split into 23 new records with each record now being roughly 1 second of recorded EEG activity (178 sampled EEG datapoints). The total number of data records now became 11,500 due to each patient now having 23 records associated with them. The 178 data parameters all contain numerical integer values from the EEG reading and there are no missing values present. An initial column is present at the start of the record to uniquely identify the recording and tie it back to the original individual. A final column is the output/classification column which is a categorical integer in the set 1-5. The value 1 indicates that the record is a recording of an active seizure, a value of 2 indicates the recording is from a region containing brain tumors, a value of 3 indicates the recording is from a healthy brain region from a patient who has brain tumors elsewhere, a value of 4 is a patient whose eyes were open for the recording, and a value of 5 is a patient whose eyes were closed for the recording.

Typical research with this data set seeks to perform a binary classification of the data with the seizure class (1) against the other 4 categories. For this project, the goal would be to classify the data records into the proper version of each of their 5 categories rather than a binary sectioning. Each class has exactly 2,300 records associated with it making the task of classification easier by having an already balanced dataset. Interestingly, this would pose an extra hurdle for the explorations into binary classification as the data would be unbalanced 4 to 1.

Models:

This being a classification problem, the choice for models to apply are logistic regression, the k-nearest neighbors (KNN) classifier, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), decision trees (DTs), random forests (RFs), bagging, and support vector machines (SVMs) based on what we learned in class. However, this is a problem necessitating classification with greater than 2 classes (not a binary problem). Boosting was used in class as a regression and binary classification technique, but a distribution of "multinomial" exists for the function in R which works for multi-class problems, so this was also employed. Logistic regression works for binary classification problems and can be adapted to work for larger class problems but is remarkably unstable for these cases and frequently outperformed by LDA and QDA models, so it was not incorporated as part of the test suite.

The KNN classifier functions by taking the *k* closest neighbors to a given vector of X predictors (i.e. a record of data excluding the last column) and taking the majority class to which those voting members belong. Closeness is most typically measured using Euclidean distance between the vectors. Here, the values of *k* were tested using 1, 2, 5, 10, 25, and 50.

LDA works by fitting a vector of *v* values of the same length as the X predictors which are pairwise multiplied together and has a constant *a* added. These *v* values are chosen so that the means of the different classes are *maximized*. Those predictors, *x* values, which have more impact will ultimately have larger *v* values because they are the discriminating factors. LDA makes the assumption that every class has the same covariance and variance. QDA works the same as LDA except it estimates separate variances for the difference classes. This separation of variances allows for the boundaries between regions to become quadratic.

DTs function by dividing the possible values for each vector X into distinct regions R. Each X which is in a particular region R_j receives the same class prediction. They are referred to as trees due to the branching nature of the separation of the regions. The regions themselves can be thought of as terminal nodes in the tree and are called "leaves". Where to split the regions is determined by splitting on every possible predictor and every value *s* less than and greater than the predictor. The best split is determined by the value with the lowest mean squared error (MSE) on the training data. The class of the region is determined from the most frequently occurring class of the records which fall into the region. Trees with too many regions may overfit their data so to combat this they can be *pruned*, a procedure by which the number of regions is reduced. The splits in the data are re-combined based on the lowest MSE in the training data thereby merging overfit regions.

RFs work by reducing the high variance of individual trees. The RFs are grown by only considering a subset *m* of the predictor columns for any given split of a tree. The number of trees to be grown was tested with values 50, 100, 1000, and 5000. The number of predictors to consider was tested with values 5, 10, 13 (the square root of the total number), and 50. The square root of the number of predictors is commonly used for classification problems so therefore it was tested here. Bagging is basically the same as RF except all predictors are considered at each split and this was tacked onto the RF tests as an additional mtry value of 178. Boosting is basically the same as bagging except that each tree is grown sequentially and has at its disposal information from previously grown trees.

SVMs are a more general case of support vector classifiers (SVCs). SVCs functions by attempting to find a hyperplane to separate the data across the predictors. A hyperplane is essentially an n-dimensional line in n-dimensional space, in 2 dimensions it is a line and in 3 dimensions it is a plane. A SVC is deemed a SVM when the boundary is no longer linear. This is achieved by transforming the predictors into a domain where a hyperplane can be used to separate them. The result is a nonlinear boundary in the original problem domain. Both SVCs and SVMs were tested here using cost values of 0.001, 0.01, 0.1, 1, 5, 10, and 100 and gamma values of 0.5, 1, 5, 10, and 50.

Evaluation:

The models were compared on the basis of accuracy as this was the only universal measure available to compare all of the functions used for the given models in R and because the classes were exactly balanced so a model guessing just one class would not appear to be more successful than the others. Exhaustive comparison of large and small tuning parameters for the models was used wherever possible to find models with the most ideal accuracy.

Results

The KNN classifier was built using all the predictor variables performed relatively poorly, showing that with an increasing *k* value the error of the model steadily increased. The best performance had an error rate of 44.94% and this was with a *k* value of 1. This consistent was based on the averaged results of 10-fold cross validation. Figure 1 shows the error rates with respect to the increasing *k* values. The total training took about 2-3 hours to complete.

The LDA classifier was trained against all the columns as well and performed very poorly. It had an error rate of 74.63% which put it slightly better than random guessing for a 5-class problem. The indication here seems to be that the data does not have uniform variances as that is where LDA fails. The tests for LDA were performed using 10-fold cross validation as well and completed in less than 5 minutes. Figure 2 shows the error values for the 10 cross validation runs.

The QDA classifier was also trained using all the predictor variables and had an average error rate of 35.45% which was much better than the other basic models. The indication here is assuming separate variances for the predictor variables makes for a more ideal model. This makes sense as this is at its core time series data and while some points will not be closely related if they are far apart enough in time, the closer the points are in time to one another the more likely they will be related hampering some assumptions of independence with models. The execution of the QDA tests were done using 10-fold cross validation and completed in under 5 minutes as well. Figure 3 shows these error values with the respective run of the cross validation.

The DT was trained on the entire set of predictors and yielded an average error rate of 62.85%. This was moderately better than the KNN classifier, but still a subpar performance. A sample tree pulled from the cross validation showed the classification was only occurring for 3 out of the 5 classes meaning at least a 40% error was incurred by default for not classifying 2 of the classes. This is obviously not ideal and indicates that the model was not learning enough from the data. The models were trained using 10-fold cross validation and completed quickly as well finishing in under 10 minutes. Figure 4 shows the sample tree. Figure 5 shows the error values against their respective runs of cross validation.

The pruned DT was trained the same way as the unpruned DT, but with the pruning applied with 10-fold cross validation. The best pruned model was chosen automatically and used to predict with. The pruned DT had an average error rate of 62.84% which was basically identical to the unpruned version indicating that as suspected overfitting was not an issue here with this model and not enough information was being learned from the data to begin with. Figure 6 shows a sample tree which also confirms this by looking nearly identical to the sample unpruned DT in figure 4 and in fact learning less since it only predicts for 2 out of the 5 classes (60% error rate by default). Figure 7 shows the error rates against their respective runs of cross validation. The pruned DT also ran in about 10 minutes like the unpruned variant.

The boosting was trained using the complete set of predictor values and produced an abysmal error rate of 82.98%. This is worse than random guessing for a 5-class problem and was in fact very surprising. It was evident from referencing the warning logs that the “multinomial” distribution advertised by the library was broken and had an associated error message printed to that effect. This seems to be the driving factor for the model’s poor performance, and it seems it would need to be resolved prior to realizing coherent results. A 3-dimensional plot of the errors can be seen in figure 8.

The model was fitted using 5-fold cross validation due to excess execution times and converged in just over a day and a half of continuous computing.

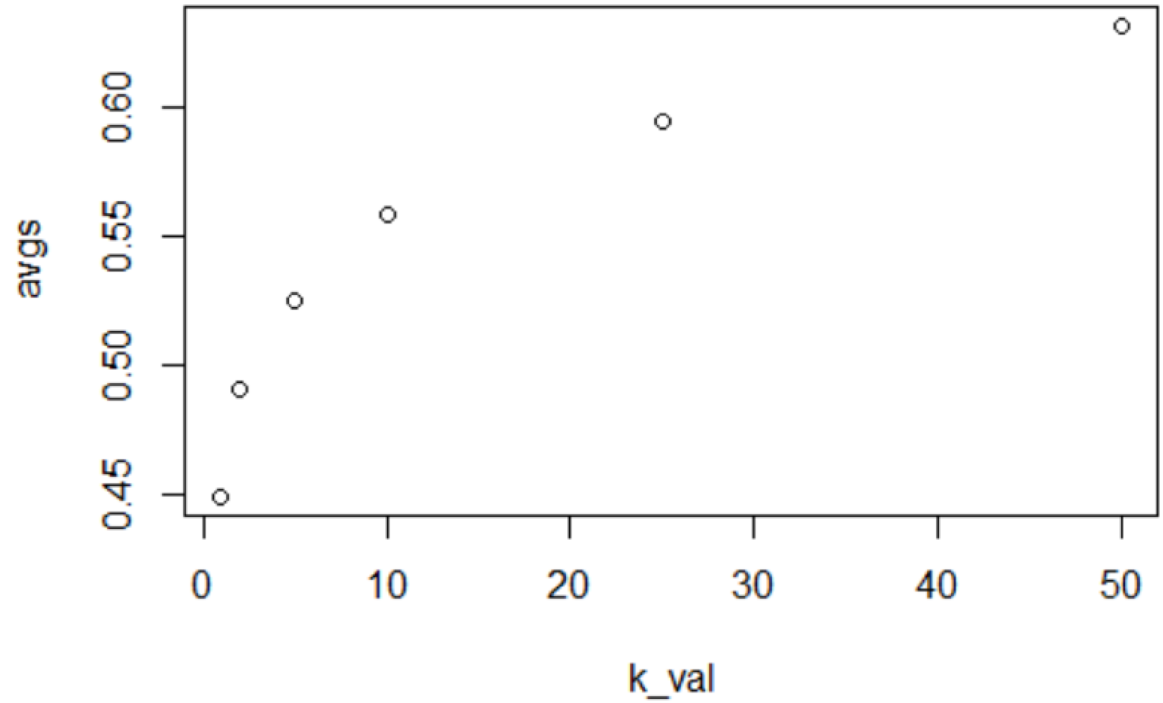

Figure 1. KNN classifier error rates with increasing *k* values

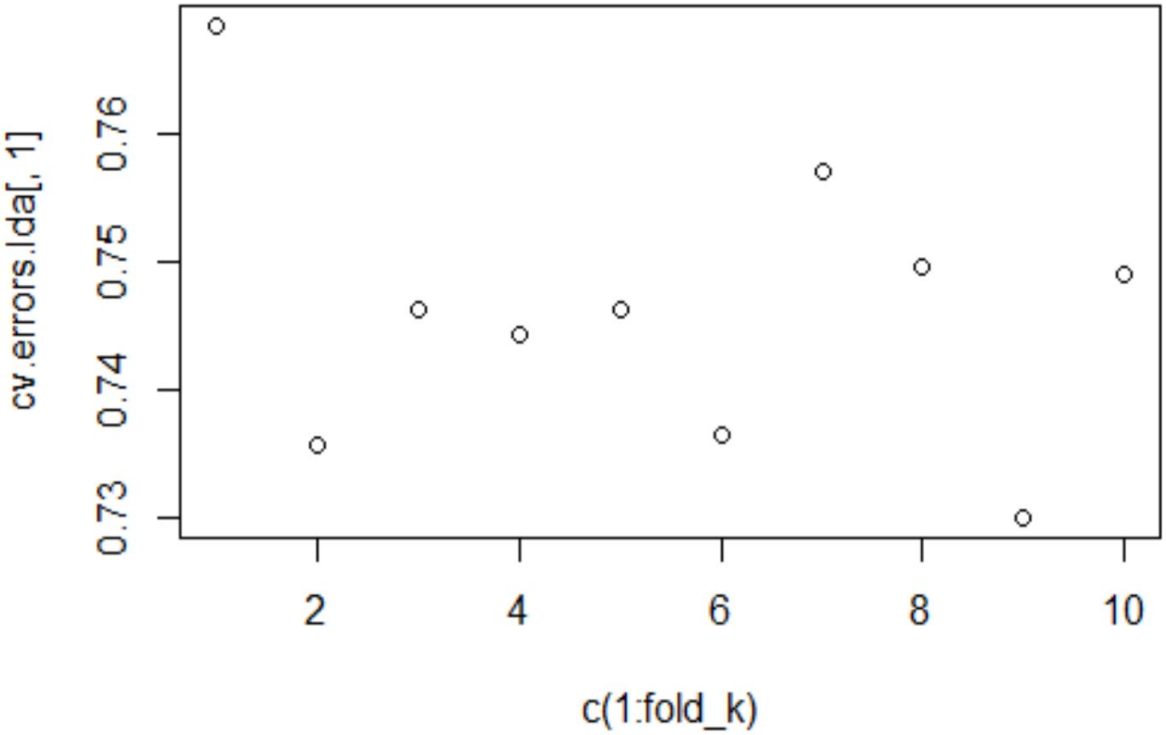

Figure 2. LDA classifier error rates across 10-fold cross validation

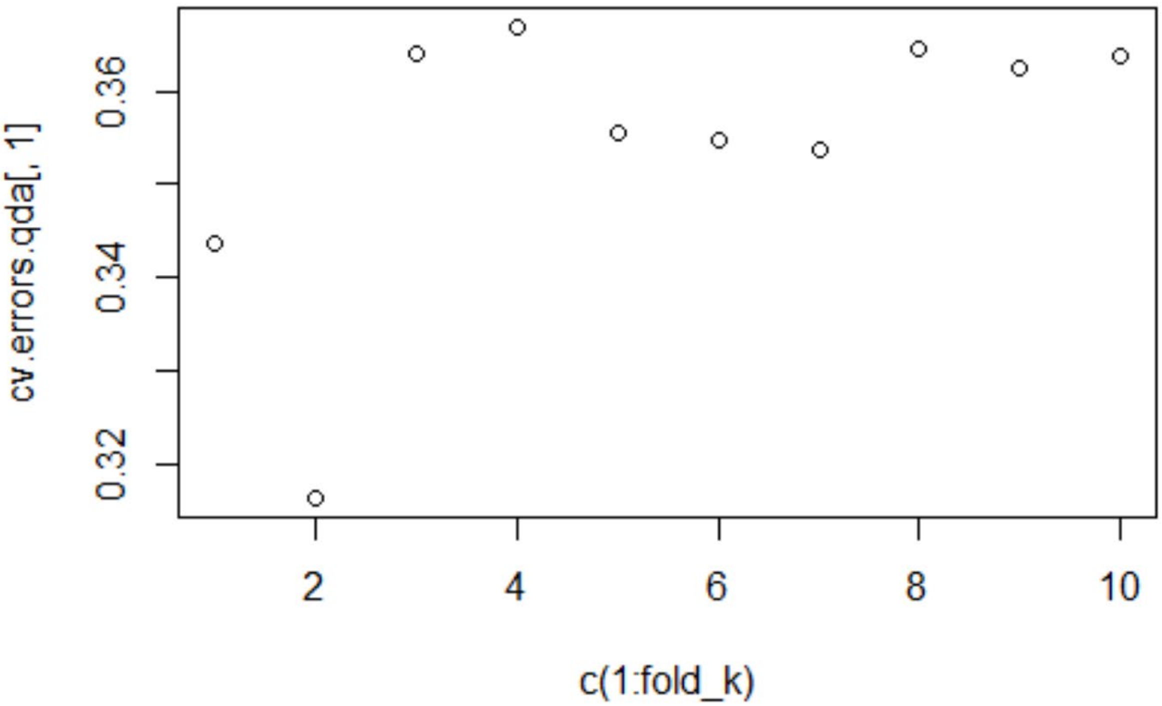

Figure 3. QDA classifier error rates across 10-fold cross validation

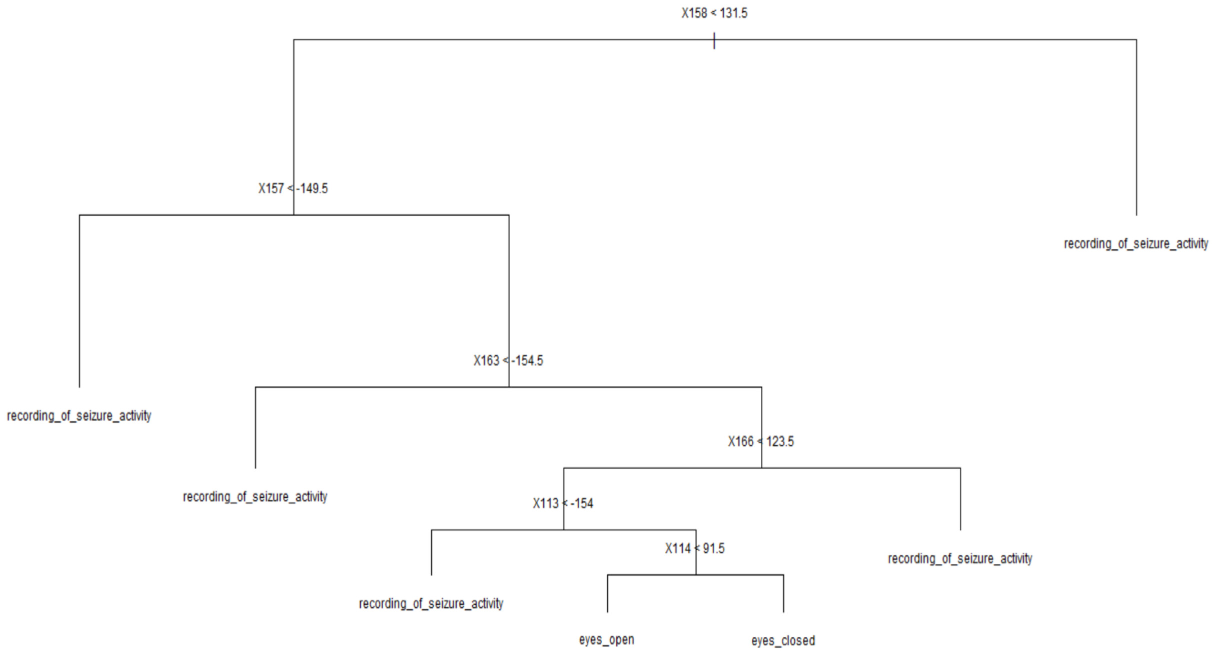
X158 < 131.5
X157 < -149.5
recording_of_seizure_activity
X163 < -154.5
recording_of_seizure_activity
recording_of_seizure_activity
X166 < 123.5
X113 < -154
X114 < 91.5
recording_of_seizure_activity
recording_of_seizure_activity
eyes_open
eyes_closed

Figure 4. A sample unpruned DT showcasing classification of only 3 out of 5 classes

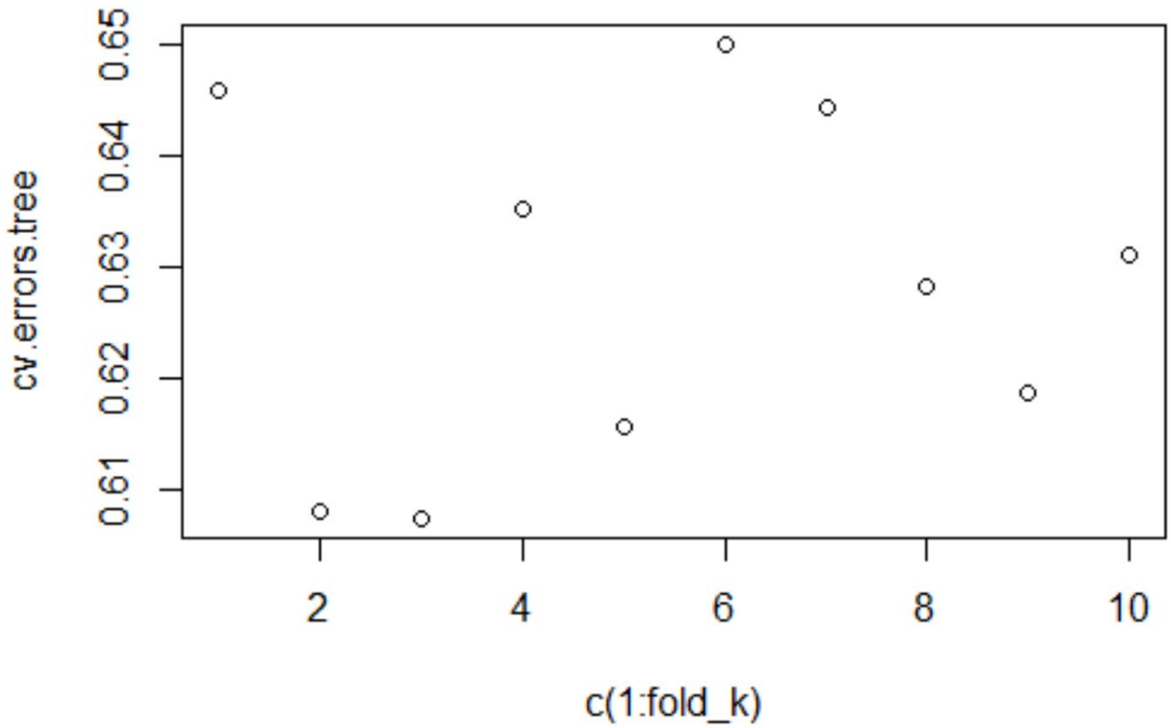

Figure 5. Unpruned DT error rates across 10-fold cross validation

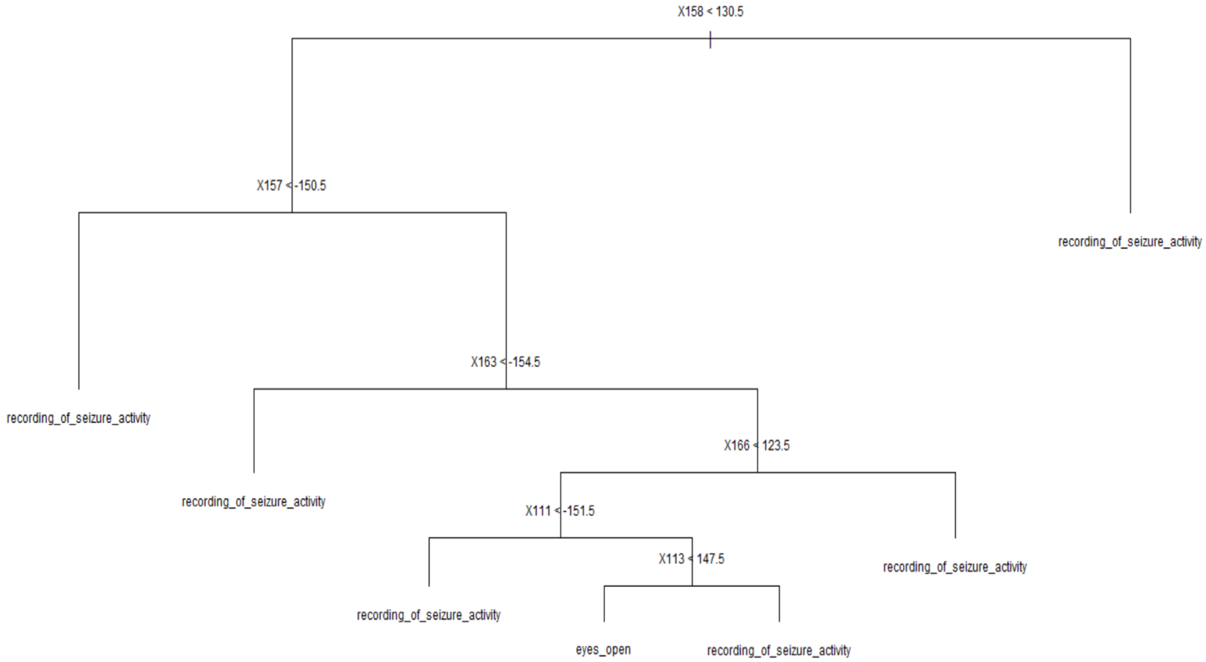
X158 < 130.5
X157 < -150.5
recording_of_seizure_activity
X163 < -154.5
recording_of_seizure_activity
recording_of_seizure_activity
X166 < 123.5
X111 < -151.5
X113 < 147.5
recording_of_seizure_activity
recording_of_seizure_activity
eyes_open
recording_of_seizure_activity

Figure 6. Sample pruned DT showcasing only 2/5 classes predicted

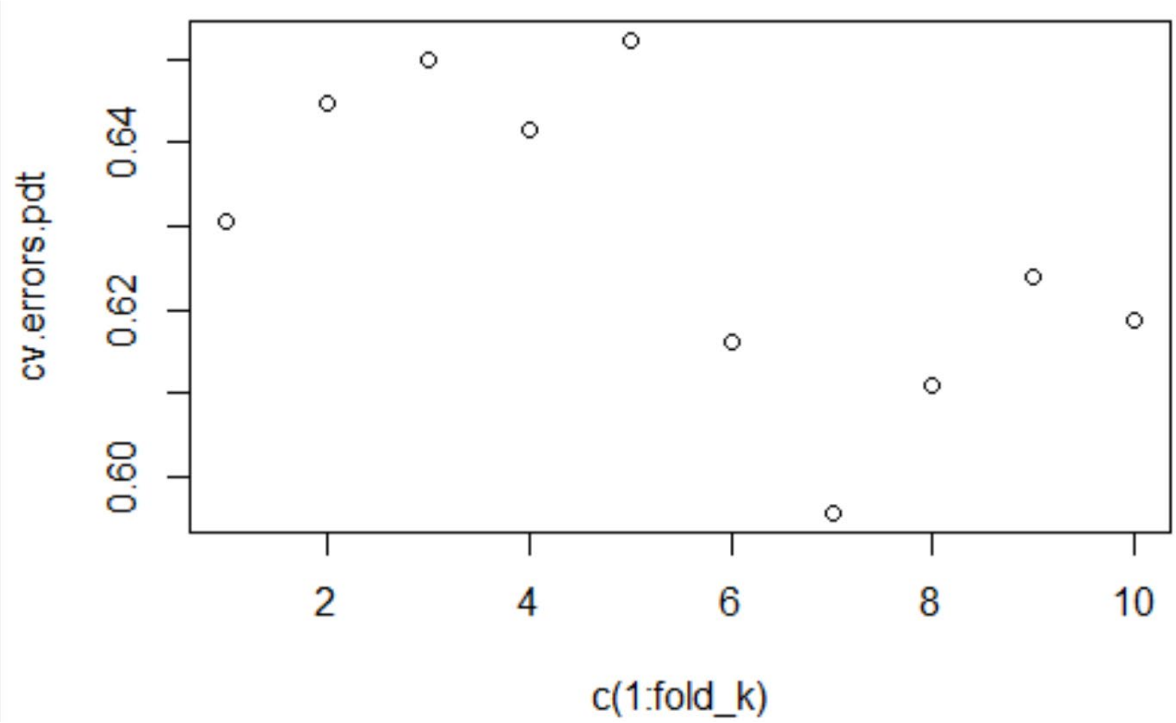

Figure 7. Pruned DT error rates across 10-fold cross validation

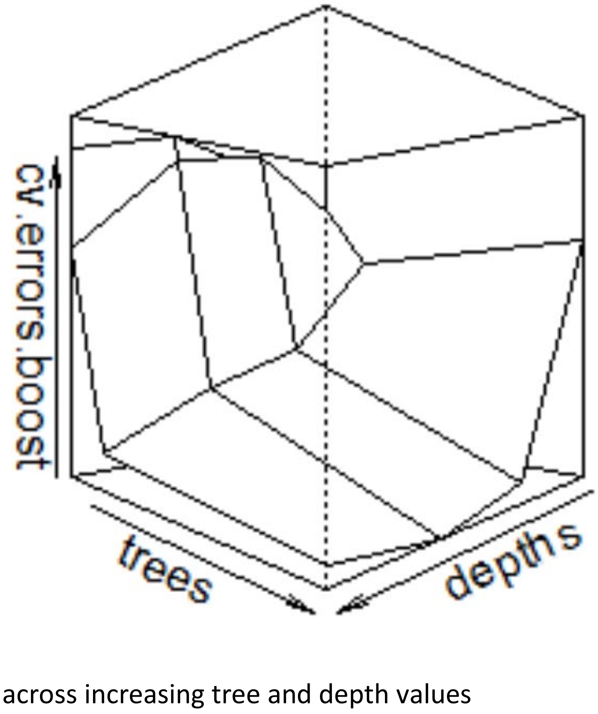

Figure 8. Boosting error rate across increasing tree and depth values

The RF was trained using all the predictor variables as well and boasted an impressive 26.87% average error rate. This was far superior to most of the models and even beat out the QDA. The top performance was achieved with 5,000 trees and 50 predictors considered at each split. This was surprising given the general rule of thumb to use the square root of the number of predictors for classification problems which is generally suggested by literature. The indication here is that with more trees, even if they are all not learning all the data, there are enough of them learning pieces to gather the complete picture. This was tested using 5-fold cross validation because the fitting took an extraordinary amount of time. With 5-fold, the fitting completed in 2 days of non-stop execution. Figure 9 shows a 3-dimensional plot of the error rates here across increasing trees and m-factor considerations.

Bagging was tacked onto the execution of RFs and can be visualized as the rightmost line coming out of the page in figure 9. The best model yielded an average error rate of 27.23% which was nearly identical to that of the RF testing. The addition of more predictor values seems to perhaps overfit the training data and not produce as good of a result on the testing data. This performance was also achieved using 5,000 trees and likewise the time execution was part of the 2 days for the RF.

The SVC and SVM were both trained using the entire set of predictors and yielded a error rates of 71.79% and 57.91%, respectively. These performances were achieved on the SVC using a cost of 0.01 and a gamma of 0.5 and on the SVM using a cost of 5 and a gamma of 0.5. The performance was subpar compared to the RF and the QDA, but the gain the SVM had over the SVC called back to mind the difference between LDA and QDA indicating that the data must inherently be hard to linearly separate and may require non-linear transformation prior to working with it further. The execution was done with 5-fold cross validation and completed in just over a day each.

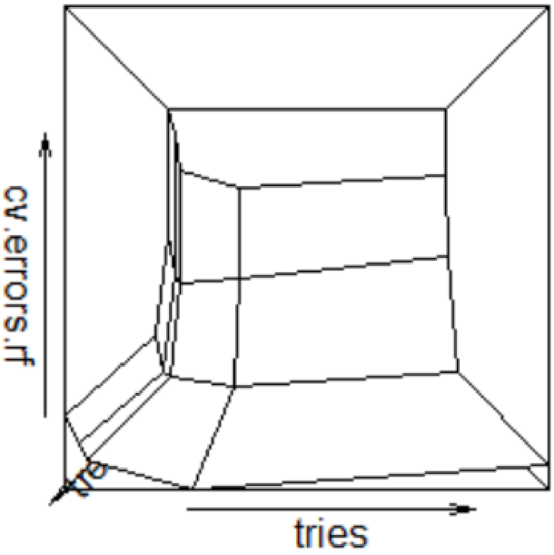

Figure 9. RF error across increasing trees (out of the page) and predictor values (tries)

Conclusions

Overall, the objective was to try and properly classify the general mental state of a patient to act as a proof of concept for larger issues like tracking sleep stage or emotion. Several statistical models were tested on a UCI Machine Learning repository for epileptic seizure data including the KNN classifier, LDA, QDA, DTs, pruned DTs, boosting, RFs, bagging, SVCs, and SVMs. Among these, RFs performed objectively the best, but also took the longest amount of time to fit and test. A case can be made for the efficacy of the QDA which was 10 percent more erroneous but completed training in a manner of minutes as compared to the RF's 2 days. The general indications gathered from the data are that it appears that the predictors are somewhat dependent on one another as is expected of time series data as well as possibly not linearly separable anyway. The more flexible models seemed to outright produce a better result and support this idea. The error rate for boosting was an outlier and most likely caused by a failure within the library function itself.

Some of the direct future work would be to attempt to be more selective with the feature set passed to each model as there are many predictors and not all of them may be relevant. It is hard to say which ones have more value as all are just points in time for EEG data. Regardless, reducing the dimensionality would probably help the training times and bring the longer training models more in line with the simpler models like LDA and QDA. In addition, applying an unsupervised method such as principal component analysis (PCA) may serve to both reduce the dimensionality while also creating independence between the predictors after transformation. Predictors have a higher collinearity the closer in time they are, but further away they become more independent.

For more distant future work, the goal would be to translate some of this ability to classify mental state into other areas which have a high number of patients such as traumatic brain injuries (TBI). This is a large market due to its tangentiality to contact sports such as football or soccer where concussions can occur quite frequently. Moreover, the research can go down the typical path of time-series analysis which is real time inference. Knowing a given point *p* given the future and the past is not realistic, so it would have to predict it given only past data up until the time of prediction.

In any case, the project shows the feasibility of classifying the mental state of patient data given a wide array of models tested across all their tuning parameters. The brute force approach coupled with cross validation made sure to produce the best results for each model given the predictors fed into it. Future work would entail further analysis into the shortcomings of the models to find the true best models. Table 1 summarizes the error rates of all the tested models.

	KNN	LDA	QDA	DT	Pruned DT	Boosting	RF	Bagging	SVC	SVM
Error	0.4494	0.7463	0.3545	0.6285	0.6284	0.8298	**0.2687**	0.2723	0.7179	0.5791

Table 1. Error rates for the best of each tested model – overall best is highlighted in bold

Citations

1. Stafstrom, C. E., & Carmant, L. (2015). Seizures and epilepsy: an overview for neuroscientists. *Cold Spring Harbor perspectives in medicine*, *5*(6), a022426. https://doi.org/10.1101/cshperspect.a022426
2. Andrzejak RG, Lehnertz K, Rieke C, Mormann F, David P, Elger CE (2001) Indications of nonlinear deterministic and finite dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state, Phys. Rev. E, 64, 061907
3. Shoeb, A. H., & Guttag, J. V. (2010). Application of machine learning to epileptic seizure detection. In Proceedings of the 27th International Conference on Machine Learning (ICML-10) (pp. 975-982).
4. Shoeb, A. H. (2009). Application of machine learning to epileptic seizure onset detection and treatment (Doctoral dissertation, Massachusetts Institute of Technology).
5. Subasi, A., Kevric, J., & Canbaz, M. A. (2019). Epileptic seizure detection using hybrid machine learning methods. Neural Computing and Applications, 31(1), 317-325.
6. Müller, K. R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., & Blankertz, B. (2008). Machine learning for real-time single-trial EEG-analysis: from brain–computer interfacing to mental state monitoring. Journal of neuroscience methods, 167(1), 82-90.