# Manoosh Samiei

# EEG Classification Report

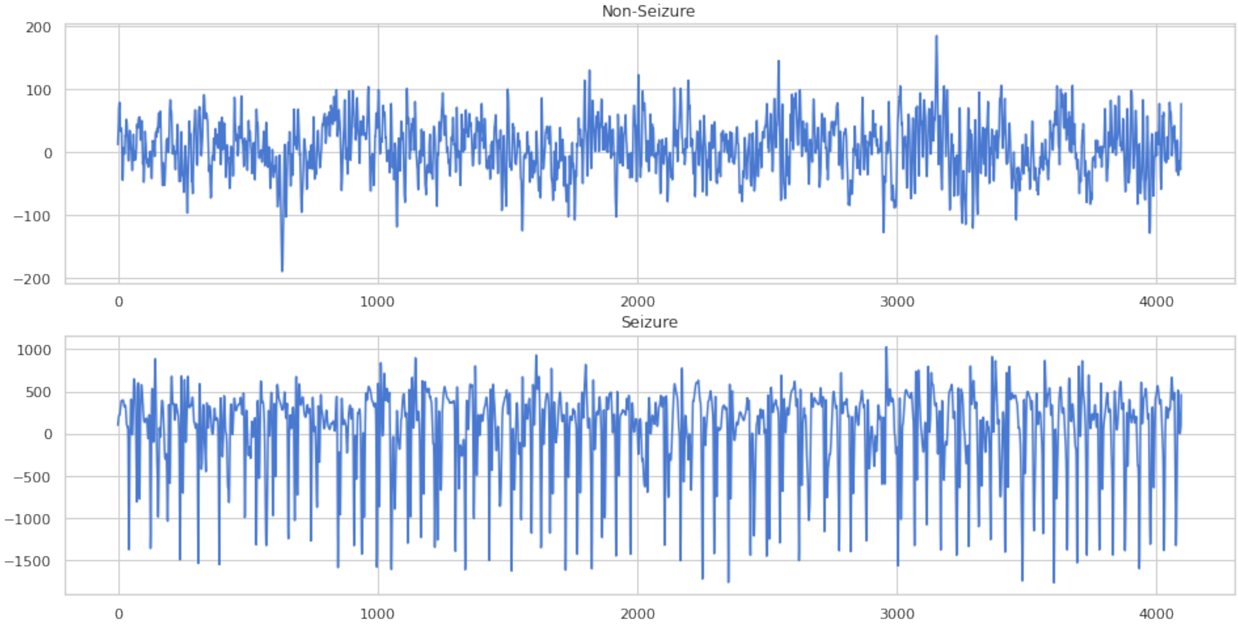
## Introduction:

Epilepsy is one of the world's most common neurological diseases. Early prediction of the incoming seizures has a great influence on epileptic patients' life. [5] If the occurrence of seizure could be predicted well in advance, it could be prevented through medication or proper actions. Electroencephalogram (EEG) is generally used to detect epilepsy as EEG is capable of capturing the electrical activity of brain. In literature, many machine learning techniques were used to extract features from EEG recordings and predict the occurrence seizures. [6]

## Dataset:

The data, published on Bonn University's Epileptology department website, presents Electroencephalogram (EEG) recording of 500 individuals. For each individual, brain activity was recorded for a duration of 23.5 seconds; these recordings are represented by 4096 evenly spaced, consecutive data points (i.e every 0.0057 seconds). There are five sets (A, B, C, D, E) available in this dataset. Set A contains the EEG recording of non-epileptic awake patients with their eyes open. Set B contains EEG recording of non-epileptic awake patients with their eyes closed. Set C contains EEG recording of epileptic patients during seizure free period using electrodes implanted in the brain epileptogenic zone. Set D contains EEG recording of epileptic patients during seizure free period from the hippocampal formation of the opposite hemisphere of the brain from C. Set E contains EEG recording of patients experiencing an active epileptic stroke. [7]

In this project I attempt to classify EEG signals into seizure and non-seizure categories using Machine Learning algorithms. I use sets A, B, C, and D as non-seizure data and set E as seizure data. Below are two sample signals from non-seizure and seizure classes.



It is clear from the above images that the change in electrical activity of the brain during seizures can be detected using EEG signals. I implemented two methods to classify EEG signals into seizure and non-seizure classes. The first method works with the original signal values while the second method decomposes signal into sub-bands using discrete wavelet transformation. For each of these methods I use different machine learning algorithms to achieve the best results. In the following, I will elaborate on my implemented methods.

## Method 1 (EEG_Classifaction_Without_Wavelet_Decomposition.ipynb):

At first, I applied simple machine learning models provided by scikit learn library to our dataset. In particular, I tested Nearest Neighbors, Linear SVM, RBF SVM, Gaussian Process, Decision Tree, Random Forest, Multi-layer Perceptron (fully connected Neural Net), AdaBoost, Naive Bayes, and Quadratic Discriminant Analysis. To prepare the dataset for these algorithms, I was inspired by this GitHub repository, which extract 6 features from the data namely: DFA (Detrended Fluctuation Analysis), HFD (Hjorth Fractal Dimension), SVD Entropy, Spectral Entropy, Fisher Information, and PFD (Petrosian Fractal Dimension), that are included in pyeeg library. For more information on each of these functions please refer to pyeeg documentation at this link.

Using these 6 features, machine learning models' accuracy are as follows:

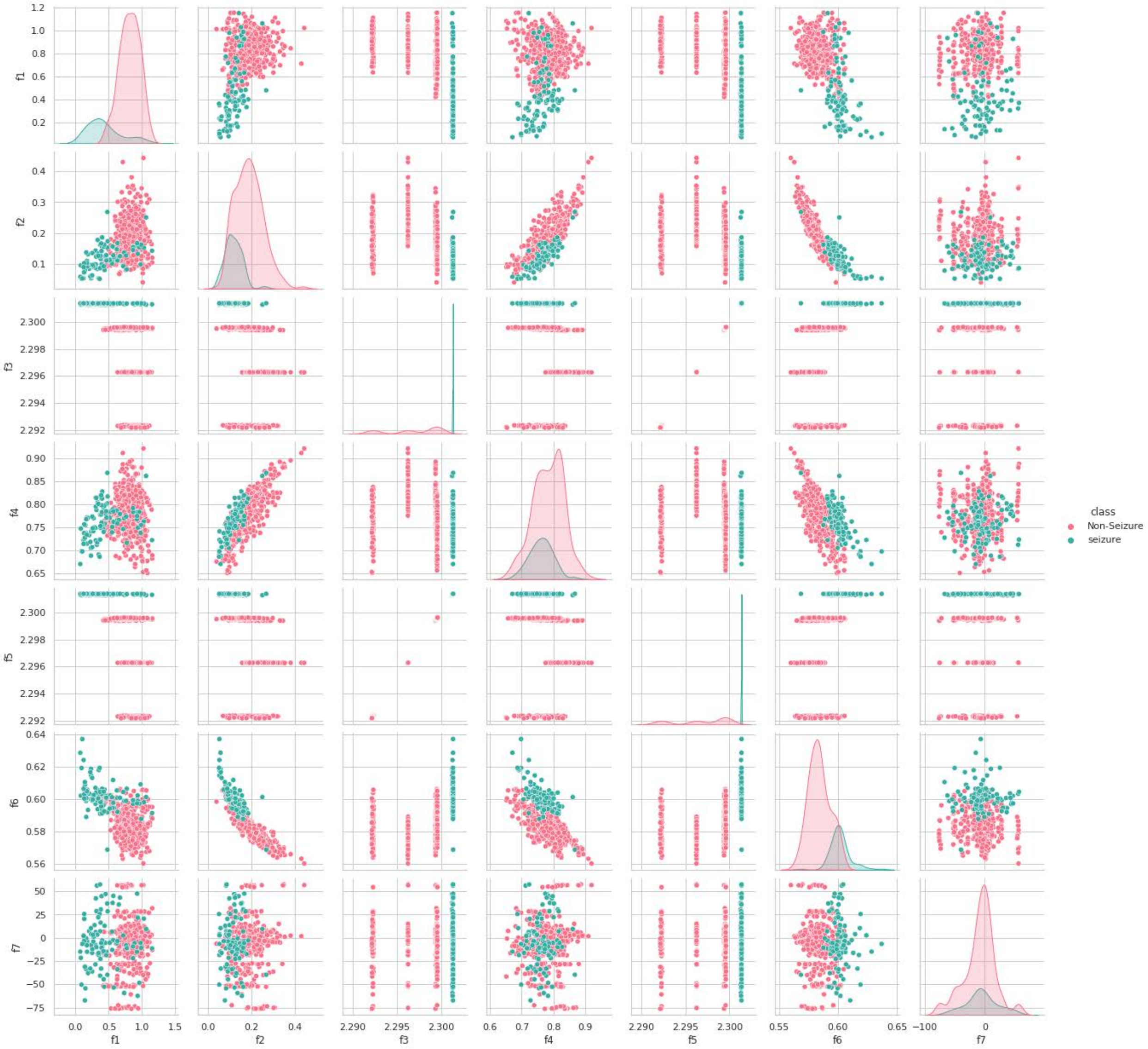
```
[[0.9, 'Nearest Neighbors'],
 [0.76, 'Linear SVM'],
 [0.87, 'RBF SVM'],
 [0.98, 'Gaussian Process'],
 [1.0, 'Decision Tree'],
 [1.0, 'Random Forest'],
 [0.86, 'Neural Net'],
 [1.0, 'AdaBoost'],
 [0.99, 'Naive Bayes'],
 [0.73, 'QDA']]
```

I added mean of each signals values as the 7th feature, and noticed a significant improvement in the accuracies across all models:

```
[[0.98, 'Nearest Neighbors'],
 [0.95, 'Linear SVM'],
 [0.96, 'RBF SVM'],
 [1.0, 'Gaussian Process'],
 [1.0, 'Decision Tree'],
 [1.0, 'Random Forest'],
 [0.98, 'Neural Net'],
 [1.0, 'AdaBoost'],
 [0.99, 'Naive Bayes'],
 [0.93, 'QDA']]
```

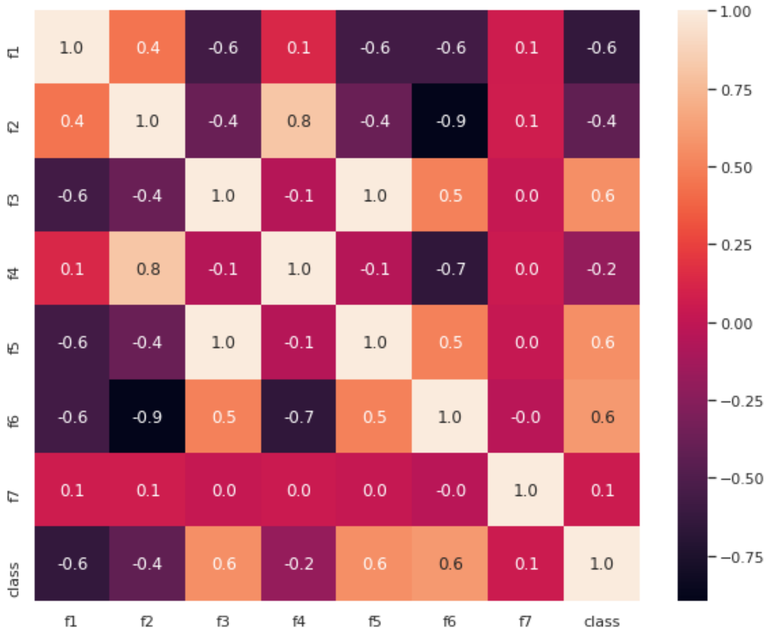
Good features are key for accurate classification. A Good feature combination is a combination that separates data of different class labels. (references: https://code-maven.com/predicting-wine-cultivar &
https://towardsdatascience.com/correlation-is-simple-with-seaborn-and-pandas-28c28e92701e )

We can investigate this property using seaborn pairplot:



We can see that our features have little overlap between the two classes. In most cases there is a clear distinction between the distribution of the two classes. However, there are some cases that the datapoints of the two classes have strong overlap and are indistinguishable. For instance, features 4 and 7 (corresponding to spectral entropy and mean), and features 1 and 7 (DFA and mean) cannot distinguish the two classes properly.

Furthermore, we can use the Pearson correlation to compute pairwise correlation of features in our data. It is worth to emphasize that the Pearson correlation is only good for measuring linear correlation.



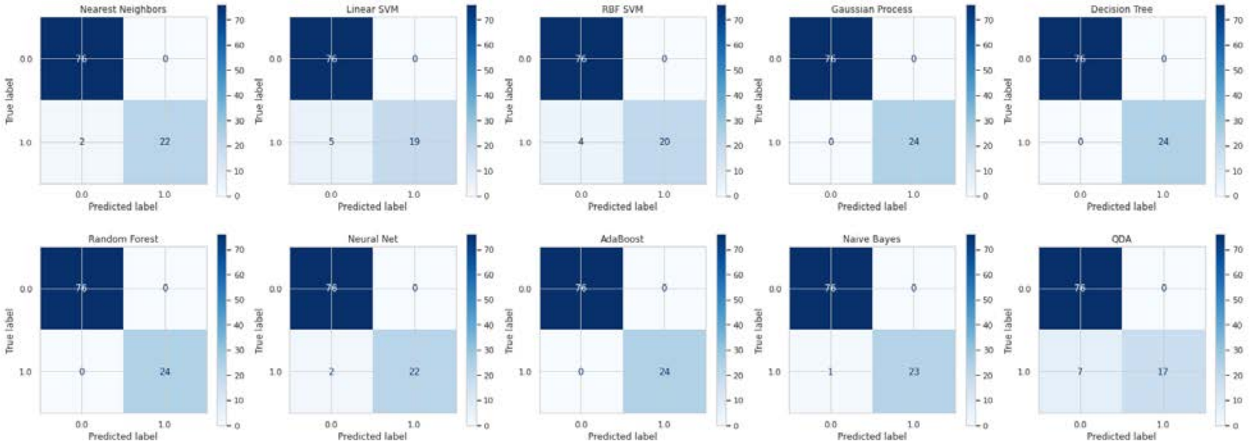
| | f1 | f2 | f3 | f4 | f5 | f6 | f7 | class |
|---|---|---|---|---|---|---|---|---|
| f1 | 1.0 | 0.4 | -0.6 | 0.1 | -0.6 | -0.6 | 0.1 | -0.6 |
| f2 | 0.4 | 1.0 | -0.4 | 0.8 | -0.4 | -0.9 | 0.1 | -0.4 |
| f3 | -0.6 | -0.4 | 1.0 | -0.1 | 1.0 | 0.5 | 0.0 | 0.6 |
| f4 | 0.1 | 0.8 | -0.1 | 1.0 | -0.1 | -0.7 | 0.0 | -0.2 |
| f5 | -0.6 | -0.4 | 1.0 | -0.1 | 1.0 | 0.5 | 0.0 | 0.6 |
| f6 | -0.6 | -0.9 | 0.5 | -0.7 | 0.5 | 1.0 | -0.0 | 0.6 |
| f7 | 0.1 | 0.1 | 0.0 | 0.0 | 0.0 | -0.0 | 1.0 | 0.1 |
| class | -0.6 | -0.4 | 0.6 | -0.2 | 0.6 | 0.6 | 0.1 | 1.0 |

We can see that features 2 and 6 (HFD and PFD) are highly correlated (linearly) with a correlation factor of -0.9. Also features 2 and 4 (HFD and Spectral Entropy) are correlated with a factor of 0.8. Therefore, it is possible to remove feature 2 from feature space, without much effect on the accuracy.

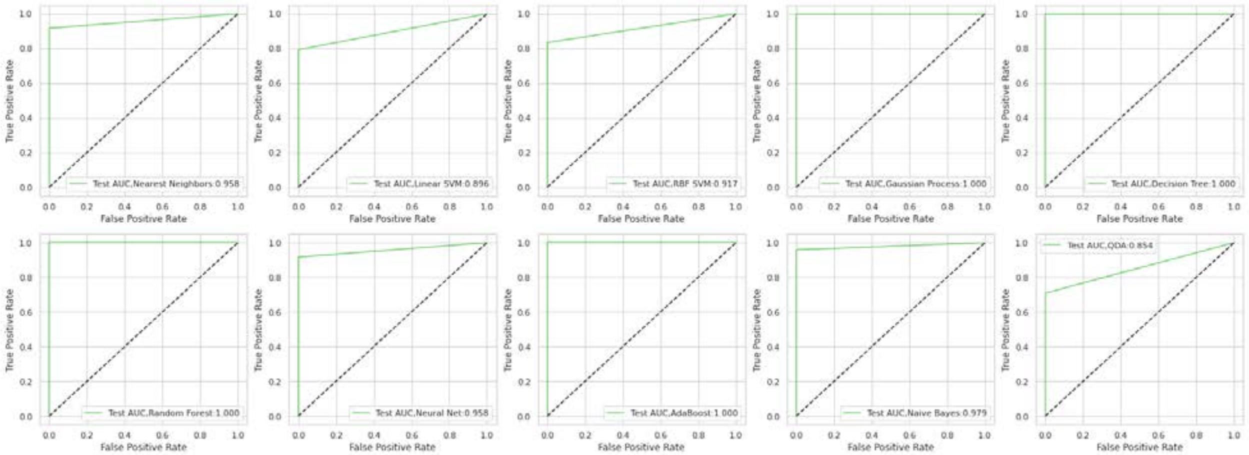
In practice, I realized that removing feature 2 from feature space will reduce the performance of Quadratic Discriminant Analysis by 10 percent while increasing RBF SVM by 2 percent.

Another method I used to check if the features have significant difference (separability) between the two classes is **Kruskal-Wallis Test**. In this test, null hypothesis assumes that the samples (groups) are from identical populations. While alternative hypothesis assumes that at least one of the samples (groups) comes from a different population than the others. To realize if null hypothesis is rejected or not we can use p-value. A very small p-value means that an observed outcome would be very unlikely under the null hypothesis. I set a threshold of 0.05 for p-value; if p-value is smaller than this threshold the null hypothesis is rejected, and significant differences exist between groups. Hence the features are wisely chosen and can distinguish between the different classes of the dataset. The 7 features used in this method reject the null hypothesis.

Also, we should note that our dataset is unbalanced, as it has 400 samples for non-seizure class (sets A, B, C, and D), and 100 samples for seizure class (set E). Even if a model always predicts label 'non-seizure', it can achieve 80% accuracy as 400 out of 500 data samples contain a non-seizure signal. Thus, we should not solely rely on accuracy metric to assess the models' performance. I plot the ROC curve and confusion matrix to obtain the number of false negatives and false positives predicted by the models.



The diagonals of the above confusion matrices show the correct classifications. The top right corner of each confusion matrix shows false positives i.e., detecting seizure while the signal is actually non-seizure, and the bottom left corner is false negative i.e., detecting non-seizure while the signal is actually seizure.



The ROC curve shows the trade-off between sensitivity (true positive rate $= \frac{\#\text{ true positives}}{\#\text{true positives} + \#\text{ false negatives}}$) and specificity (1 - false positive rate $= 1 - \frac{\#\text{ false positive}}{\#\text{ false positive} + \#\text{ true negative}}$ ).

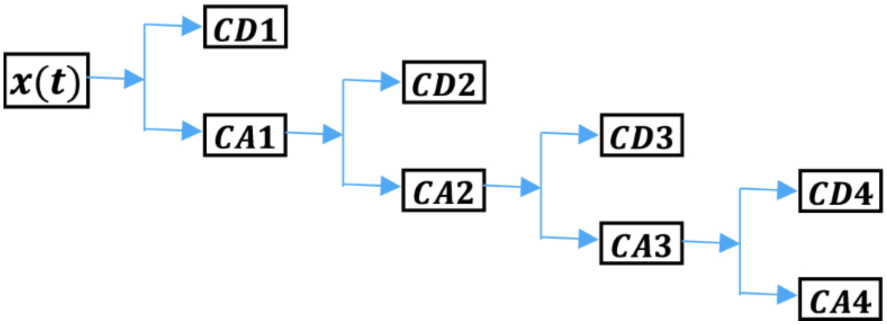
Classifiers that give curves closer to the top-left corner indicate a better performance. The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the predictions.

It can be seen from the above results that the best models are Gaussian Process, Decision Tree, Random Forest, and AdaBoost with 100% accuracy (zero false positive and false negative predictions) and then Naive Bayes with 99% accuracy (one false negative prediction).

## Method 2 (EEG_Classifaction_DWT.ipynb):

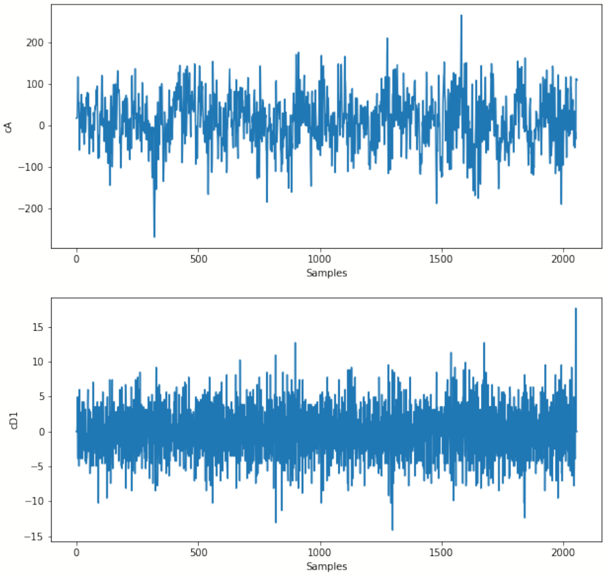
Inspired by papers [1], [2], and [3], I decided to decompose EEG signals using discrete wavelet transform before their classification. The motivation behind this decomposition is to compress the time-varying signal, which comprises many data points, into few parameters that represents the signal. [2] In pywt library there are several wavelet families available such as Haar (haar), Daubechies (db), Symlets (sym), Coiflets (coif), Biorthogonal (bior), Gaussian wavelets (gaus), Mexican hat (mexh), and Morlet (morl). I will test two wavelet families Daubechies (db) and Biorthogonal (bior) inspired by [3] and [4] respectively.

Biorthogonal family has 15 wavelets termed as 'bior1.1', 'bior1.3', 'bior1.5', 'bior2.2', 'bior2.4', 'bior2.6', 'bior2.8', 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7', 'bior3.9', 'bior4.4', 'bior5.5', and 'bior6.8'. Also, while decomposing the signal using each of these wavelets, we should define the level of decomposition. Below figure shows the 4th level signal decomposition, as the frequency of the brain signals is divided into five sub-bands. In the four levels, CD1-CD4 represents the detailed coefficients and CA4 denotes the approximate coefficient, the decomposed signal can be reconstructed by the summation of detailed coefficients and the last approximate coefficient. [3]



*Wavelet decomposition diagram. Image taken from [3].*

After some experiments, I found that **bior2.8** with decomposition **level 1** gives the highest accuracy of 99 percent by one of the classifiers. Below figure shows the two sub-bands that one sample signal is decomposed to.



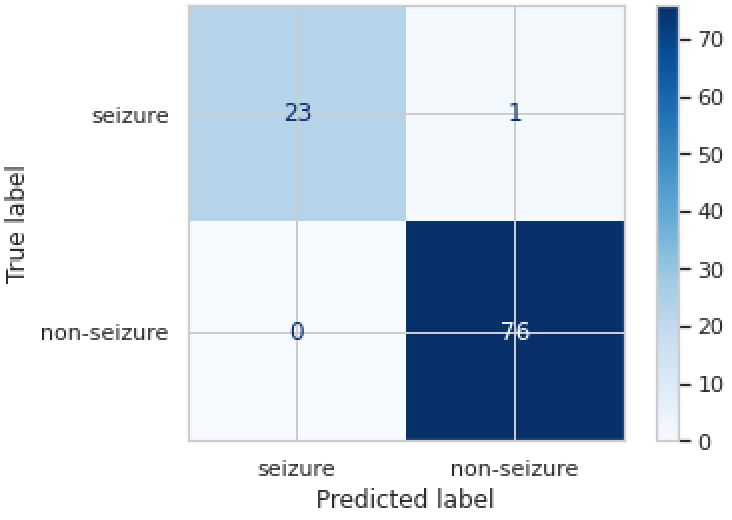
After decomposition of signal, we need to extract features from the sub-bands. In a level 1 decomposition, we have two decomposition coefficients namely cA and cD1. I first extracted 7 features (that I also used in method 1): DFA (Detrended Fluctuation Analysis), HFD (Hjorth

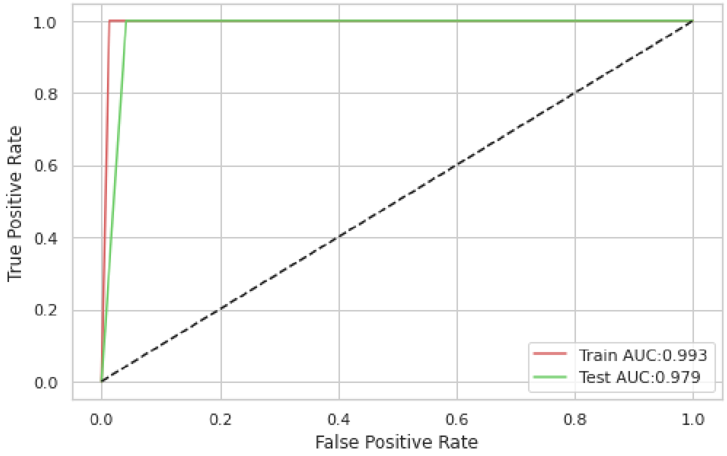
Fractal Dimension), SVD Entropy, Spectral Entropy, Fisher Information, and PFD (Petrosian Fractal Dimension), and mean for each of these coefficients. Therefore, in total we will have 14 features. (7 for each coefficient)

I tested the previous machine learning models used in method 1 on wavelet decomposed data, with same 7 feature extractions. Interestingly, the accuracy of the models decrease compared to the undecomposed data (in previous method). The accuracy of models after wavelet decomposition are as follows:

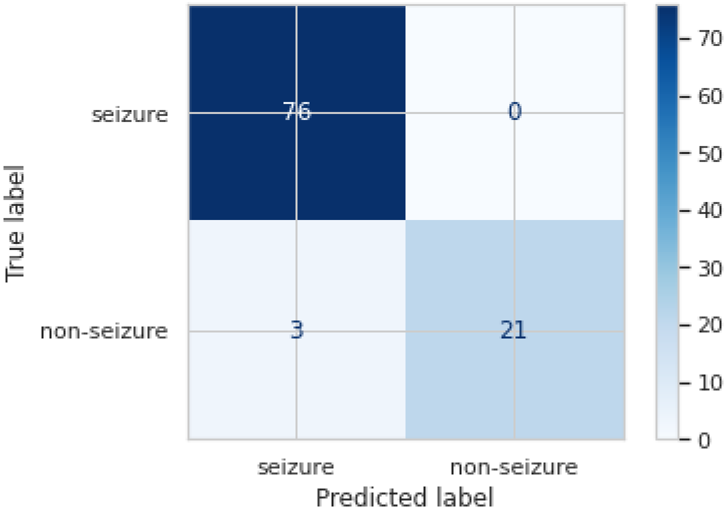
```
[[0.93, 'Nearest Neighbors'],
 [0.95, 'Linear SVM'],
 [0.76, 'RBF SVM'],
 [0.98, 'Gaussian Process'],
 [0.93, 'Decision Tree'],
 [0.91, 'Random Forest'],
 [0.96, 'Multi-layer Perceptron'],
 [0.95, 'AdaBoost'],
 [0.96, 'Naive Bayes'],
 [0.85, 'QDA']]
```

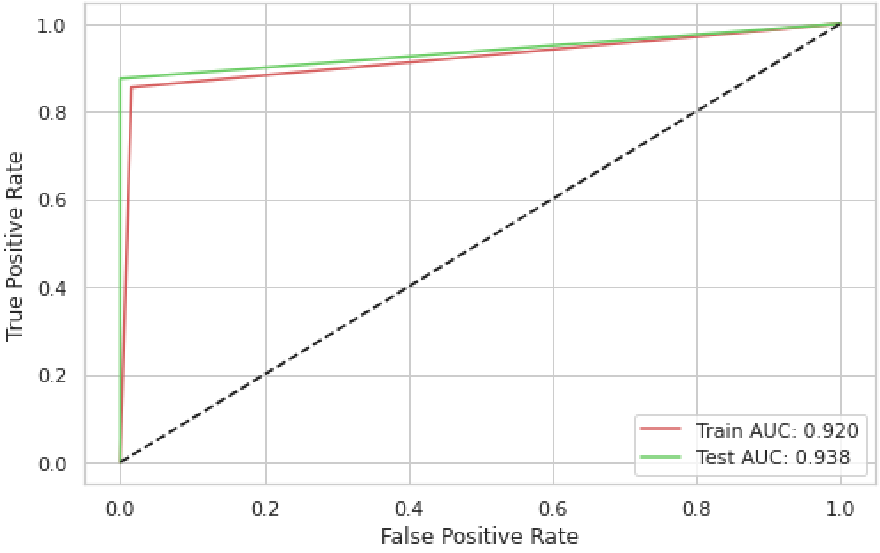
To achieve better performance, I tested the performance of two neural network architectures on the decomposed signals. Feature extraction is an important part of data preprocessing while working with neural nets. Running Kruskal-Wallis Test shows that there is some overlap between these 7 feature's distributions on the decomposed data. After some experiments I found that using three features SVD Entropy, HFD, and PFD leads to a better accuracy. The first model I use for classification of decomposed signals is a neural network with ten hidden neurons and a single hidden layer. The ANN consists of a linear activation function and a hyperbolic tangent function in its output and hidden layer respectively. The neural network training algorithm used is Levenberg-Marquardt backpropagation algorithm. The output layer consists of two nodes defining each class. This classifier achieves an accuracy of 0.99 and is built with pyrenn library. If we assign time delays between the layers we can allow for recurrent connections and convert the network from feed forward to recurrent architecture. However, in practice I realized that recurrent connections decrease the performance, and therefore used the feedforward version. The confusion matrix and ROC curve of this classifier are shown below.





The second classifier has the same architecture as the previous one (two fully connected layers, one hidden layer with 10 hidden units) but instead of hyperbolic tangent it has sigmoid activation function in the output layer and is trained with ADAM backpropagation Algorithm. This network is built with TensorFlow framework and achieves an accuracy of 0.97.





I tested the previous machine learning models used in method 1, on wavelet decomposed data, with 3 features: SVD Entropy, HFD, and PFD. The accuracy of models in this case are as follows:

```
[[0.94, 'Nearest Neighbors'],
 [0.94, 'Linear SVM'],
 [0.9, 'RBF SVM'],
 [0.97, 'Gaussian Process'],
 [0.95, 'Decision Tree'],
 [0.91, 'Random Forest'],
 [0.96, 'Multi-layer Perceptron'],
 [0.96, 'AdaBoost'],
 [0.87, 'Naive Bayes'],
 [0.96, 'QDA']]
```

The classifiers still do worse than the undecomposed method. Compared to the case with 7 features, we cannot make a general claim as some classifiers have improved and some have worsened.

Therefore, among classifiers of method 2, The first neural network with Levenberg-Marquardt backpropagation algorithm achieved the highest (validation) accuracy of 99%.

I also tested the classifiers with another wavelet decomposition based on Daubechies wavelet 'db10'. However, the classification accuracy dropped, and I achieved better accuracy with 'bior2.8' decomposition.

In summary between the two methods, I achieved the highest accuracy of 100% with classifiers: Gaussian Process, Decision Tree, Random Forest, and AdaBoost by extracting 7 features: DFA (Detrended Fluctuation Analysis), HFD (Hjorth Fractal Dimension), SVD Entropy, Spectral Entropy, Fisher Information, and PFD (Petrosian Fractal Dimension), and mean; from the original signals without wavelet decomposition.

## References:

[1] R. Panda, P. S. Khobragade, P. D. Jambhule, S. N. Jengthe, P. R. Pal and T. K. Gandhi, "Classification of EEG signal using wavelet transform and support vector machine for epileptic seizure diction," *2010 International Conference on Systems in Medicine and Biology*, Kharagpur, India, 2010, pp. 405-408, doi: 10.1109/ICSMB.2010.5735413.

[2] Al-Fahoum AS, Al-Fraihat AA. Methods of EEG signal features extraction using linear analysis in frequency and time-frequency domains. *ISRN Neurosci*. 2014;2014:730218. Published 2014 Feb 13. doi:10.1155/2014/730218

[3] Anila Glory H., Vigneswaran C., Shankar Sriram V.S. (2020) Identification of Suitable Basis Wavelet Function for Epileptic Seizure Detection Using EEG Signals. In: Luhach A., Kosa J., Poonia R., Gao XZ., Singh D. (eds) First International Conference on Sustainable Technologies for Computational Intelligence. Advances in Intelligent Systems and Computing, vol 1045. Springer, Singapore. https://doi.org/10.1007/978-981-15-0029-9_48

[4] Dinesh Bhati, Manish Sharma, Ram Bilas Pachori, Vikram M. Gadre, Time–frequency localized three-band biorthogonal wavelet filter bank using semidefinite relaxation and nonlinear least squares with epileptic seizure EEG signal classification,Digital Signal Processing,Volume 62,2017,Pages 259-273,ISSN 1051-2004,https://doi.org/10.1016/j.dsp.2016.12.004.

[5] H. Daoud and M. A. Bayoumi, "Efficient Epileptic Seizure Prediction Based on Deep Learning," in *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 5, pp. 804-813, Oct. 2019, doi: 10.1109/TBCAS.2019.2929053.

[6] Patel V., Buch S., Ganatra A. (2020) A Review on EEG Based Epileptic Seizure Prediction Using Machine Learning Techniques. In: Pandian A., Ntalianis K., Palanisamy R. (eds) Intelligent Computing, Information and Control Systems. ICICCS 2019. Advances in Intelligent Systems and Computing, vol 1039. Springer, Cham. https://doi.org/10.1007/978-3-030-30465-2_43

[7] Andrzejak RG, Lehnertz K, Mormann F, Rieke C, David P, Elger CE. 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 Stat Nonlin Soft Matter Phys. 2001 Dec;64(6 Pt 1):061907. doi: 10.1103/PhysRevE.64.061907. Epub 2001 Nov 20. PMID: 11736210.