Pattern Recognition
Final Project: Monsoon 2020

EYE STATE PREDICTION USING EEG

Group 6 Team members:
Manohar K S20180020215
Rohith A S20180020238
Madhan G S20180020210

1 Introduction

Electroencephalography (EEG) is used to measure the brain activity which can be used as inputs for applications requiring human actions. For example, brain stimuli have been used for computer games [1], for handicapped persons etc. So these require accurate detection of the stimuli to avoid false predictions.
Many studies has been done to find differences between the two eye states, whether eyes are open or closed. [2] came to the conclusion that the "greatest difference between two states was that the power in the eye closed state was much higher than that in the eye open state". [3] investigated how to track eye blinking based on EEG input.

2 Problem Description

The problem is to find whether eye state can be predicted using EEG signals collected from the brain through various sensors.

3 Data extraction

The total duration of the measurement is 117 seconds. The duration of the measurement was 117 seconds. Both open or partially open eyes were categorized as open, only completely closed eyes were categorized as closed. Sensors could

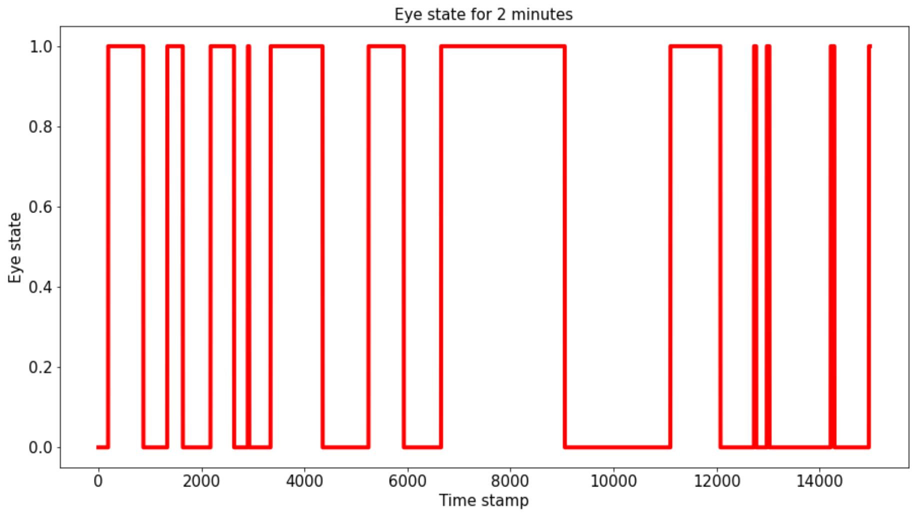


Figure 1: Distribution of eye states during the two minutes.0 represents eye open and 1 eye closed.

be split into two groups. In the first group, the maximum increases when eyes open while, in the other group, the minimum decreases in the same event. Most sensors of the first group happen to be located on the right hemisphere while most of the second group are on the left hemisphere of the brain, as displayed in Figure 2.

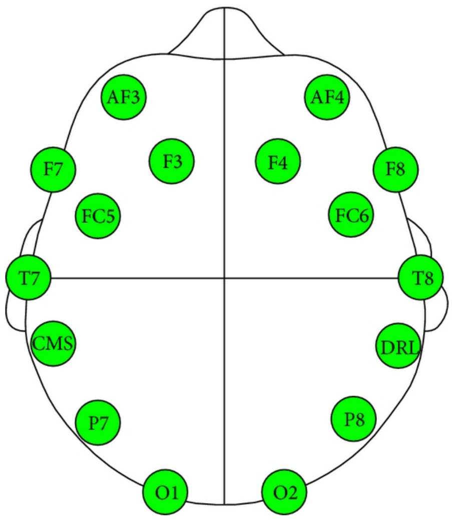


Figure 2: The sensor positions placed on the brain.

3.1 Data cleaning

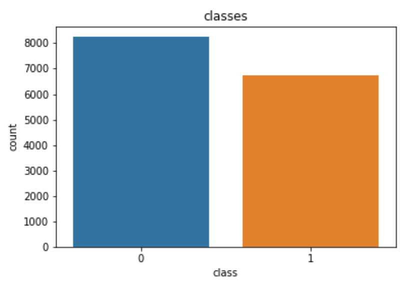
- First of all we have assigned labels to the dataset, which helps in understanding the dataset clearly.
- Then the starting step to do in any data mining is finding the null or duplicate values. So we have checked the presence of nulls and duplicate instances, there are no such values in the dataset.
- There are total 14980 instances, 14 features and 1 column representing the class.
- Now we have deleted the outliers presenting in the dataset using three processes.

3.2 Tests

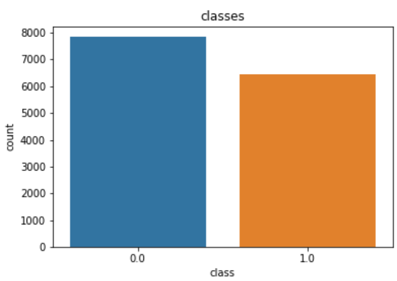
- Z-test: For checking whether the data is having normal distribution or not, we performed Z-test using QQ-plot. Some of the features have failed z-test as shown in Fig 4.
- So we found z-score to each instance and the data having score less than 3 is considered as required data and the following analysis is applied.
- After removing the unwanted data there are 14892 instances.
- Using IQR: In this we have found the quartiles (Q1,Q3) for each feature and also the inter quartile range(1QR = Q3-Q1). Now the data in between

the range ((Q1 - 3*IQR)),(Q3 + 3*IQR)) is considered as necessary data and following analysis is applied.

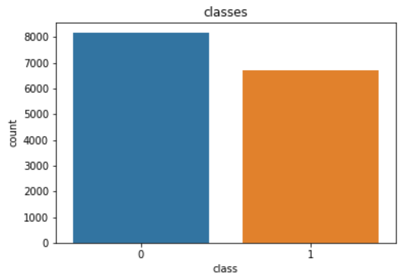
- After removing these instances there are 13865 data instances.
- Outlier: For this we have choose 4 times the standard deviation of the feature as the threshold. So whatever the data lies in (mean - std*4, mean + std*4) range, it will be considered as actual data.
- Now after removing the outliers the instances remained are 14304. So a total of 676 instances were removed.



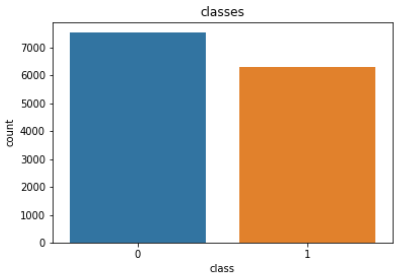
(a) Actual data



(b) After removing using mean and variance

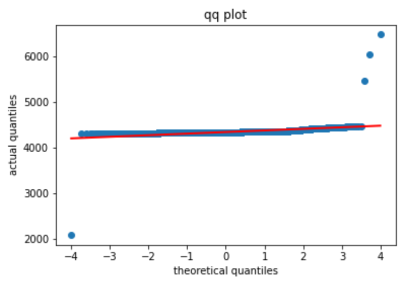


(c) After removing using z-test

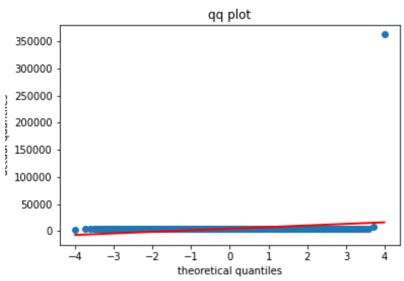


(d) After removing using IQR

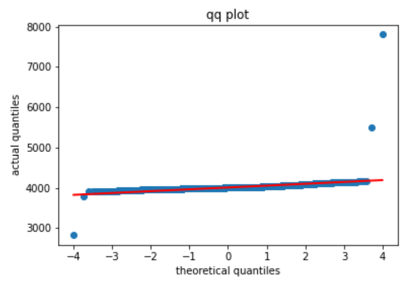
Figure 3: Data split of the classes



(a) feature "T7" q-q plot



(b) feature "F7" q-q plot



(c) feature "P7" q-q plot

Figure 4: q-q plots

3.3 Sensor data visualisation

- After removing the outliers, the senors data can be clearly seen. All these 14 sensor's data is used for our analysis. As mentioned before these are the different sensor values recorder over 117 seconds of time.

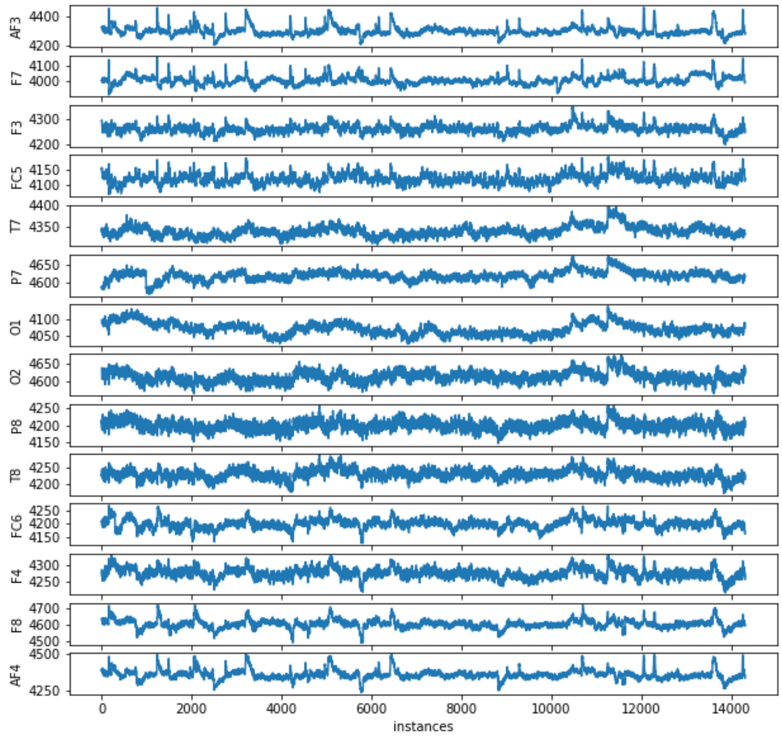
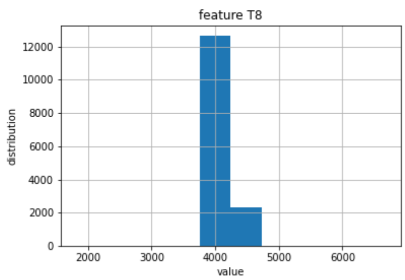
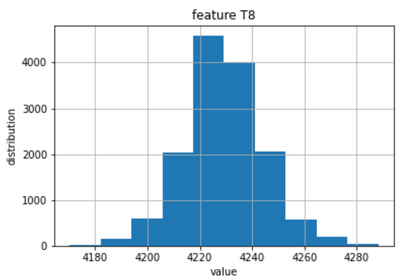


Figure 5: The sensor data

- In the below figure we can see, even the distribution is now normal. This type of data will create dependent relations between the features.



(a) Before removing the outliers

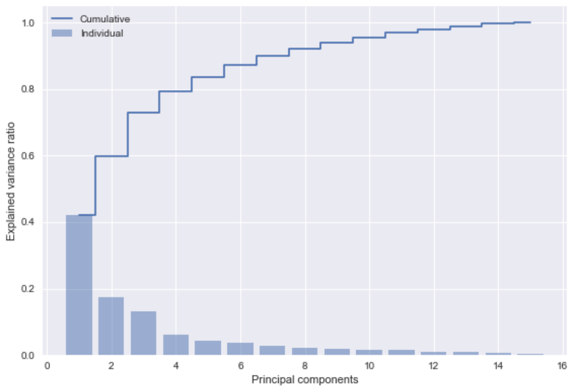


(b) After removing the outliers

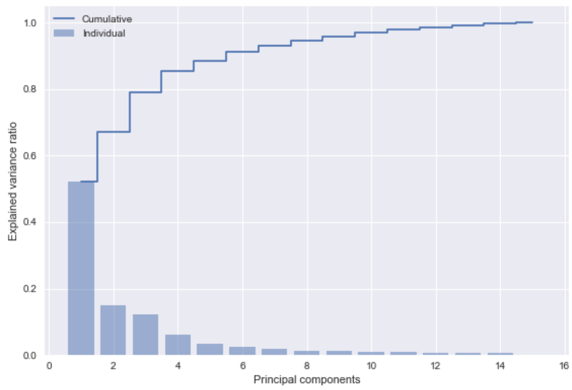
- So after we applied statistical test for dimensionality reduction.

3.4 PCA

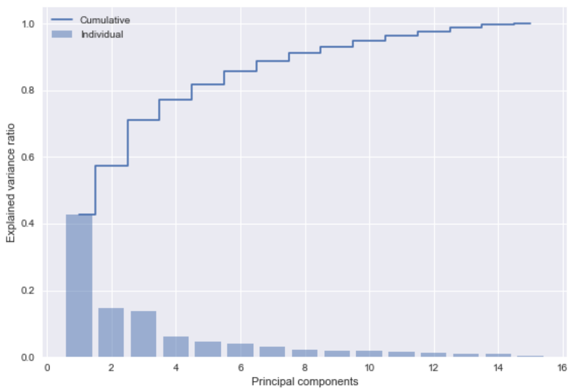
- Principal component analysis: PCA uses an orthogonal transformation to convert a set of observations of correlated variables into a set of linearly uncorrelated variables. These are called Principal Components.
- First covariance matrix is computed. Then eigen decomposition is performed on the covariance matrix and eigenvalues and vectors are computed. The PC's were sorted out based on the corresponding eigenvalues. And then the cumulative sum is computed and plotted as shown below.



(a) data without outliers



(b) filtered data using z-test



(c) filtered data using IQR

Figure 7: ratio of variance explained

- From the plots, 95% of the variance is within the first 10 principal components for the data without outliers.
- Whereas data filtered using z-test and IQR has 95% variance explained within the first 9 principal components and 11 principal components respectively.

3.5 Factor analysis

- A factor is a hidden variable which describes the association among the number of observed variables. They explain the variance of each variable.
- So we performed adequacy test using Kaiser-Meyer-Olkin Test, and the score is 0.856, which is excellent. It tells that there are hidden factors which describes the variables.
- Now factor analysis is applied for 6 factors.
- Factor 1 has high factor loadings for AF3, P8 and F8.
- Factor 2 has high factor loadings for FC5 and O1.
- Factor 3 has high factor loadings for F3, P7 and AF4.
- Factor 4 has high factor loadings for F7, O2 and T8.
- Factor 5 has high factor loadings for T7 and FC6.

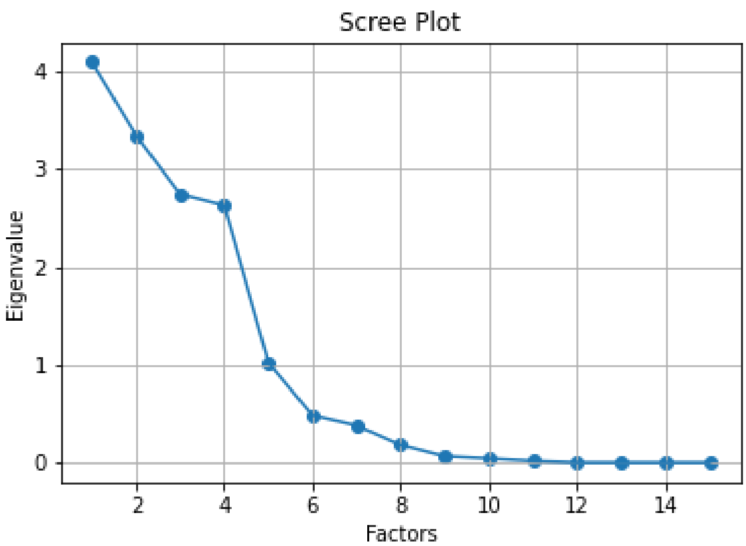


Figure 8: scree plot for actual data

- Here in the above figure we can observe an elbow at 5 number of factors. It represents that 5 factors are explaining the variance of all the features.

3.6 Correlation heatmap

- For the actual dataset, correlation heatmap is plotted which can be seen in below figure.

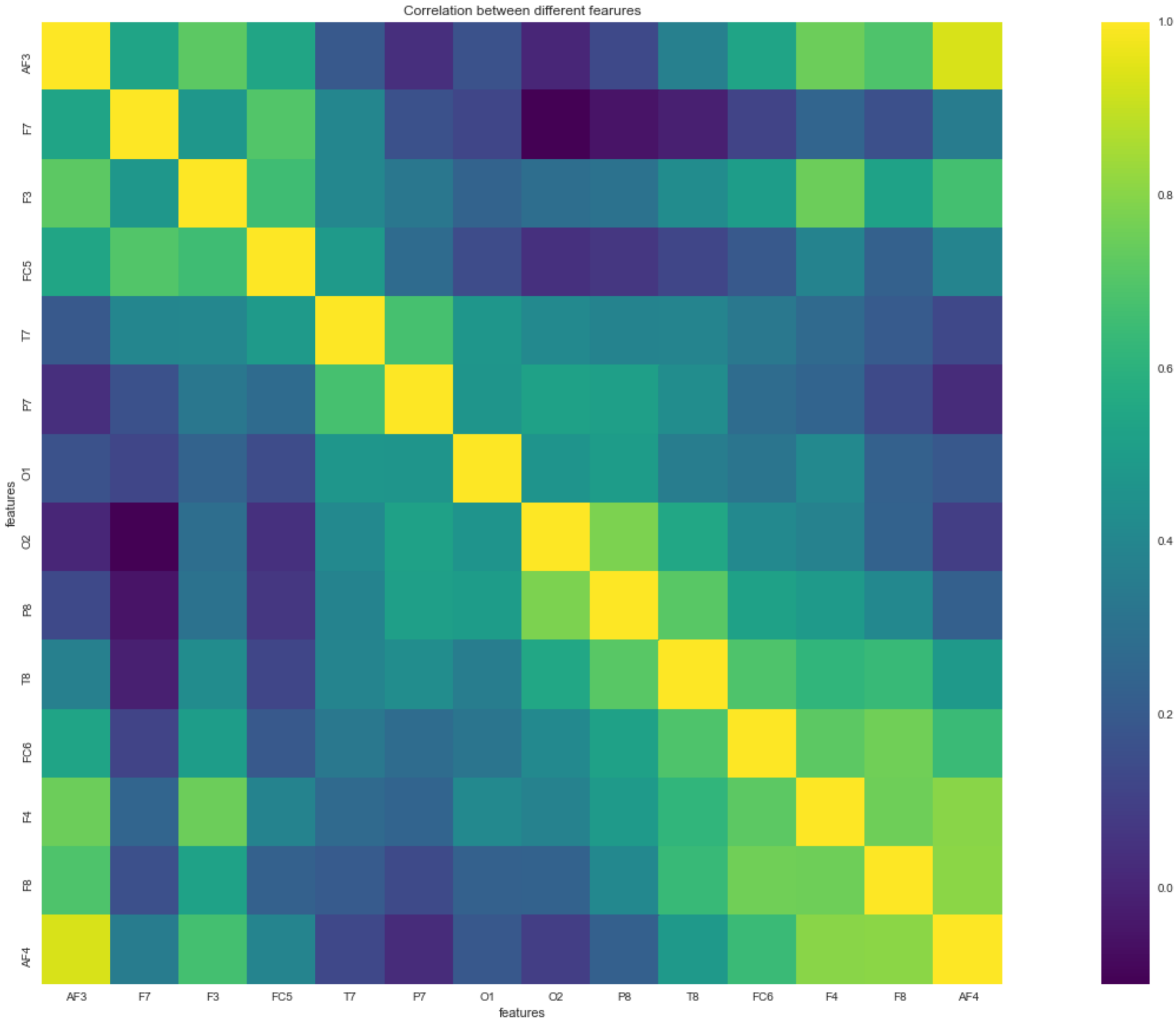


Figure 9: Correlation heatmap

- The yellow colour represents correlation of 1 and navy blue represents less than 0.
- The diagonal matrix represents correlation between features vs the same feature. So it is having correlation of 1.
- Whereas most of the features has either very less or no correlation with other.
- features such as T8, FC6, F4 and F8 have moderate correlation of around 0.4 to 0.7 with AF4.

4 Methodology

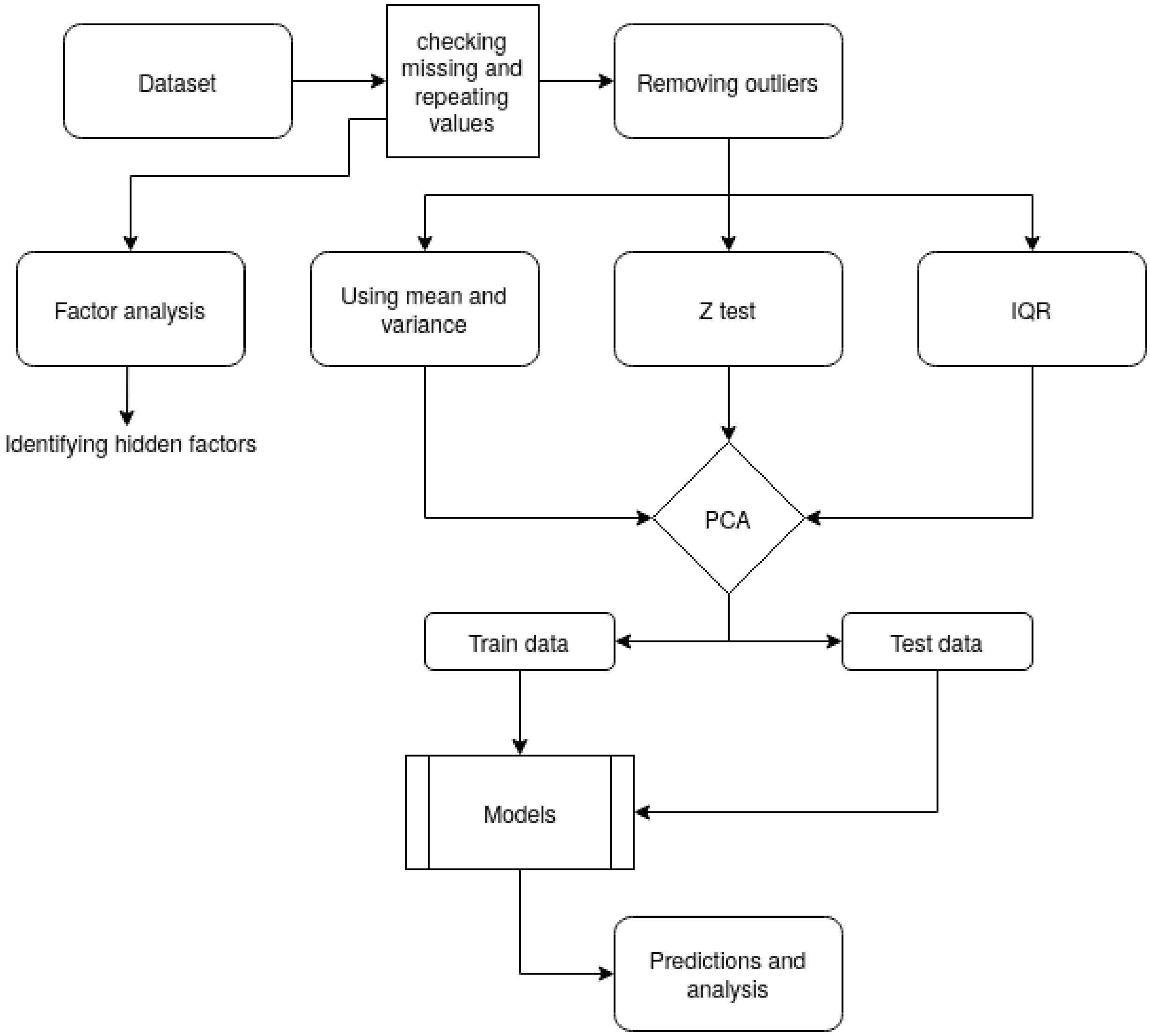


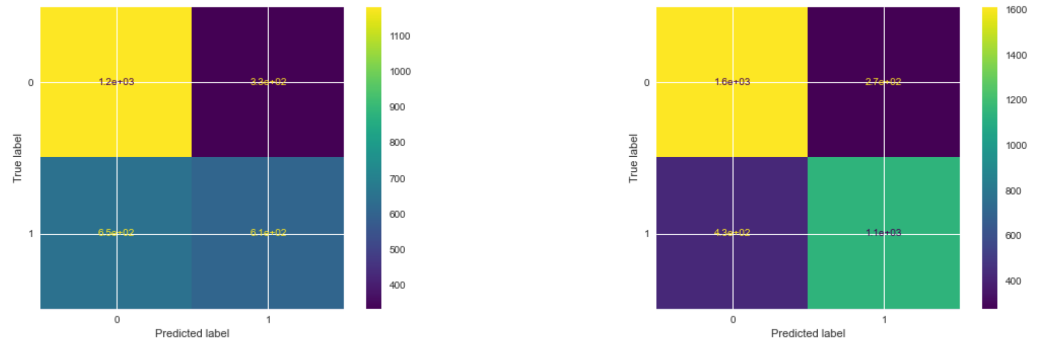
Figure 10: The overall flow of the project work

- So as we can see in the above figure, the data is first cleaned and outliers are removed using three methods.
- And for the actual dataset we have did factor analysis. They represent how those features are inter related to each other.
- Now PCA is applied on the filtered data and found the relevant number pf principal components required.
- For the respective principal components, we have split the data into 80:20 ratio for training and testing.
- Different classifiers are trained and tested. After a comparative analysis is made using the metrics such as accuracy, precision and Recall.

5 Implementation and Results

For our classification experiments, we used sklearn library for implementing machine learning models. All the datasets, models such as LDA, QDA, SVM-gaussian, SVM-polynomial, Logistic Regression, k-NN, Neural Network, Decision Tree, Random Forest, Gradient Boosting are applied.

- LDA is a classifier with a linear decision boundary, generated by fitting class conditional densities to the data and using Bayes' rule. whereas QDA is a variant of LDA in which an individual covariance matrix is estimated for every class of observations.

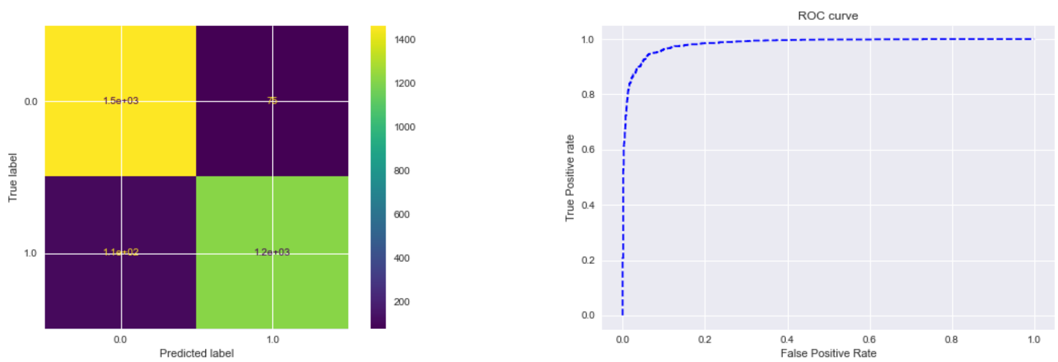


(a) Confusion matrix LDA

(b) Confusion matrix QDA

Figure 11: Confusion matrix for LDA and QDA

- SVM: This algorithms use a set of mathematical functions that are defined as the kernel. The function of the kernel is to take data as input and transform it into the required form. SVM algorithm with different types of kernel functions have been implemented. Gaussian is really effective in higher dimension, polynomial kernel with degree 3 is implemented. The ROC value observed when SVM with gaussian kernel implemented is 98.3.

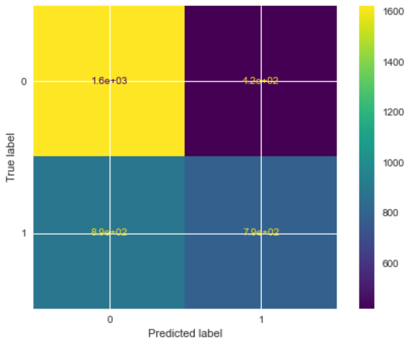


(a) Confusion matrix

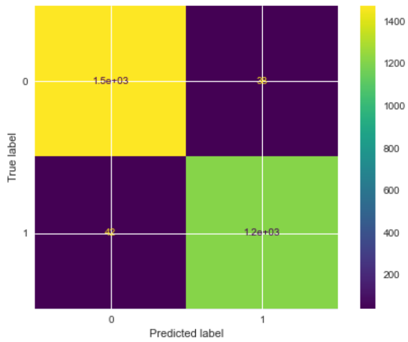
(b) ROC

Figure 12: Confusion matrix and ROC for SVM with gaussian kernel

- Logistic regression: Data is fit into linear regression model, which then be acted upon by a logistic function predicting the target categorical dependent variable. The learning depth (C) is set to 1.
- k-NN: The k-NN algorithm is arguably the simplest machine learning algorithm. Building the model consists only of storing the training dataset. The neighbor is set to 1.



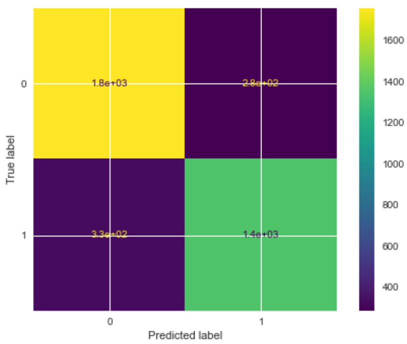
(a) Confusion matrix Logistic regression



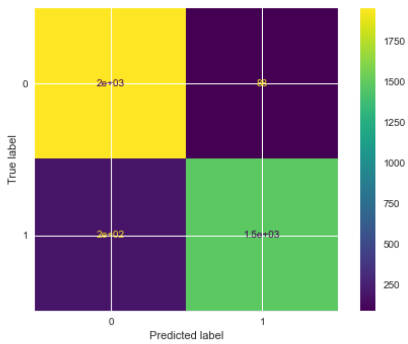
(b) Confusion matrix k-NN

Figure 13: Confusion matrix for LR and k-NN

- Decision tree: A decision tree is a supervised learning technique that has a predefined target variable. The training process resembles a flow chart, with each internal node, each branch is the outcome of that test, and each leaf node contains a class label.
- Random forest: It is an ensemble method, meaning that a random forest model is made up of a large number of small decision trees, called estimators, which each produce their own predictions.



(a) Confusion matrix Decision tree



(b) Confusion matrix Random forest

Figure 14: Confusion matrix for decision tree and random forest

- Neural network: For this we have implemented two neural networks. one using multipercentron neural network and the other with the same concept, but implemented by our own.

- This neural network has three hidden layers, both the input layer and hidden layers are with 64 neurons.
- Among all the models, whatever the models performed better, those model's confusion matrix are plotted.

6 Analysis

- First we have found the accuracy for different models for the raw data as shown in the below table

S.NO	Model	Accuracy
1	SVM-poly	55%
2	SVM-Gauusian	55%
3	KNN	98%
4	LDA	64%
5	QDA	50%
6	RFC	93.2%
7	GB	94.2%
8	Decision tree	84.1%
9	MLP NN	45%
10	ANN (own)	56.3%
11	logistic regression	63.3%

Table 1: classification analysis without PCA for different filtered datasets

- Now we can see the performance of different estimators for all the datasets using different filtering processes, with and without PCA can be seen in below table. So the number of features used without PCA are 14.
- Whereas with PCA, the data without outliers has 10 features, data filtered using z-test has 11 features and data filtered using 1QR has 9 features.

S.NO	Model	without outliers	z-test	IQR
1	SVM-poly	82%	74%	87%
2	SVM-Gauusian	94%	92%	96%
3	KNN	96%	93%	97%
4	LDA	63%	61%	65%
5	QDA	74%	73%	76%
6	RFC	90.5%	90.1%	93%
7	GB	91.1%	89.8%	92.9%
8	Decision tree	80.2%	82.3%	82.4%
9	MLP NN	91%	91%	94%
10	ANN (own)	95.5%	94.2%	90.7%
11	logistic regression	62.5%	61.4%	63.6%

Table 2: classification analysis with PCA for different filtered datasets

S.NO	Model	without outliers	z-test	IQR
1	SVM-poly	65%	66%	65%
2	SVM-Gauusian	60%	62%	61%
3	KNN	97%	97%	97%
4	LDA	64%	64%	64%
5	QDA	79%	78%	80%
6	RFC	92.6%	91.7%	93%
7	GB	83.3%	93.2%	93.8%
8	Decision tree	83.6%	83.6%	82.8%
9	MLP NN	46%	55%	45%
10	ANN (own)	90.8%	87.9%	91.1%
11	logistic regression	63.8%	64.8%	63.5%

Table 3: classification analysis without PCA for different filtered datasets

- From the computed results of estimators, feature selection based on PCA has shown better accuracy results when compared to without dimensional reduction.
- Though some of the models have accuracy lesser than the value when PCA is not applied, the computation has improved, since the generated features have less dimensions compared to previous data.

7 Conclusion

We can conclude that it is possible to predict eye state using EEG sensor input with an accuracy of around 97%. Even the models performed better, still there might be some influencing features which effect the sensor readings. Among all the classifiers SVM, KNN and ANN performed better after applying various filtering and dimension reduction techniques.

Another interesting observation is neural network performed well for the data from beginning without any filtering. So concentrating more towards neural network can lead amazing results, not only the eye state, but whatever the human activities connected to brain stimuli.

8 code

The drive folder for the whole project: All code files.
Link : https://drive.google.com/drive/folders/1SxpR6tLzotGjRSWi4cZYdMYgBjbcf2vg

References

[1] P. Pour, T. Gulrez, O. AlZoubi, G. Gargiulo, and R. Calvo, "Brain-Computer Interface: Next Generation Thought Controlled Distributed Video Game Development Platform," in Proc. of the CIG, Perth, Australia, 2008.

[2] L. Li, L. Xiao, and L. Chen, "Differences of EEG between Eyes-Open and Eyes-Closed States Based on Autoregressive Method," Journal Of Electronic Science And Technology Of China,, vol. 7, no. 2, 2009.

[3] B. Chambayil, R. Singla, and R. Jha, "EEG Eye Blink Classification Using Neural Network," in Proc. of the World Congress on Engineering, London, UK, 2010.