Analysis of the Impact of Online Education Using EEG Signals and Machine Learning Algorithms

by

Ehsanul Hoque
17101127
Tausif Ahmed
17101067
Mohammad Adituzzaman Shabab
17101007
Tahsin Mohammad Bakhtier
17101112
Sayeem Md Abdullah
17101009

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

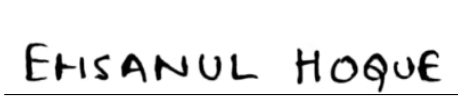
Department of Computer Science and Engineering
Brac University
January 2021

Declaration

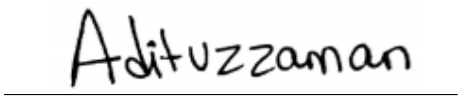
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Ehsanul Hoque
17101127



Mohammad Adituzzaman Shabab
17101007

Tausif Ahmed
17101067

Tahsin Mohammad Bakhtier
17101112

Sayeem Md Abdullah
17101009

Approval

The thesis/project titled "Analysis of the Confusion Level of Students Using Electroencephalogram (EEG) Signals and Machine Learning" submitted by

1. Ehsanul Hoque (17101127)
2. Tausif Ahmed (17101067)
3. Mohammad Adituzzaman Shabab (17101007)
4. Tahsin Mohammad Bakhtier (17101112)
5. Sayeem Md Abdullah (17101009)

Of Fall, 2020 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 15, 2021.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Mahbubul Alam Majumdar, PhD
Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement (Optional)

We, hereby declare that this thesis is based on the findings we obtained from our own research. Due acknowledgement has been made in the text to the dataset and all the other resources used. This thesis, neither in whole nor in part, has previously been submitted by anyone or any team to any other university or institute for the award of any degree.

Abstract

Online learning has allowed students from different walks of life to access a vast amount of information, allowing them to gain new skills. However, only having access to that information does not mean that the students will comprehend it. In this report, we study the impact of online education on students, specifically their confusion levels. The dataset that we have used in this report was taken from Kaggle. The dataset consists of mostly preprocessed Electroencephalogram (EEG) brain wave values i.e., Attention, Mediation, Raw, Delta, Theta, Alpha, Beta, and Gamma. Due to the limitations of the dataset, the accuracies of the Machine Learning models when only using EEG signal values were not satisfactory. Therefore, later into our research, we have decided to modify our dataset in order to better determine the confusion level of students. We have synthesized the dataset taken from Kaggle to form another dataset, where we took the content being viewed into account which led to better classification. The Machine Learning Algorithms that we have implemented in this paper are Decision Tree, Random Forest, Bagging with Random Forest, Gaussian Naive Bayes, K-Nearest Neighbors, Gradient Boosting, XGBoost, and Bidirectional-LSTM. For the dataset which consists of only EEG signal values, Bagging with Random Forest algorithm performed the best. It was able to predict whether or not a student was confused with an accuracy of 67.3%, while in the modified dataset, Bidirectional-LSTM had the highest accuracy of 80.9%. For both of the datasets, Gaussian Naive Bayes performed the worst with an accuracy of 59.2% and 63.6%, respectively.

Keywords: Electroencephalogram; Machine Learning; Online learning; Confusion levels

Acknowledgement

First and foremost, we want to thank the Almighty for His great support in helping us to proceed with our research with the difficulties we faced throughout the year. Furthermore, we want to thank our supervisor for guiding us, enduring our mistakes and giving us the necessary feedback to ameliorate our work. We also want to thank our parents and peers for their tremendous support throughout the semester.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In the past few years, there has been a shift from the traditional method through which students acquire knowledge. Due to increasingly flexible delivery modes, the students have a multitude of pathways and opportunities when pursuing higher education. However, a student completing an online course might face a number of barriers that may not be experienced by someone completing the same course in the traditional learning environment. Moreover, the current pandemic has disrupted the lives of many people across the world. It has also created a tremendous level of stress among the students which may lead to adverse effects on the learning and psychological health of students. Consequently, a student struggling or getting confused with material provided online could hinder their learning experience. Thus, we discuss throughout the report, how we can use EEG signals and complexity of the visual medium to ease the learning experience of a student.

1.2 Problem Statement

The past decade has seen an immense surge in online courses since people have easy and affordable access to the internet. The concept of online learning is not new to us, especially when the ongoing pandemic has forced all the academic institutions to move all of their offline learning to online learning. Many institutions have even developed a completely new online learning system/portal, which may be quite intimidating to students who are not well-versed in the technological implementation of online learning. Many students are more comfortable with the opportunity to ask questions to their respective teachers while being physically present in class. Such students are prone to get more confused while learning the same topic online.

From the perspective of Bangladesh's learning system, it is known that the majority of the students here have been accustomed to in-class learning. Hence, the sudden shift to online learning is an entirely new experience for them. This new environment of understanding could lead them to face difficulties that did not exist before. There could be several reasons why online learning might result in unsatisfactory performance from students. Students are less likely to give attention to online classes since they are not being monitored. The lack of attention could result in students not understanding a topic, and the inability to ask questions directly to teachers

might result in them being more perplexed.

So, we can see that online learning has a lot to offer as it comes with both advantages and disadvantages at the same time. However, in our report, we plan to acknowledge the impact of online learning, where it becomes a hurdle for the students, especially when the students are likely to become confused.

1.3 Aim of Study

Predicting the confusion levels in general, and of students while viewing Massive Open Online Course (MOOC) video, with the aid of EEG signals and Machine Learning is the aim of our study. In our thesis, we have attempted to prove that with the aid of EEG signals and Machine learning, these two can be great tools in analyzing and predicting confusion levels. The dataset consists of various EEG signals, i.e. attention, mediation, raw, delta, theta, alpha, beta and gamma, obtained from students as they viewed MOOC video clips. All the signals had relatively similar contributions to our findings. Later we discovered that by categorizing what is being viewed and including that in our dataset, it led to better performances as they were of varying confusion levels.

1.4 Research Methodology

Our goal is to detect whether a student is confused or not. Due to the current pandemic, we proceeded with our research using a dataset we obtained from Kaggle [1]. While conducting exploratory data analysis (EDA), we identified large variances in the values of the features of the dataset. Hence, we decided to use different scaling techniques to bring the features to a comparable scale. The scalers that we made use of were MinMaxScaler, StandardScaler, MaxAbsScaler, Quantile Transformer, Power Transformer and RobustScaler. We discovered Power to be the best performing scalers. We also applied dimension reductionality techniques, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), to choose the most significant features. We also applied t-distributed stochastic neighbor embedding (t-SNE) for visualizing the high dimensional dataset. Then, we trained various machine learning models with the processed dataset. We used the tree based algorithm such Decision Tree, the ensemble learning method like Random Forest, the probabilistic classifier like Gaussian Naive Bayes, the non-parametric algorithm like k-nearest neighbor (KNN), decision-tree-based ensemble algorithm like XGBoost, the neural network Bi-directional long short term memory (Bi-LSTM), and finally the ensemble algorithm Bagging Classifier with Random Forest. In the later part of this report, metrics like accuracy, precision, recall and f1-score have been computed for all the models to identify the most suitable model for our prediction.

However, if we stick with EEG signals only, we have seen that the accuracy and result that our models give is not satisfactory. While training our first dataset, we have seen that more students were likely to get confused while watching some particular videos. This has happened because some videos were more complex than others, as a result of which more students tend to get confused over the complex videos.

Therefore we have decided to take advantage of this extra information, and have decided to create a modified dataset, which would not only take the EEG signals from students, but would also take another feature called VideoID. Our models can use this extra information and learn which videos are more complex. While using this alternative approach, we could increase the performance of our models significantly. We have synthesized our first dataset to train the models. The EEG signals from the first dataset were completely unchanged. The only change that we made was to replace the VideoIDs of more complex videos with CourseIDs of 400 level courses, and the VideosIDs of less complex videos with CourseIDs of 100 level courses. We have then used one hot encoding to preprocess that categorical data. Then we have applied all the techniques mentioned above and compared our results.

1.5 Thesis Outline

The target of our research was to establish the fact that Machine Learning can be used with EEG Brainwave signals to distinguish confusion levels, which can subsequently be used to assess difficulty levels and effectiveness of online learning. We used a dataset from Kaggle , performed necessary processing, synthesized our new dataset, and used the dataset to train supervised Machine Learning models to classify confusion states [1]. The overall report focuses on the steps that were followed in the research.

Chapter 1 states the inspiration behind our work. We have addressed how the ongoing COVID-19 pandemic has forced traditional learning to take a more online approach than the world had ever seen before. The goal of our research, that is to determine the confusion level of students while watching online courses, is discussed in this chapter.

Chapter 2 is dedicated to the research works that have already been done related to our topic. We have given a brief overview of the works of the other researchers, and have discussed their findings.

Chapter 3 has a detailed description of our dataset. We have addressed how we have used secondary data to fit our purpose. We have also described how we have synthesized the secondary data to form a dataset of our own preference. The limitations of both of the dataset were also discussed here. This section contains information about how we have applied techniques like scaling, PCA, LDA, t-SNE to our dataset. This section also contains data visualizations for better understanding of our dataset.

Chapter 4 is about the results and findings of our research. We have given a brief description of the models that we have used to classify our problem. The results that were extracted from the models were then tabulated and visualized in this chapter.

Finally, in Chapter 5, we have given a conclusion of our research work. We have also talked about how in the future, we can improve our results. Scopes about further improvement of our dataset and results were addressed in this chapter. Figure 1.1 illustrates the thesis outline with the help of a workflow diagram.

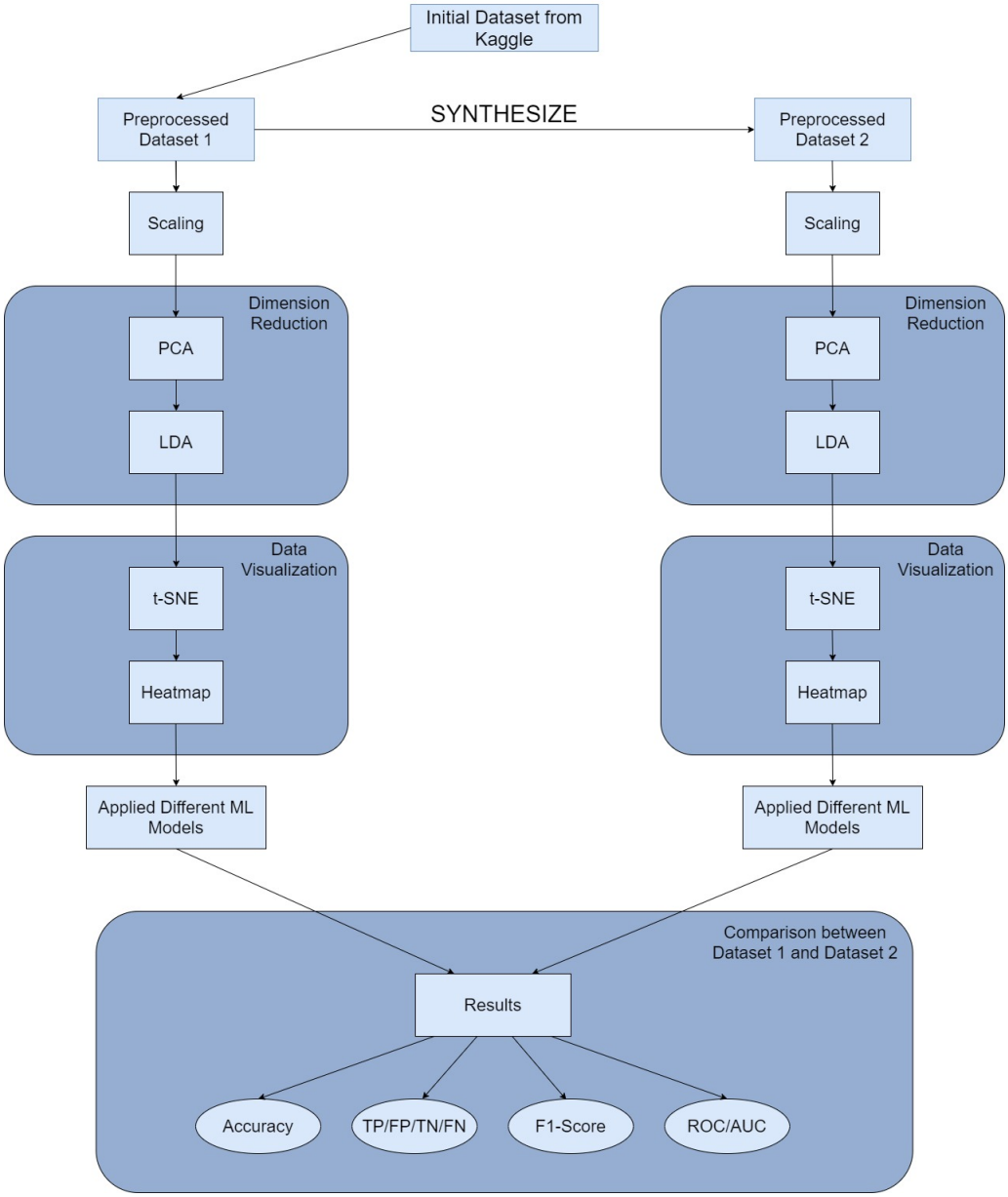


Figure 1.1: Workflow of methodology

Chapter 2

Related Work

It is natural to be perplexed when understanding a complicated topic online through a visual medium. Not understanding the way a topic is being taught is one of the many reasons for the underperformance of students. Recently, there have been related works that use various machine learning models on EEG data to predict or analyze the outcome for mental states such as confusion.

In [2], the authors initially used their dataset for their research paper, which we will be using for our study as well. They had trained Gaussian Naive Bayes classifiers for estimating the probability that a given session was perplexing for a student. This method contains logistic regression because it is typically the best approach for solving problems that include sparse and noisy training data [3]. The mental states that were to be predicted were labeled in two ways: one was set by the authors by the way they set up the experiment called pre-defined confusion level and the other was set by the user based on the difficulty they faced whilst viewing the content called user-defined confusion level. Characterizing the EEG signals' overall values was done by computing their means throughout each utterance. Several features were calculated to illustrate the EEG signal's temporal profile, such as minimum, maximum, variance, skewness, and kurtosis. Unfortunately, when having a small number of data points, the inclusion of these features normally overfit the training data. Finally, only the means of those features were used as the classifier features. They applied the cross-validation technique in order to ensure that there was no chance of overfitting. They trained the student specific and independent classifiers differently. They only kept the data for one student when training the student specific classifier. After which, they applied cross-validation. However, they trained the student independent classifier on all students except for one. Finally, they calculated the average of the accuracies that they had obtained and performed one final cross-validation among the students. The average accuracies of student- specific and student independent classifiers were 67% and 57%, respectively. It was observed that both classifiers performed significantly above chance ($0.5(p<0.05)$) in 6 out of 9 students. While training and testing classifiers for student-defined confusion, it was considered that students have different senses of confusion. Hence, the confusion rating that they had created was transformed to a binary rating of confused or not confused, where there remained approximately a similar amount of confused or not confused classes. They also ensured that the training data was balanced. The average accuracy of student-specific and student-independent classifiers was 56% and

51%, respectively. As a result, it did not manage to outperform the student classifier.

In another paper [4], the authors used the confusion dataset for their medical research [1]. Their primary goal was to increase the accuracy of neural networks when used for diagnosis by eliminating characteristics like gender and age. Their experiments involved various medical techniques such as MRA, CT-scan and EEG signals while using LSTM and CNN to identify efficiency of the model. For each of the experiments, they eliminated characteristics like gender and age using the confounder filtering (CF) method to observe if accuracy improved. Any weights associated with these factors were identified, then tracked the ones which were updated during the training phase of the model and replaced with zeroes. This led to the creation of a model exempt from confounding factors. Finally, they used the Bidirectional LSTM where they set the number of neurons to be 50. The activation function they used was tanh. Then, the output was calculated as it was connected to a fully connected layer where the activation function was sigmoid. Lastly, they performed five-fold cross-validation to avoid overfitting using the CF-improved Bidirectional LSTM. The results they obtained showed an improvement to the predictive performance once the CF method was applied compared to previous results. An accuracy of 75% was achieved when using the CF-improved Bidirectional LSTM.

The authors in [5] wanted to improve MOOCs' performance by creating an evolutionary online framework and used the confusion dataset for their initial training of their model [1]. In the recent past, the evidence of students experiencing mental fatigue while participating in online courses has been reported [6]. Mental fatigue can significantly hinder the learning experience of students. This is less likely to occur during in-class learning, where the instructor is present to respond to the student by asking questions or sharing fun anecdotes. The successful implementation of this framework means detecting the student's confusion levels in real-time via the EEG headset. As a result, the instructors will be able to track the students' progress and their difficulties with the materials. This information would allow them to quickly take the necessary steps to ease the learning of the students. Their research used a multi-objective genetic programming strategy based on non-dominated sorting genetic algorithm II (NSGA-II) considering the optimization of the area under the ROC curve as the fitness and subtree complexity as the complexity measure simultaneously. The 11 extracted features from the EEG signals of the confusion dataset used as the inputs of the Genetic Programming classifier. They also tried to find the correlation between each of the features with each other. Hence, they formed a correlation matrix. Thus, they concluded that beta and gamma signals showed the highest linear correlation among the 11 extracted features. The GP model ran for 5000 generations with 1000 population considering 5-folds cross-validation to overcome any possible over-fitting. They calculated specific metrics such as accuracy, precision, recall, f-score, false-positive rates, and false-negative rates. They achieved an accuracy of 89.16% when using the GP function classifier to detect the students' confusion level. The comparative study that they performed suggests that it greatly outperforms other classifiers used in past experiments.

In [7], the authors wished to conduct an analysis on the level of confusion in students by analyzing their EEG signals . The study conducted by them is based on the as-

sumption that the EEG signal of students who are confused will differ from those who are not confused. EEG signals give different waveforms, depending on the state of one's mind. EEG signal is extracted from an individual by placing electrodes on their scalp, which reads the electrical activity of the individual's brain [8]. The voltages between neurons of the brain are primarily read by the EEG. The amplitude and frequency of such voltages are then measured [9]. This study's main objective was to detect whether students are confused or not confused while watching Massive Open Online Course (MOOC) videos. For their research also, they have used the confusion dataset from Kaggle [1]. The authors have implemented 32 classifiers to work out which model works best for the analysis of EEG signal and confusion level. A train-test split of 70-30 was done. Later cross-validation of 5 folds was used to validate the model. For each classifier, four performance metrics were used, which are (a) Accuracy, (b) Precision, (c) Recall, (d) F1-Score. After training and testing were done, it was found out that Bagging with Random Forest gave the best result, with an accuracy of 66.6%, followed by Random Forest, with an accuracy of 65.89%, and Bagging with Extra Trees, with an accuracy of 65.69%.

In [10], the authors have tried to achieve the best machine learning model to predict whether a student is confused or not from EEG signals. The paper is based on the hypothesis that when a person is a confused mental state, there are vast differences between their EEG Signals compared to that of someone who is not confused. This motivated the authors to predict student confusion while watching online course videos. In the paper [11], the authors have shown 99.3% accuracy in their research of detecting drowsiness of car drivers using Support Vector Machines (SVMs). By using the same classifiers, a group of authors wrote a paper [12] in which they classified EEG signals of a person having epileptic seizures with an accuracy of 100%. This paper explored other methods to improve the results of predicting the same outcome using the same dataset. The data from this research is collected from Kaggle [1]. The authors used Batch Normalization technique. By doing so, they were able to ensure that the value of each and every feature is normalized, and eventually has a standard deviation of 1 and a mean of 0. During the process, the size of the test data was chosen to be 20, which was decided based on keeping the size of the test data to be equal. This decision was taken based on the paper [13] where the authors put forward a layer of batch normalization, in which features of a deep neural network are standardized. The standardization is done with the help of mini-batch statistics. By doing so, the process takes place much faster with higher accuracy. In this research, to make the best use of the EEG data properties and as the goal of the research is a binary classification, the authors proposed an LSTM Recurrent Neural Network. A Long Short-Term Memory, also known as an LSTM is dependent on the previous input to derive its current output. A Bi-Directional LSTM, as the name suggests, does not only depend on the previous input, but also the future in order to derive its current output. Hence a Bi-Directional LSTM model with 50 neural units was used in this framework. The activation function for the LSTM layer was "Tanh". The hidden states after the LSTM layer were given as an input to a fully connected layer of neural networks with the sigmoid activation function. The outputs achieved are between 0 and 1. The outcomes of this framework were compared with other baseline classification methods such as SVM (linear kernel), SVM (rbf kernel), SVM (sigmoid kernel), K-Nearest Neighbors,

Convolutional Neural Networks, Deep Belief Networks, and RNN-LSTM, it was found that Bi-directional LSTM has the best performance. Upon performing 5-fold cross-validation, the Bi-directional LSTM model's accuracy varies from 71% to 74%, showing that this model is not all the most accurate, but the most consistent. By dropping a feature every time in 12 experiments, they identified gamma-1 to contribute most to the model as it dropped the most accuracy.

Chapter 3

Data Description and Feature Engineering

3.1 Description of Dataset

In this chapter, there is a detailed explanation of the function of EEG, how each brain wave correlates to activities in our daily lives. There is also a description in [2] where the authors elaborated on how they constructed the original dataset and how we created a synthesized dataset to meet our research objective.

3.1.1 EEG

The EEG of electroencephalogram signal reflects the brain's electrical activity by calculating certain factors related to brain waves. In nature, they are highly random, non linear and non stationary, however, even if it is very strenuous to get useful information directly just by observing, extracting important features for different purposes may show us a pattern to utilize them [14]. The cortical nerve cell inhibitory and excitatory postsynaptic potentials summate in the cortex and extend to the scalp surface where they are recorded as EEG. There are a number of steps taken during an EEG test. One is the placement of electrodes on certain points of the scalp of the test subject using removable adhesive. An EEG recording machine and an amplifier is used to form a bridge with each electrode. This results in signals on a monitor which is in the form of wavy or distorted lines that represents the converted electrical brain signals. A typical EEG signal measured from the scalp, will have a range of about 1 µV to 100 µV in a normal adult (which is approximately 10-20 mV when measured with subdural electrodes such as needle electrodes), and a frequency in the range of 1 Hz to about 100 Hz. Since the architecture of the brain is nonuniform and the cortex is functionally organized, the EEG can vary depending on the location of the recording electrodes. The placement of the electrodes is of high importance, because different lobes of the cerebral cortex are responsible for processing different types of activities. The standard method for the scalp electrode localization is the international 10-20 electrode system. In this method the actual distances between neighbouring electrodes are either 10% or 20% of the total front to back or right to left distance of the skull. The positions are largely determined by two points: "nasion", the point between the forehead and the nose, level with the eyes, and "inon" which is the bony prominence at the base skull on the midline at

the back of the head [15]. The frontal lobe is separated from the parietal lobe and the temporal lobe by central sulcus and lateral sulcus respectively. It is generally where higher executive functions such as personality, emotional regulation, problem solving, motor development, planning and reasoning, parts of speech and move etc [16]. This is the reason for a single channel of Neurosky mindset being placed on the frontal lobe since the EEG signal generated inside this region is of our interest.

One of the key factors for comprehending human behaviour in cognitive study and assessing abnormalities is frequency. It is a representation of the repetitive pattern within a certain amount of cycles in seconds. It is seen that the amplitudes and frequencies of signals change from one state to another, for example wakefulness and sleep. There are five major brain waves explicitly distinguished by the different frequency ranges which are delta (δ), theta (θ), alpha (α), beta (β) and gamma (γ), sorted from the lower frequency bands to the higher frequency bands.

3.1.2 Brain Waves

Human brain is an electrochemical organ which consists of billions of neurons, where each neuron is connected to thousands of others, creating a massive, complex network of neurons. This enormous network facilitates the communication that allows us humans to have feelings, thoughts, emotions and behaviors, making each of us unique from the others. This communication takes place among the masses of neurons, generating electrical pulses. Brainwaves are the product of this synchronized phenomena. This electrical activity can be detected by using EEG.

Delta: Delta (δ) waves are the brain waves with the lowest frequency with the range from 0.5Hz to 4Hz. They are mostly found in infants and young children. Humans tend to produce less delta waves even during sleep, as they age [17]. These waves are associated with the deepest levels of relaxation and restorative sleep. This slowest brain activity is found in all stages of sleep, especially in stage 3 and 4 [18]. Delta values which are not normal indicate that a person would face difficulty in learning, or even find it difficult to keep up their cognitive awareness. Sometimes, not having normal delta wave activities may even cause injuries to the brain, or severe ADHD. Figure 3.1 shows a typical Delta wave [15].

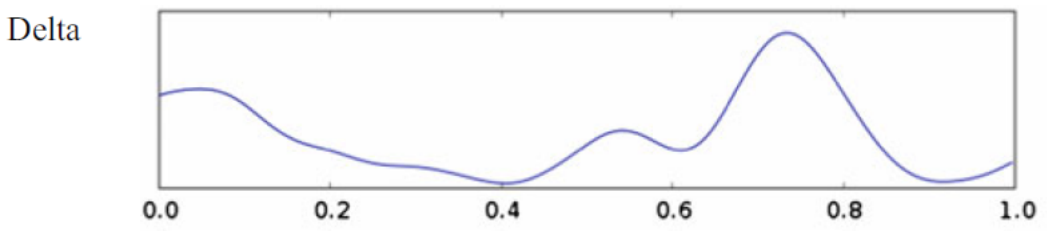


Figure 3.1: Delta Wave

Theta: Theta (θ) waves are in the mid-lower range (4Hz to 8Hz) [17]. These are associated with "experiencing" and feeling deep and raw emotions. Excessive theta

activity may make people prone to being depressed and make them "highly suggestible" based on the fact that they are in a state where they are deeply relaxed. Optimal theta activity can improve creativity, emotional connection, intuition and relaxation. As stated before anomaly in theta waves may indicate to ADHD, depression, hyperactivity, impulsivity, inattentiveness when there is too much theta activity, and in contrast, too little theta activity indicates anxiety, poor emotional awareness and stress [17]. Figure 3.2 shows a typical Theta wave [15].

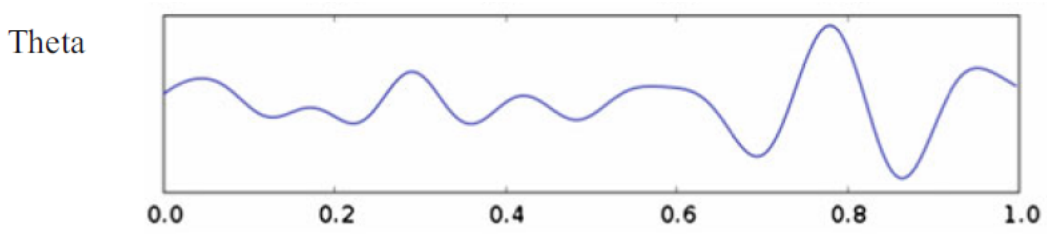


Figure 3.2: Theta Wave

Alpha: The frequency range of alpha (α) is a bridge between subconscious mind and cognitive awareness [17]. It has a frequency range of 8Hz - 12Hz which is moderate. It promotes the feeling of deep relaxation. It is abnormal in adults but normal for children under 13 years. It promotes the production of human growth hormone, that increases relaxation, helps memory and learning [18]. Becoming stressed may cause a phenomenon called "alpha blocking" to occur, when there is excessive beta activity and very little alpha. Essentially the production of alpha gets stalled by the beta waves, when humans become too aroused. Low production of alpha waves indicate anxiety, high stress, insomnia and OCD [17]. Figure 3.3 shows a typical Alpha wave [15].

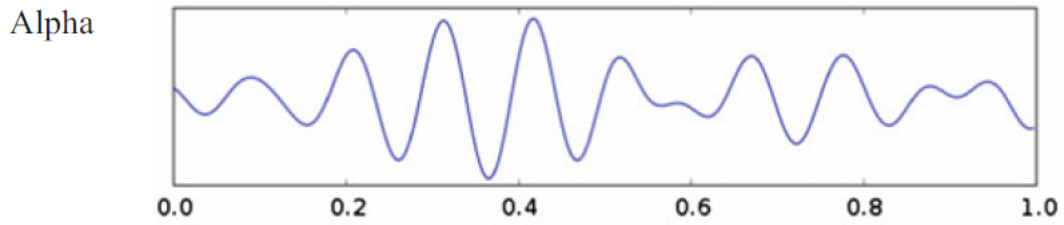


Figure 3.3: Alpha Wave

Beta: Beta (β) wave activities are prominent when humans are awake with a fully conscious mind. Beta waves are known to have high frequency (12Hz to 40Hz) and low amplitude [17]. It is usually seen on both sides of frontal and parietal lobes [18]. Conscious thought, logical thinking are associated with beta waves. The optimal amount of beta wave production promotes focus and ease of mundane works and increment in problem solving ability. If an individual's beta wave frequency is found to be high, the individual's arousal level will be elevated. High anxiety and stress level is also associated with high beta frequencies. This is the brain wave that is initially exhibited while completing conscious tasks which involves critical thinking, writing, reading or socialization [17]. Figure 3.4 shows a typical Beta wave [15].

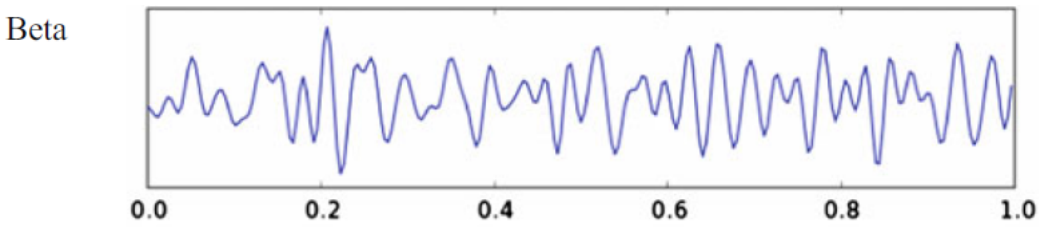


Figure 3.4: Beta Wave

Gamma: Gamma (γ) activities can be seen when an individual is required to perform a task which needs a highly processing mind and intensive cognitive thinking. It has the highest frequency range (40Hz to 100Hz) [17]. These waves are important for learning, memory and information processing. It is believed that a 40Hz gamma wave is important while learning a new skill or material. Gamma activities which are lower than average can be found in mentally challenged individuals who will suffer from learning disabilities [17]. Figure 3.5 shows a typical Gamma wave [15].

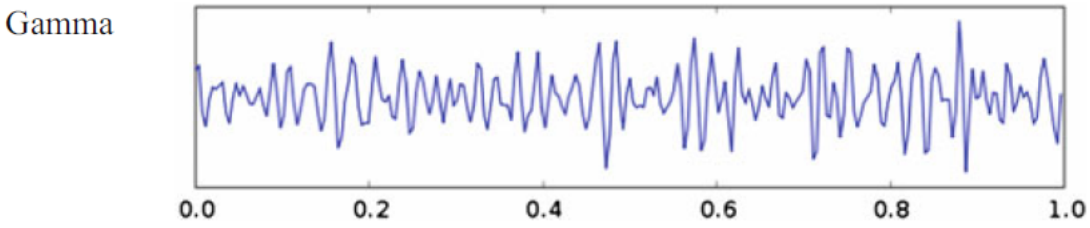


Figure 3.5: Gamma Wave

3.2 Initial Dataset

The dataset is publicly available from Kaggle, which provides open-source data for various challenges, under the name of "EEG brain wave for confusion" [1]. 10 college students were asked to wear a single-channel wireless MindSet EEG device that detected their brain activity over the frontal lobe by placing an electrode resting on the forehead and the two other electrodes (one ground and one reference), being placed on two ears, each in contact with the other [19]. The features extracted by the device is shown in table 3.1 along with the additional features which are "SubjectID", "VideoID", "pre-defined label" and "user-defined labeln". "Attention" indicated the level of mental focus of the student, and "Meditation" measures the calmness. "Raw" is the average of the original EEG signals. The features Delta, Theta, Alpha1, Alpha2, Beta1, Beta2, Gamma1, and Gamma2 are values in different frequency regions of the power spectrum, all being continuous data. Along with these 11 features, there is also "SubjectID", "VideoID" and "pre-defined label". "SubjectID" consists of values from 0-9, a value representing the subject of each video recording, and "VideoID" is the same for the videos. "Pre-defined label" is a value of either 0 or 1, indicating whether the video has already been labeled as easy or difficult. None of these 3 are used as features of our machine learning models when using the first dataset. During the experiment, while wearing the device, the

students were asked to watch 10 videos each. The researchers prepared 20 videos, 10 in the category of being pre-labeled as "easy", which consisted of videos of topics that were not confusing for college students such as basics of algebra and geometry. The other 10 videos were pre-labeled as "difficult" and covered videos on areas college students are not familiar with, such as Quantum Mechanics and Stem Cell Research. To make the videos even more confusing, the two minute long videos were clipped from the middle and presented to the students. The entire dataset has 12000+ rows, where a single data is collected every 0.5 seconds. For each subject watching a video, features are extracted at a sampling frequency of 2 Hz. The final features are truncated to around one-minute long. For the 10 students watching 10 videos, there are 100 data points where each data point is a one-minute video, consisting of 120+ rows. Signals with higher frequency are represented as the mean value in every 0.5 seconds. After viewing the videos, each student rated his or her confusion level on a scale of 1-7, where 1 indicated the student being least confused, and 7 indicated the highest level of confusion. The labels were quantized into two classes of students, representing whether the students are confused or not. The two-class label serves the aim of our prediction task, where we use machine learning models to predict whether they are confused or not. In the experiment, 8 out of 10 participants had the age of 24/25 and the other two participants were 28 and 31 years old, which results in the variation in the age group to be really restricted. While age might have been a primary feature to affect the level of confusion in humans (for example infants and olderly people typically should have a higher confusion level than the one of a fully grown adult within age 20 to 30), for our research it was not a significant contributing factor into the levels of confusion. The * in Table 3.1 refers to the features from the Mindset.

Features	**Description**	**Sampling rate**	**Statistic**
Attention*	Proprietary measure of mental focus	1 Hz	Mean
Mediation*	Proprietary measure of calmness	1 Hz	Mean
Raw*	Raw EEG signal	512 Hz	Mean
Delta*	1-3 Hz of power spectrum	8 Hz	Mean
Theta*	4-7 Hz of power spectrum	8 Hz	Mean
Alpha1*	Lower 8-11 Hz of power spectrum	8 Hz	Mean
Alpha2*	Higher 8-11 Hz of power spectrum	8 Hz	Mean
Beta1*	Lower 12-29 Hz of power spectrum	8 Hz	Mean
Beta2*	Higher 12-29 Hz of power spectrum	8 Hz	Mean
Gamma1*	Lower 30-100 Hz of power spectrum	8 Hz	Mean
Gamma2*	Higher 30-100 Hz of power spectrum	8 Hz	Mean
SubjectID	N/A	N/A	N/A
VideoID	N/A	N/A	N/A
pre-defined label	N/A	N/A	N/A
user-definedlabeln	N/A	N/A	N/A

Table 3.1: Features in the initial dataset

3.3 Pre-processed Dataset

3.3.1 Preprocessed Dataset 1

We strictly wanted to stick with EEG signals for Dataset 1. We wanted to study how only the EEG signals affect the confusion level of students. Therefore, the initial dataset was preprocessed to remove the SubjectID, VideoID and pre-definedlabel columns. This leaves us with a dataset which only contains information regarding EEG signals extracted from the students. Table 3.2 shows the features of Dataset 1.

Features
Attention
Mediation
Raw
Delta
Theta
Alpha1
Alpha2
Beta1
Beta2
Gamma1
Gamma2
user-definedlabeln

Table 3.2: Features used in Dataset 1

3.3.2 Preprocessed Dataset 2

In our initial dataset, we have a column called VideoID. We have synthesized our initial dataset to obtain Dataset 2 using the help of that particular column. Dataset 2 not only has the EEG signals from the students, but it also has a tag of which particular video the student is watching. This alternative approach should allow our models to achieve better results. The reason behind why our models achieve better accuracy while training over the new synthesized dataset is because our models can now learn which particular video with particular VideoID creates more confusion. For example, in the initial dataset which has 12811 rows, there are 1158 rows where VideoID is 8. Within these particular rows, 896 rows has user-defined labels as 1, while only 262 has user-defined labels as 0. This indicates that students tend to get confused while watching a video with VideoID of 8. Our models, therefore, can learn from this occurrence and use this extra piece of information, alongside with the EEG signals, to predict better results. Table 3.3 shows the features of Dataset 2. The details of how we have feature engineered our way from our initial dataset to Dataset 2 is described in section 3.4.2.

Features
CSE_110
CSE_111
CSE_230
CSE_260
CSE_330
CSE_340
CSE_420
CSE_421
CSE_422
CSE_427
Attention
Mediation
Raw
Delta
Theta
Alpha1
Alpha2
Beta1
Beta2
Gamma1
Gamma2
user-definedlabeln

Table 3.3: Features used in Dataset 2

3.4 Feature Analysis and Feature Engineering

3.4.1 Scaling

One of the most vital steps in creating a well performing machine learning model is feature scaling during the pre-processing of the data. Feature scaling may distinguish a good machine learning model from a relatively poor performing one. As the machine learning model tries to infer from the values of the data only, when values from both the extreme high and low ends of a given range co-exist, the algorithm assumes that the values in the extreme high end may have a more decisive role in the target variable. That may not necessarily be the case and as a result, the decision making of the model has become clouded.

Figure 3.6 shows some graphical representations of the input variables from a statistical viewpoint prior to scaling.

One of the many algorithms which do benefit from data being scaled is the K-nearest Neighbors algorithm. As that is one of the many algorithms we have worked with in our research, also as there are see the extremely high values of standard deviation in the band value columns, we decided to scale our data. The various scalers we have used to scale the data in our research are:

	Attention	Mediation	Raw	Delta	Theta	Alpha1	Alpha2	Beta1	Beta2	Gamma1	Gamma2
count	11536.000000	11536.000000	11536.000000	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	11536.000000	1.153600e+04	11536.000000	11536.000000
mean	45.880028	52.397451	34.339806	5.900149e+05	1.588479e+05	3.857353e+04	2.816351e+04	20256.508148	2.462005e+04	16925.292042	8214.482490
std	19.641741	17.226824	131.652667	6.365997e+05	2.365106e+05	6.860648e+04	4.838641e+04	29163.164448	3.711200e+04	25797.069942	11541.638785
min	0.000000	0.000000	-2048.000000	4.480000e+02	1.700000e+01	2.000000e+00	2.000000e+00	3.000000	2.000000e+00	1.000000	2.000000
25%	34.000000	43.000000	-10.000000	9.010200e+04	2.408875e+04	6.628750e+03	6.465750e+03	5678.000000	6.952000e+03	3758.000000	1959.750000
50%	47.000000	53.000000	34.000000	3.684790e+05	7.338950e+04	1.645600e+04	1.386600e+04	11598.000000	1.417650e+04	8590.000000	4470.000000
75%	60.000000	64.000000	80.000000	8.913540e+05	1.905018e+05	4.117225e+04	3.042725e+04	24025.750000	2.923800e+04	19290.000000	9829.500000
max	100.000000	100.000000	1440.000000	3.964663e+06	2.567643e+06	1.369955e+06	1.016913e+06	840994.000000	1.083461e+06	658008.000000	283517.000000

Figure 3.6: Statistical representation of input variables before being scaled

1. MinMax Scaler
2. Standard Scaler
3. MaxAbs Scaler
4. Robust Scaler
5. Quantile Transformer
6. Power Transformer

Figure 3.7 illustrates how different scalers perform under different classification models. From the figure, we can see that there are only a few cases where the Power transformer scaler is outperformed. The PowerTransformer scaler is outperformed by MinMax and MaxAbs scalers for Gaussian Naïve Bayes Classifier, and is outperformed by Quantile scaler for Gradient Boosting classifier. However, for all the other classifiers, namely the Decision Tree, Random Forest, Bagging with Random Forest, K-Nearest Neighbors, XGBoost and Bi-LSTM, the PowerTransformer scaler either outperforms all the other scalers, or is almost on par with other scalers. Hence, we have decided to stick with PowerTransformer scaler for the rest of our research.

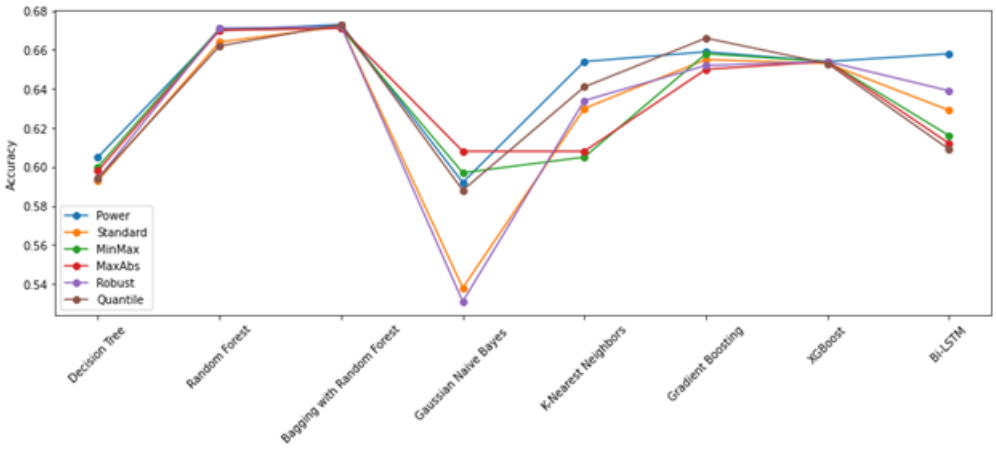


Figure 3.7: Accuracy Comparison of Different scalers on Machine Learning models

The PowerTransformer belongs to a class of parametric and monotonic transformations that make the data close to that of a Gaussian Distribution. It automatically finds the optimal scaling factor required to have a more stable variance and minimize

the skewness through maximum likelihood estimation to estimate a transformation parameter, λ in the equation. There are two categories of PowerTransformer, one being the Box-Cox transform and the other is the Yeo-Johnson transform. The Box-Cox transform works when the data is strictly positive only. Yeo-Johnson on the other hand works on both positive and negative data. Since the 'Raw' column values in our dataset comprise of negative values, we proceeded with the Yeo-Johnson method. Equation 3.1 shows how the values are transformed after computing λ for each case. A value of λ=1 produces an identical transformation.

$$\psi(\lambda, y) = \begin{cases} \frac{((y+1)^{\lambda}-1)}{\lambda} & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ \frac{-[(-y+1)^{2-\lambda}-1]}{2-\lambda} & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases} \tag{3.1}$$

	Attention	Mediation	Raw	Delta	Theta	Alpha1	Alpha2	Beta1	Beta2	Gamma1	Gamma2
count	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04	1.153600e+04
mean	3.762979e-17	2.898745e-17	-1.487002e-16	1.237799e-15	-1.067877e-16	-1.382745e-15	-6.796074e-16	-2.292999e-16	2.115352e-16	-2.567487e-16	1.678673e-16
std	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00	1.000043e+00
min	-2.477877e+00	-2.674350e+00	-1.347181e+01	-2.097834e+00	-4.976113e+00	-5.826670e+00	-7.267628e+00	-6.630325e+00	-7.707574e+00	-6.343238e+00	-4.905864e+00
25%	-5.858827e-01	-5.855817e-01	-3.799671e-01	-7.230260e-01	-7.608685e-01	-6.925906e-01	-6.708024e-01	-6.813742e-01	-6.935767e-01	-6.976009e-01	-6.917530e-01
50%	7.586291e-02	-9.436762e-05	-4.212999e-02	9.084845e-02	2.436897e-02	-1.885729e-02	-1.546403e-02	-1.714219e-02	-1.247776e-02	-1.071044e-02	-1.113068e-02
75%	7.228952e-01	6.688158e-01	3.288921e-01	7.765107e-01	7.292363e-01	6.763264e-01	6.596010e-01	6.761714e-01	6.859686e-01	6.755395e-01	6.758626e-01
max	2.651521e+00	3.002341e+00	1.187797e+01	2.361486e+00	2.814991e+00	3.484405e+00	3.673392e+00	4.305362e+00	4.271470e+00	3.839354e+00	4.055517e+00

Figure 3.8: Statistical representation of input variables after being scaled with PowerTransformer

In figure 3.8, we can see that the values have been scaled as such that they have a standard deviation of 1, a reflection of the data having a normal distribution.

Moreover, the following figures from 3.9 to 3.19 show the before and after effect on the distribution and skewness of the data, through boxplots. The impact of scaling the data on the distribution and skewness for each column in the dataset has been visualized through boxplot diagrams.

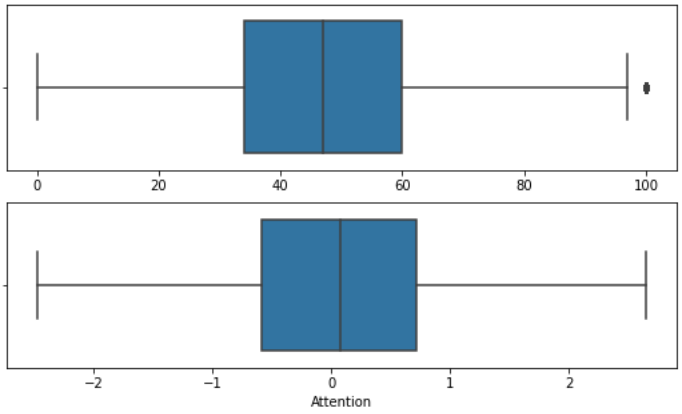


Figure 3.9: Boxplot of Attention feature before and after being scaled

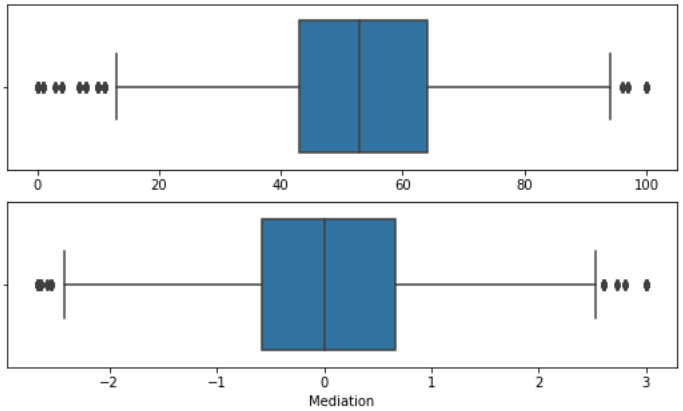


Figure 3.10: Boxplot of Mediation feature before and after being scaled

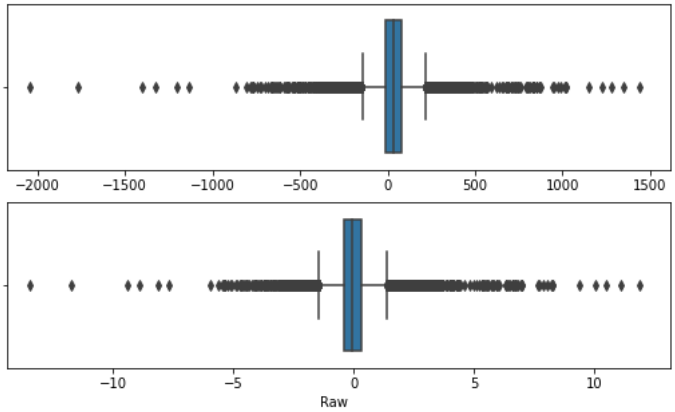


Figure 3.11: Boxplot of Raw feature before and after being scaled

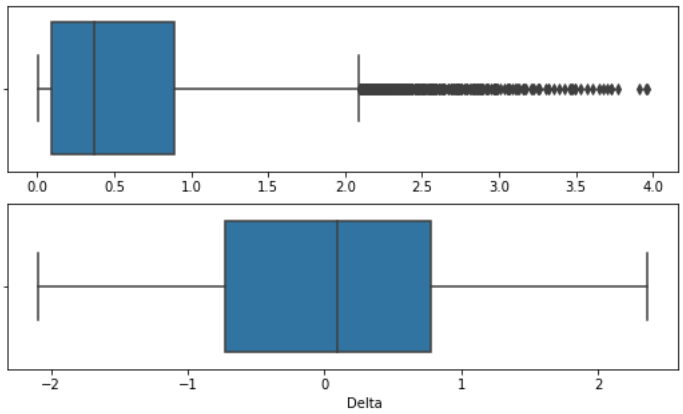


Figure 3.12: Boxplot of Delta feature before and after being scaled

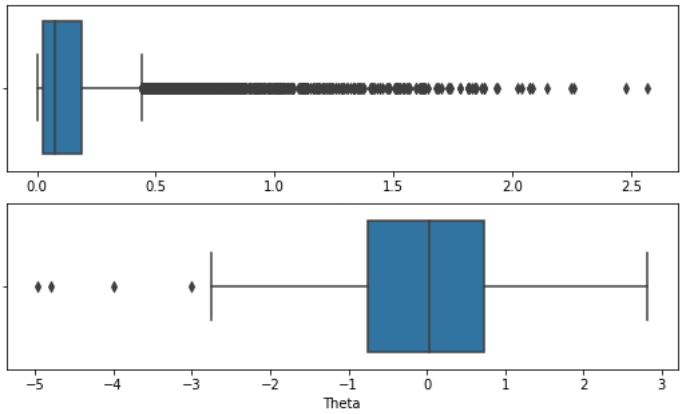


Figure 3.13: Boxplot of Theta feature before and after being scaled

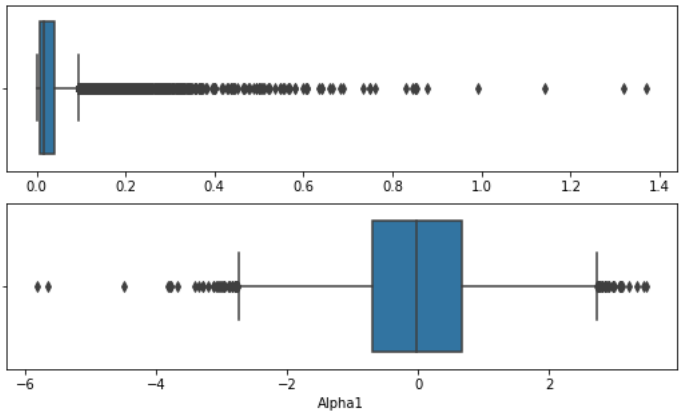


Figure 3.14: Boxplot of Alpha1 feature before and after being scaled

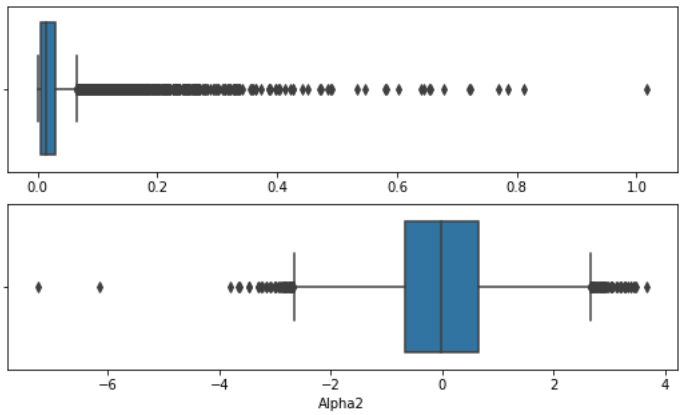


Figure 3.15: Boxplot of Alpha2 feature before and after being scaled

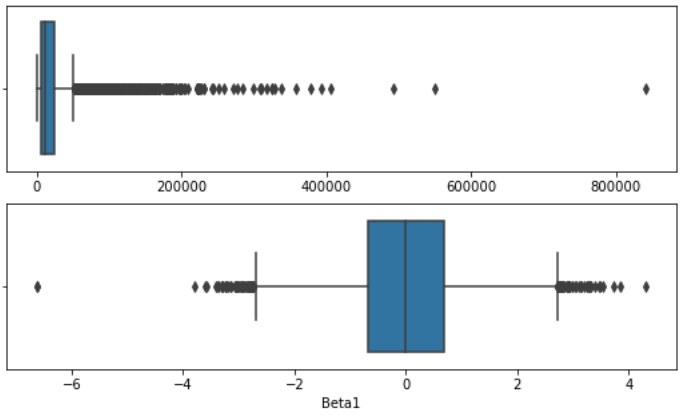


Figure 3.16: Boxplot of Beta1 feature before and after being scaled

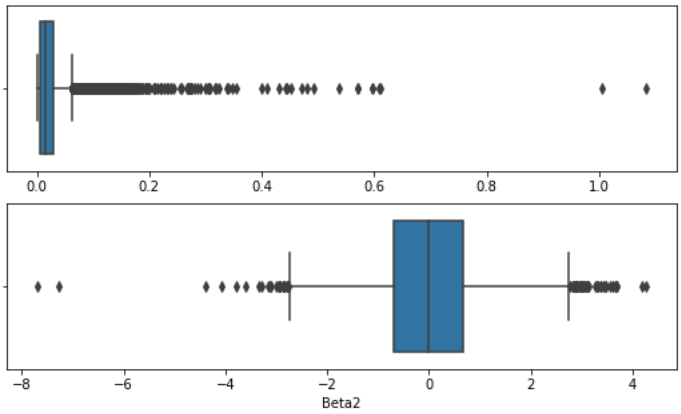


Figure 3.17: Boxplot of Beta2 feature before and after being scaled

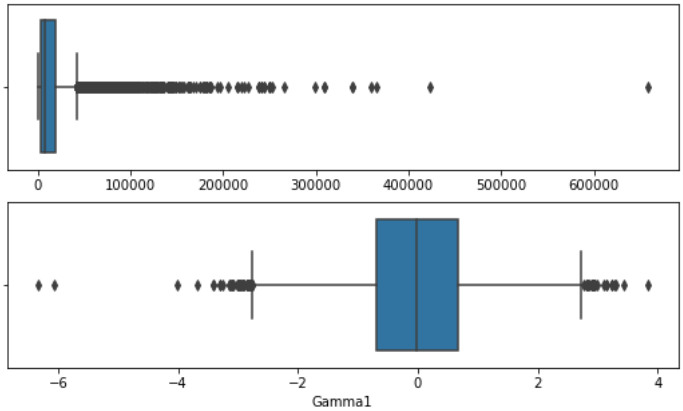


Figure 3.18: Boxplot of Gamma1 feature before and after being scaled

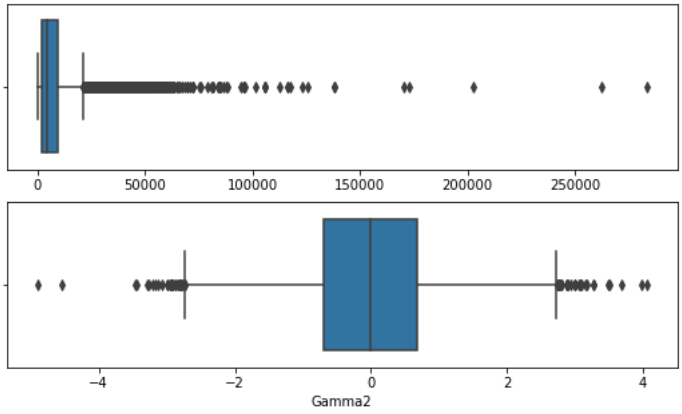


Figure 3.19: Boxplot of Gamma2 feature before and after being scaled

3.4.2 Feature Engineering to obtain Dataset 2

First, let us revisit the features of our initial dataset. Table 3.4 shows the features of the initial dataset.

Features
Subject ID
Video ID
Attention
Mediation
Raw
Delta
Theta
Alpha1
Alpha2
Beta1
Beta2
Gamma1
Gamma2
pre-definedlabel
user-definedlabeln

Table 3.4: Features in the initial dataset

The first step of feature engineering our dataset involved us in finding the proportion of students who were confused for each particular VideoID. It is found that most of the students were confused while watching a video of VideoID 8, while most of the students were least confused while watching a video of VideoID 9. Figure 3.20 shows the proportion of students who are confused for each VideoID.

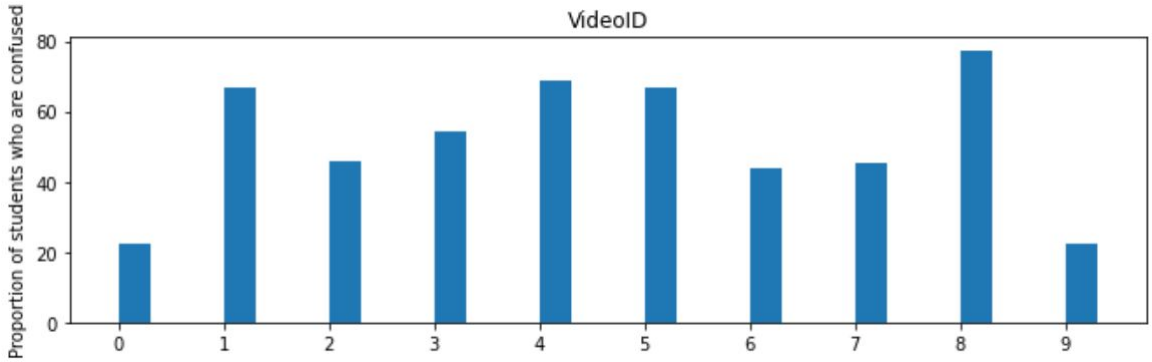


Figure 3.20: Proportion of students who are confused for each VideoID

Next step was creating a new feature called CourseID. We have then named the videos which were most confusing with CourseID of 400 level courses, while videos which were least confusing were named with CourseID of 100 level courses. Figure 3.21 shows the proportion of students who are confused for each CourseID.

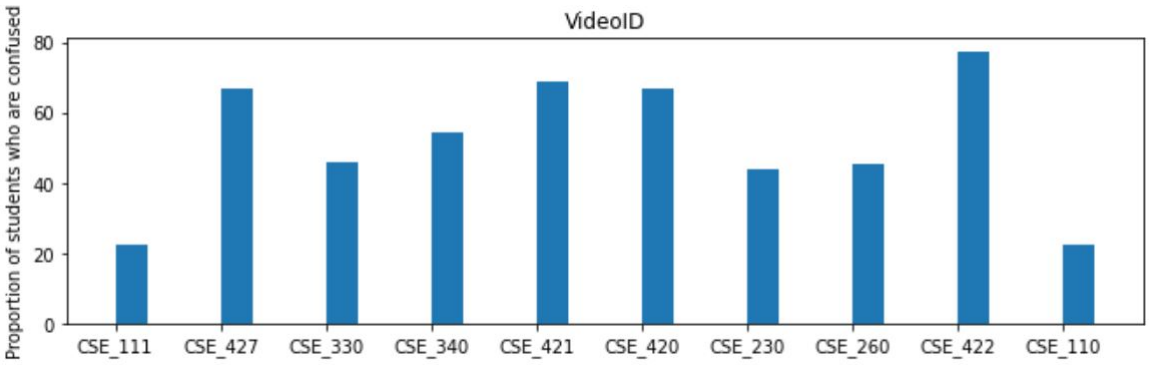


Figure 3.21: Proportion of students who are confused for each CourseID

Table 3.5 shows the features of our intermediate dataset after adding the CourseID column.

Features
Course ID
Attention
Mediation
Raw
Delta
Theta
Alpha1
Alpha2
Beta1
Beta2
Gamma1
Gamma2
user-definedlabeln

Table 3.5: Features in the intermediate dataset

We were now left with the task of encoding the categorical data represented by CourseID. We have made use of two methods for encoding the categorical data. The two methods which were used are:

1. Label encoding
2. One hot encoding

It was found that all the classifiers performed better while One Hot Encoding was used. Hence, we have used the dataset where One Hot Encoding was applied, for the rest of our research. Figure 3.22 shows the difference in accuracies when both One Hot Encoding and Label Encoding is applied to all of our classifiers.

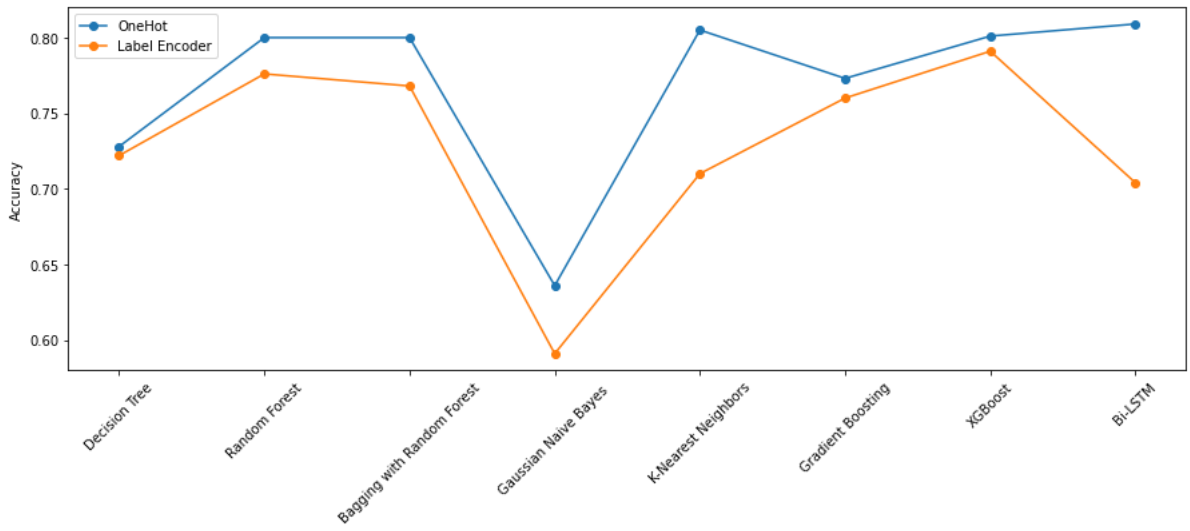


Figure 3.22: Accuracy comparison for One Hot Encoding Vs. LabelEncoding

Table 3.6 shows the features of our new and final Dataset 2, after applying One Hot Encoding.

Features
CSE_110
CSE_111
CSE_230
CSE_260
CSE_330
CSE_340
CSE_420
CSE_421
CSE_422
CSE_427
Attention
Mediation
Raw
Delta
Theta
Alpha1
Alpha2
Beta1
Beta2
Gamma1
Gamma2
user-definedlabeln

Table 3.6: Features in Dataset 2 including the various CourseIDs after One Hot Encoding

3.5 Limitations and Drawbacks of our Datasets

For our thesis work, we were supposed to collect the dataset ourselves with the help of the EMOTIV EPOC+ EEG headset available in our Computer Science and Engineering Department. However, due to the pandemic and our inability to physically attend our campus since March, we had to resort on finding a suitable dataset online. Unfortunately, our work of trying to determine the confusion level of students requires a very unique dataset, and such dataset cannot be readily found on the internet. Therefore we had to utilize the only resource that we could find online. The dataset that we have acquired from Kaggle has some limitations. It is a relatively smaller dataset. The dataset was made by collecting EEG signals from only 10 students, and it had only about 12,000 rows. Since the dataset was not a good one in particular, our models did not give satisfactory performance. Therefore, to increase the performance of our models, we had to resort to using an alternative approach which not only uses the EEG signals to measure the confusion level of students, but also uses the CourseID alongside with it. For this very purpose, we have synthesized Dataset 1 to form Dataset 2. However, Dataset 2 also has some limitations. Every time a new course is added, we have to feature engineers our way from Dataset 1 to Dataset 2 from the very start. It is because Dataset 2 is obtained through the means of one hot encoding, so if there is an addition of course, there should also be an addition of column indicating that course. With the addition of a new course, and subsequently, the addition of a new column, all the models should be retrained to learn how complex the new course is.

3.6 Feature Selection

In feature selection, features which have the most significant contribution in predicting the target variable, are selected either automatically or manually. By eliminating the irrelevant features, the following benefits can be achieved:

1. Reducing overfitting: less noise interference in decision making
2. Reduced complexity: without the irrelevant features, the algorithm will train faster and use less resources
3. Improved accuracy: less unnecessary data leads to improved modelling accuracy

Dimensionality reduction is one of the many feature selection techniques that may help to automatically eliminate irrelevant features.For our research, one of the dimension reductionality techniques we chose is Linear Discriminant Analysis or LDA. Another dimension reductionality technique we chose is Principal Component Analysis or PCA.

3.6.1 Principal Component Analysis

PCA or Principal Component Analysis is one of the dimension reductionality techniques that we have used on the two datasets. The goal when using PCA is to decrease the amount of features where there is also a possibility to increase the

accuracy. This is because smaller datasets are much more easier to explore and visualize. This also allows machine learning models to analyze the data much faster than before. As a result, PCA reduces the number of features whilst retaining as much of the information as possible.

Before we could apply PCA, we had to standardize the range of the features. This needed to be done so that every one of the features provide equally to the analysis. PCA is known to be quite sensitive to the variances of the features. For example, if there were features which had a larger range compared to other features, then those features would have a bigger impact on the results.

Hence, it would lead to a more biased result. This is why transforming the data to similar scales is necessary. As a result, we used the StandardScaler from the Scikit-learn library to transform the data for both the datasets. Equation 3.2 is used to standardize a value.

$$z = \frac{data\ value - mean\ of\ the\ feature}{standard\ deviation\ of\ the\ feature} \tag{3.2}$$

In equation 3.2, z represents the standardized value, *data value* is each of the input data of each of the features. The *mean* and *standard deviation* are of each of the individual features.

When applying PCA, certain computations take place. One of which is the covariance (COV) matrix computation. It can be seen that features are highly correlated to each other in a dataset to a point where they consist of inessential details. Then, the calculation of the COV matrix takes place to idenify these correlations. Since the two datasets are 11-dimensional and 21-dimensional, there are 11 features and 21 features, respectively. Hence, the covariance matrix has the shape 11×11 for one of the datasets and 21×21 for the other dataset. In the end, we are mostly concerned with the sign of the covariance. The positive and negative signs signify the direct and indirect proportionality of the features, respectively.

Next, to determine the *n_components* or principal components of the data, the eigenvalues and the eigenvectors are calculated using COV matrix. The principal components are the new features that are assembled from the linear combinations of the original features. This is constructed in such a manner that the principal components are uncorrelated unlike the original features. However, most of the information of the original features are squeezed or compressed within the first few components. Since we have 11-dimensional and 21-dimensional data, we acquired 11 and 21 principal components, respectively. The maximum amount of information can be seen in the first component, then the remaining maximum information on the second component and so on for both the datasets. Figures 3.23 and 3.24 illustrates the contribution of the components to the variance for Dataset 1 and Dataset 2, respectively.

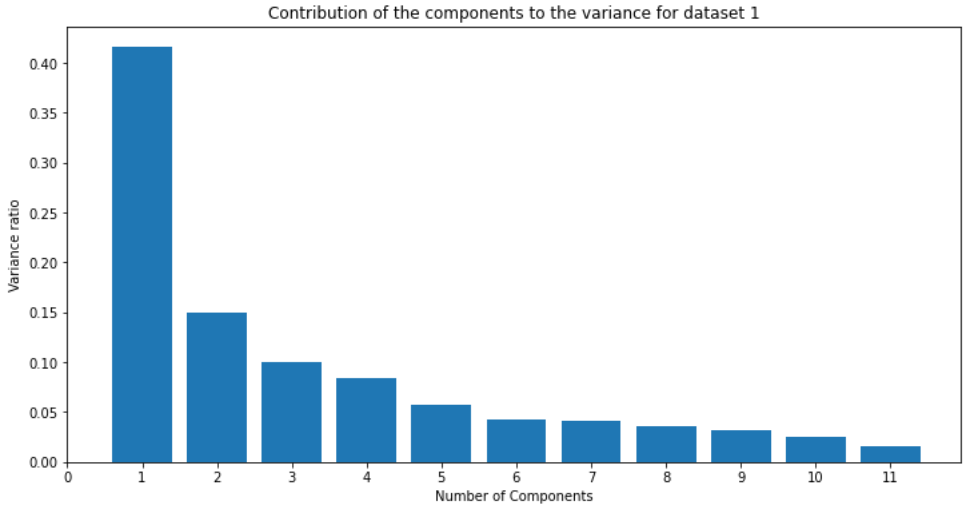


Figure 3.23: Contribution of the components to the variance for Dataset 1

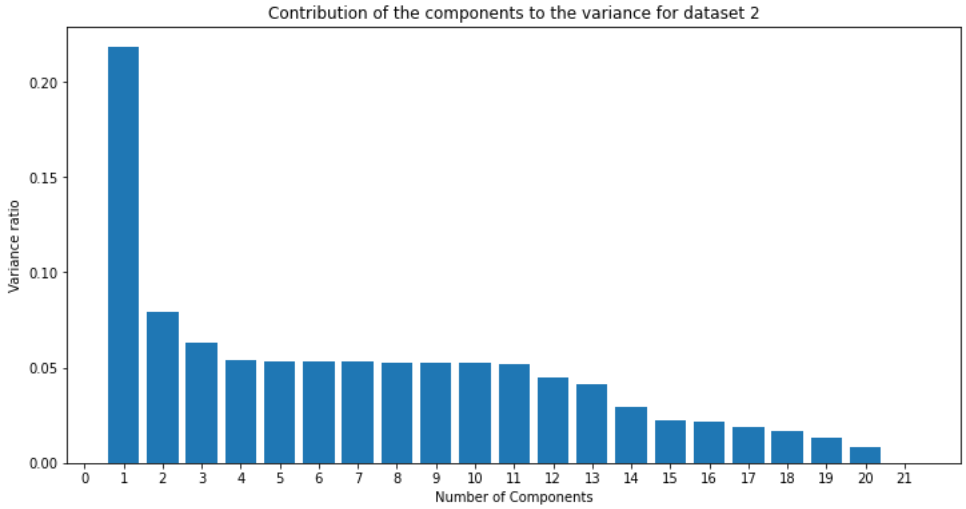


Figure 3.24: Contribution of the components to the variance for Dataset 2

The arrangement of the information in this way allowed us to decrease dimensionality without losing much information. We then proceeded to eliminate the components with low information and use the remaining components for the two datasets as the new features for the machine learning models. One point to remember is that the principal components are less interpretable and don't have any definite meaning when compared to the original features. If we were to think of the principal components in a different approach, they indicate and describe the spread of the data in different positions. These are the lines that express most information of the data. The relationship that exists between the variance and information here, is that, the greater the variance carried by a line, the greater the spread of the data points along it, and the greater the spread of the data points along a line, the more the information it has.

The eigenvalue and the eigenvector come in pairs and their amount is equal to the dimension of the dataset. So, for the 11-dimensional dataset, there are 11

eigenvectors and 11 corresponding eigenvalues. Similarly, there are 21 eigenvectors and 21 corresponding eigenvalues for the 21-dimensional dataset. The eigenvectors of the covariance matrix are the directions of the axes where there is the most variance or the most information. Those are called the principal components. For each principal component, the details revealed by the spread of the data is achieved through the coefficients, which are the eigenvalues that are allocated to the eigenvectors. So, the eigenvectors are sorted from highest to lowest which results in the principal component with the most variance coming in first and so on. The percentage of variance associated with each of the components are calculated by the eigenvalue of each of the components divided by the sum of the eigenvalues [20].

When we applied PCA on the two datasets, we decided to keep 80 percent of the variance. As a result, the number of principal components for Dataset 1 were 5 and 11 for Dataset 2. The cutoff threshold and the cumulative variance for the two datasets are shown in the figures 3.25 and 3.26.

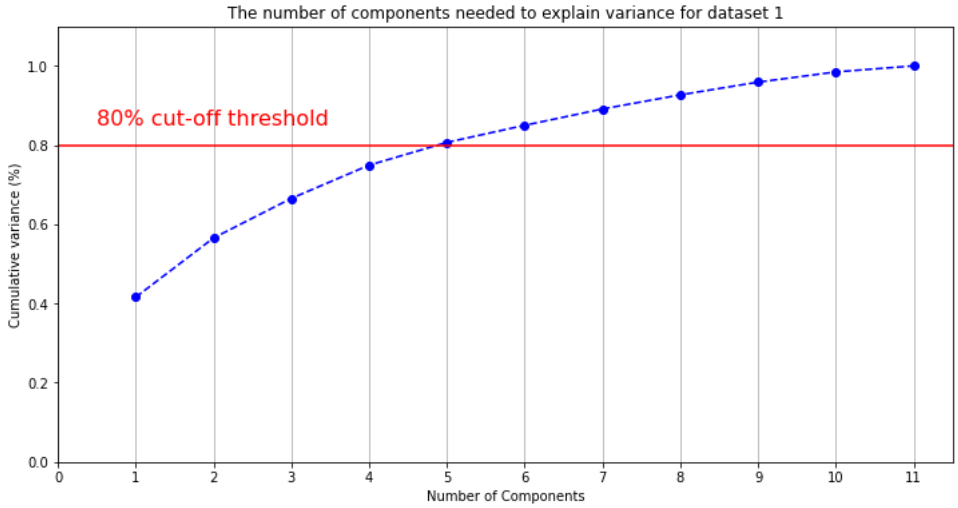


Figure 3.25: Number of components needed to explain the variance for Dataset 1

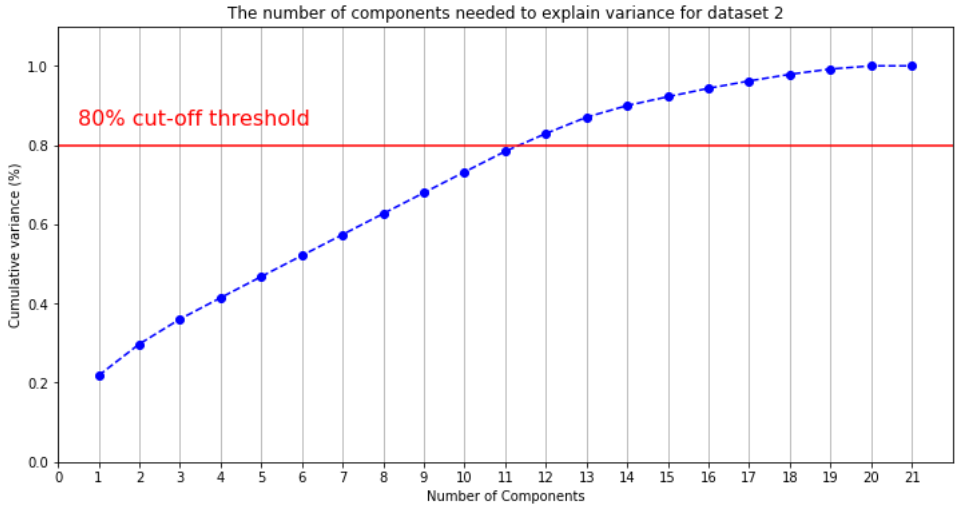


Figure 3.26: Number of components needed to explain the variance for Dataset 2

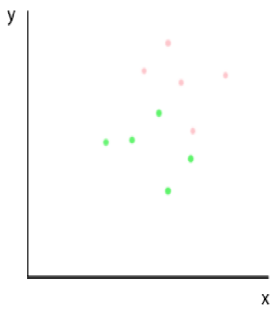
This resulted in lower accuracy when the new features for the two datasets were used as input for the machine learning models. However, we did decrease the dimension of the two datasets and analyzing for both the datasets were much faster by the model. Since we did not achieve our goal to increase accuracy through PCA, we decided against adding it to our final results.

3.6.2 Linear Discriminant Analysis

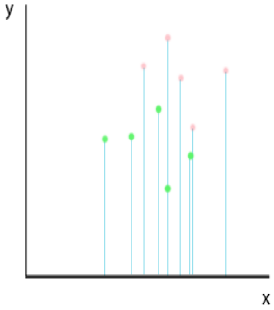
Linear Discriminant Analysis or LDA is another one of the dimension reduction techniques that we have used on the two datasets. To get a general idea on how LDA works, we should consider an example. If certain data points (two classes) were plotted on a graph considering two features on the x and y axis, one way to reduce the number of dimensions could be to project all the data points to one of the axes. Figures 3.27(a) and 3.27(b) shows data points being projected to one axis.

A problem with this approach is that any valuable information from the other feature is lost. It is also possible that there is no clear way to separate the two classes when projected. Figure 3.27(c) shows the loss of information that takes place after data points are projected to one axis.

However, LDA does provide a solution to this issue. It uses information from both the features to create a new axis which maximizes the distance between the means of the two classes. It also minimizes the variation (scatter) within each class. Figures 3.28(a), 3.28(b) and 3.28(c) shows the data points being successfully projected to a new axis and that there is no overlapping between the two classes.



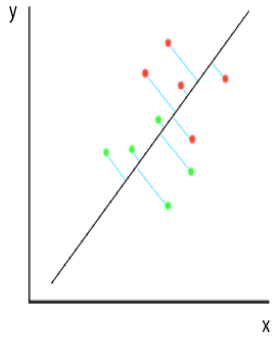
(a) Example of two classes plotted on a graph with two features on the axes



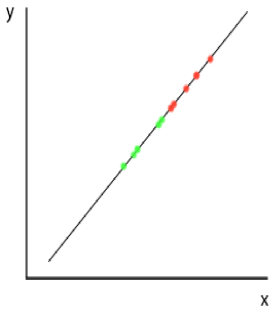
(b) The two classes being projected on one axis

(c) The overlapping of two classes

Figure 3.27: The projection of data points to one axis leading to information loss



(a) The two classes being projected on a new axis



(b) Successful projection of the two classes

(c) No overlapping between two classes

Figure 3.28: A new axis is created where the two classes are projected with no overlapping after LDA is applied

LDA is also applicable for two or more classes. A simplified ratio of the process is represented by equation 3.3.

$$\frac{\eta_1 - \eta_2}{s_1^2 - s_2^2} \tag{3.3}$$

In equation 3.3, η_1 and η_2 represent the means of the two classes. The s_1^2 and s_2^2 are the variances or scatter of the two classes. This also applies to the two datasets we have as we are trying to classify between two classes, which are confused or not confused [21].

LDA emphasizes on projecting the features of the two datasets from a higher dimension space to a lower dimensional space. The first step is to calculate the separability between our two classes.

This means calculating the distance between the means of the two classes and maximizing the distance. This is also called the between-class variance. The between-class variance is calculated by using equation 3.4.

$$S_b = \sum_{i=1}^{g} N_i(\overline{x_i} - \overline{x})(\overline{x_i} - \overline{x})^T \tag{3.4}$$

In equation 3.4, S_b represents the between-class scatter matrix, the N_i represents the sample size of class i, $\overline{x}$ represents the overall mean and $\overline{x_i}$ represents the sample mean of class i.

Secondly, the distance between the mean and sample of each class is calculated. This is called the within-class variance. The within-class variance is calculated by using equation 3.5.

$$S_w = \sum_{i=1}^{g}(N_i - 1)S_i = \sum_{i=1}^{g}\sum_{j=1}^{N_i}(x_{i,j} - \overline{x})(x_{i,j} - \overline{x})^T \tag{3.5}$$

In equation 3.5, S_w represents the within-class scatter matrix, N_i represents the sample size, S_i and $\sum_{j=1}^{N_i}(x_{i,j} - \overline{x})(x_{i,j} - \overline{x})^T$ represent the scatter matrix for class i.

Finally, the lower dimensional space is constructed which maximizes the between-class variance and minimizes the within-class variance. This is known as Fisher's Criterion [22].

When applying LDA in code, we imported LDA from the Scikit-learn library. Afterwards, we constructed a method to get the appropriate *n_components* to use when applying LDA to transform the features of both the datasets. The *n_component* to be used was 1 for the two datasets. Then, after transforming the data and using the machine learning models, we saw that the accuracy has decreased significantly for both the datasets in all models. This led us to the conclusion that LDA did not manage to separate the confused and not confused labels or classes in either of the two datasets as accurately as expected. Hence, this

approach was not beneficial in our research.

3.7 t-Distributed Stochastic Neighbor Embedding (t-SNE) for Visualization

3.7.1 What is t-SNE?

Visualizing high dimensional data is one of the challenges faced in machine learning. It is quite difficult for us to comprehend and analyze data which have dimensions greater than 2 or 3. Therefore, there is a requirement for a method which brings data to a low dimensional space from a high dimension, which in turn, would make analyzing the data much simpler. One of the methods which allow us to map data from high dimensional space to low dimensional space is called t-Distributed Stochastic Neighbor Embedding (t-SNE). Applying t-SNE to a high dimensional space dataset gives us the intuition of how the data is arranged in the high dimensional space. If there are n number of features such that F = $\{f_1, f_2,..., f_n\}$, what t-SNE does is that it converts F into D, such that D = $\{d_1, d_2,..., d_m\}$. Usually d_m is much smaller than f_n. Generally, the value of d_m is 2 or 3. It is because visualizing data in 2 or 3 dimensional space is much easier to analyze than visualizing data in dimensions which are higher than 3.

To understand how t-SNE works, let us consider the figure 3.29.

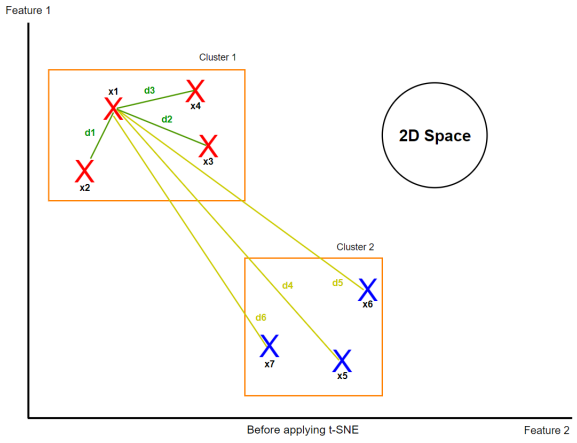


Figure 3.29: Visualization of data before applying t-SNE

In figure 3.29, x1 through x7 are data points. In this particular case, each data point can be plotted using 2 values, feature 1 and feature 2. Hence, these data points can be represented using a 2D space. If we apply t-SNE to these data, we can represent the above information using a lower dimensional space, so that analyzing the data can be made easier. To understand the information after applying t-SNE, we first need to understand the concept of 'Neighbor' in t-SNE. If we consider the data point x1 and if we set the perplexity parameter in the code to be 3, the

neighbor of x1 would be x2, x3 and x4. The neighbor of x1 is said to be x2, x3 and x4 because x2, x3 and x4 have the 3 minimum distances from x1. Similarly, if we want to find the neighbors of x5 and set the perplexity parameter to be 2, the neighbors of x5 would be x6 and x7. So what t-SNE does is that it preserves the distances between the neighbors from one dimension to another. If we look at the above diagram, we can see that x1, x2, x3 and x4 makes a cluster, while x5, x6 and x7 makes another cluster. If we apply t-SNE to this dataset, what t-SNE would do is that it would preserve the distance between neighbors residing inside a particular cluster. One thing that t-SNE does not promise is that it may or may not preserve the distance between each cluster.

Now let us see how the information is presented if we apply t-SNE to the previous dataset. Figure 3.30 represent the data after applying t-SNE.

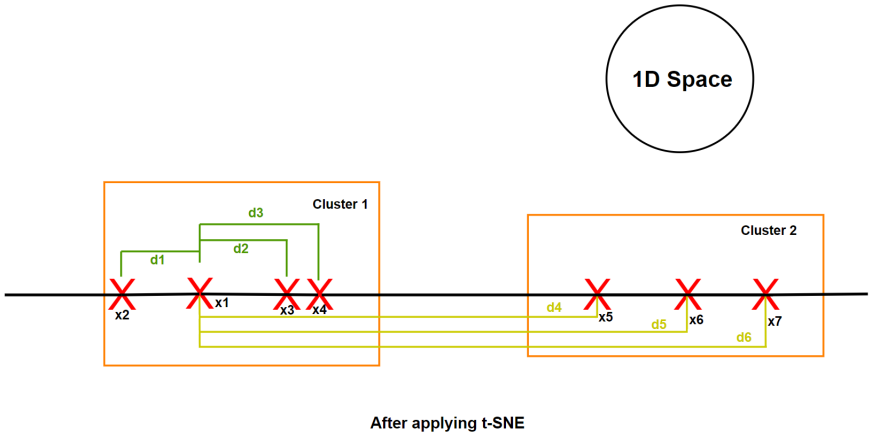


Figure 3.30: Visualization of data after applying t-SNE

In figure 3.30, we can see that the distances between the neighbors in cluster 1 and cluster 2 are preserved. To understand it better, let us consider the data point x1. We can see that the distances d1, d2 and d3 are roughly the same both before and after applying t-SNE. However, the distances d4, d5 and d6 are not the same before and after applying t-SNE, as t-SNE does not promise us that it would preserve the distances between the clusters. It only promises us that it would preserve the distances between neighbors in a particular cluster.

3.7.2 Applying t-SNE to our Datasets

Dataset 1

A lot of information can be obtained by having a look at the data points after applying t-SNE. It is known that machine learning models would work exceptionally well if the clusters are clearly distinguishable even after applying t-SNE. If the clusters are clearly distinguishable after applying t-SNE, it implies that the high dimensional data contains sufficient information to map labels to clusters. However,

for a particular dataset, if t-SNE results show overlapping of labels, it means that it is very hard to map each label to each cluster. Therefore, for such cases, machine learning models would not work particularly well. The result after applying t-SNE to Dataset 1 is shown in figure 3.31.

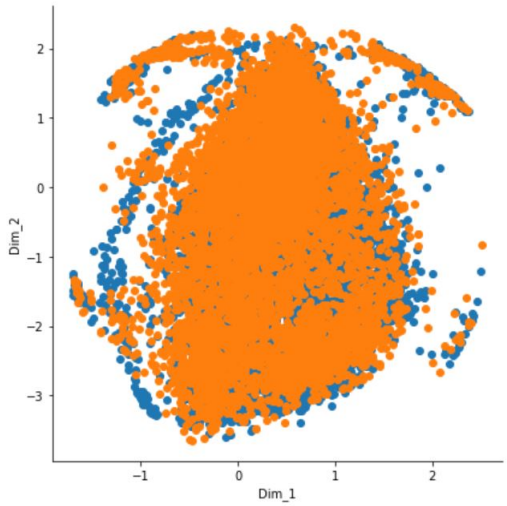


Figure 3.31: Visualization of Dataset 1 after applying t-SNE

It can be seen that the clusters of label 0 and 1 overlap with one another. Therefore, we can come to a conclusion that applying machine learning model to our dataset would not give satisfactory results.

Dataset 2

Not much is changed after applying t-SNE to Dataset 2. Even though there is a significant increase in accuracy of models while using Dataset 2, overlapping is still seen in the t-SNE implementation. The result after applying t-SNE to Dataset 2 is shown in figure 3.32.

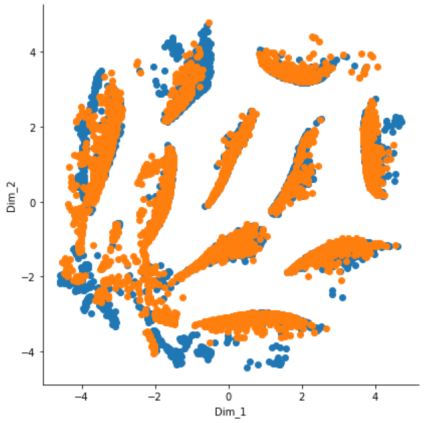


Figure 3.32: Visualization of Dataset 2 after applying t-SNE

3.8 Heatmap

3.8.1 Correlation heatmap of Dataset 1:

A correlation heat map shows the correlation between all of the features. The cells in the heat map are colored in a way so that it can be easier to interpret and visualize the correlation between the features. Figure 3.33 shows the correlation heat map that is generated for dataset 1. Here, darker color represents strong positive correlation between features, while lighter color represents strong negative correlations. The figure shows us that there is a strong positive correlation between (i) Beta2 and Gamma1, (ii) Gamma2 and Gamma1. Also, moderate positive correlation can be seen between (i) Alpha2 and Beta1, (ii) Attention and Mediation, (iii) Alpha1 and Alpha2, Beta1, (iv) Theta and Alpha1, Alpha2, Beta1. No strong or moderate negative correlations can be found.

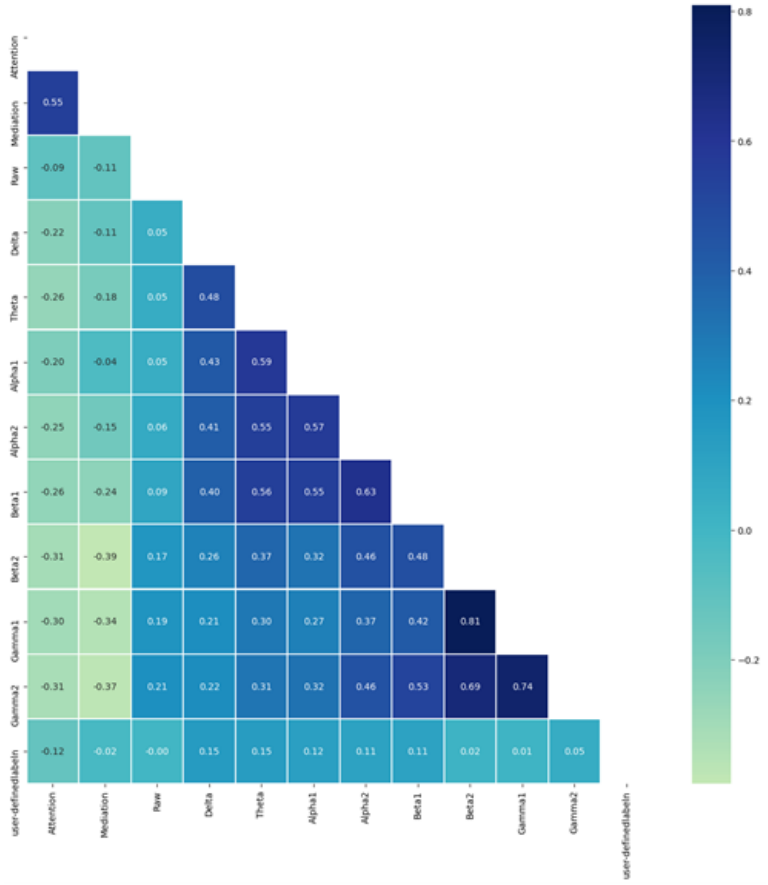


Figure 3.33: Heatmap for Dataset 1

3.8.2 Correlation heatmap of Dataset 2:

Figure 3.34 shows the correlation heat map for dataset 2. While the information represented by figure 3.34 is almost the same as figure 3.33, we can see that there is no correlation between the EEG brainwave features and the courses with various CourseID. We had hoped to see some correlation between highly confusing course like CSE427 with the features of the datasets, and wanted to draw some conclusion

of how complex level of courses affect the EEG brainwave values like alpha, beta, gamma. Unfortunately, no such correlation existed for our given dataset. We hope to study more deeply into this matter in the future, when we will be having access to EEG head gear, and when we can extract primary data ourselves, instead taking the ones which are available online.

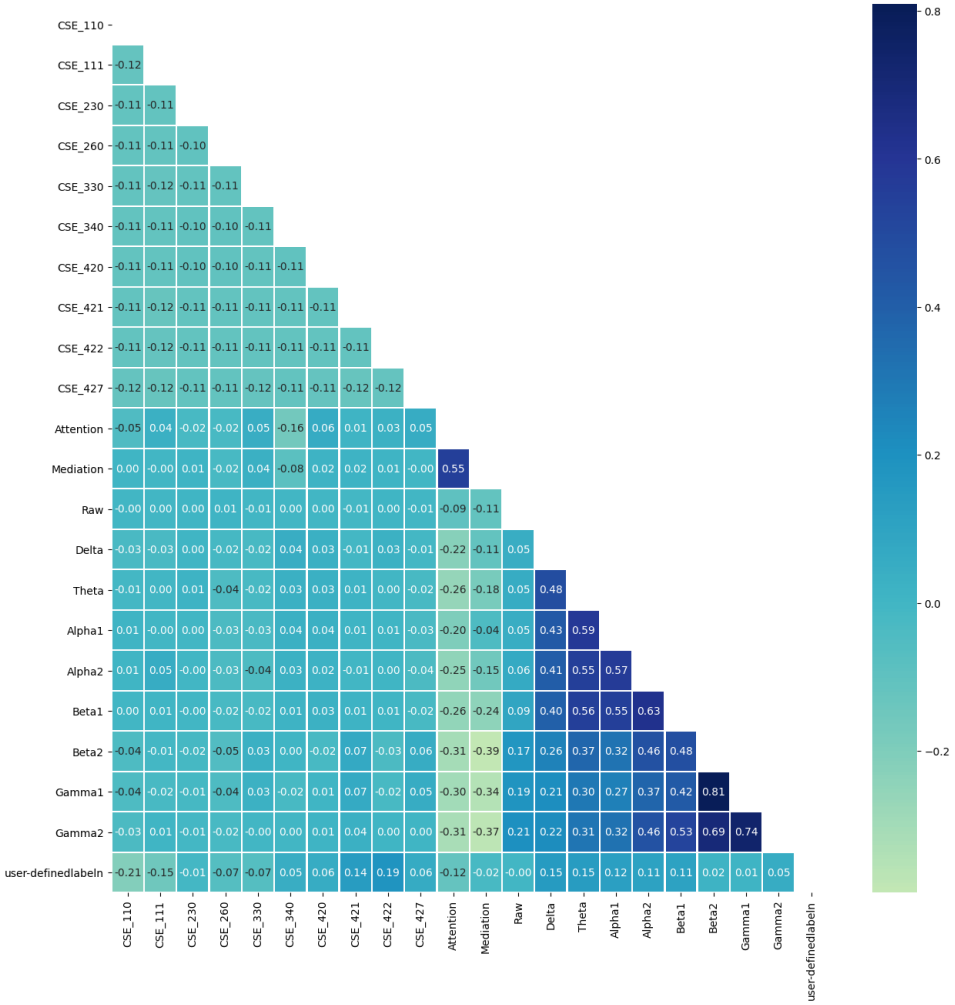


Figure 3.34: Heatmap for Dataset 2

Chapter 4

Model Selection and Result Analysis

4.1 Machine Learning

The concept of Machine Learning originates from ground breaking research by Arthur Lee Samuel, who defined it as "Machine Learning is a field of study that gives computers the ability to learn without explicitly being programmed" [23]. It can be considered as a subset of Artificial Intelligence, where the algorithms create a model which without being explicitly programmed, by taking advantage of patterns on a sample data provided to the model. This sample data is commonly referred to as the training data. With the aid of complex mathematical and statistical models, the Machine Learning model then performs its given task which could be but is not limited to delivering a prediction or a decision. Another great contributor in the history of Artificial Intelligence and Machine Learning, Tom Michael Mitchell defined Machine Learning to be, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" [24]. Whenever it is no longer feasible to explicitly program a computer with an algorithm to perform a task, such a situation is an example of where the task can be performed with the aid of Machine Learning. Machine Learning has a wide variety of applications such as image recognition, speech recognition, medical diagnosis and traffic prediction. One of Machine Learning's many subsets is computational statistics where the model can deliver predictions. In our research, we implemented a machine learning approach to predict the confusion state of a student from their EEG signals.

4.1.1 Supervised Learning

Machine Learning consists of various categories. These are supervised learning, unsupervised learning, reinforcement learning etc. Since the problem we are trying to solve involves providing input that results in an output, we have decided to use supervised learning. The target of the supervised learning algorithm is to use the dataset to produce a model that takes a number of features as input and outputs information that allows deducing a label for those features [25]. In our case, for Dataset 1 we have various brain waves as the input features and a total of 9608 training examples, for Dataset 2 we have the various brain waves as well as tag of

a particular video as the input and a similar total of 9608 training examples. For both the datasets, the output is a label of whether a student is confused or not. The training data can be represented as a matrix while each training example is an array or vector. Finally, the test data is given as the input features to the trained model which then predicts an output label of confused or not confused. Hence, this is the way supervised learning predicts the outcome of a problem.

4.2 Models

4.2.1 Decision Tree

Decision tree is a type of machine learning model which, by simplest description, is a divide-and-conquer approach to classification or prediction/ regression. When used in large databases, decision trees are able to discover features and extract patterns that are of importance for discrimination and predictive modeling. Common usages of decision tree models include variable selection, assessing relative importance of variables, handling of missing values, prediction, data manipulation etc [26]. The advantage of this model over most others is the interpretability of the constructed model. It consists of a flowchart like tree structure in which each internal (non-leaf) node is labeled with an input feature. Each internal edge has two edges, labeled with each of the possible values of the features, which also depicts the outcome of each test computed on an attribute in the internal nodes [27]. Lastly, each leaf of the tree is labeled with a class or probability distribution over the classes. This tree is learned by the source dataset into subsets based on an attribute value test. The process is repeated through recursive partitioning on each subset. The recursive is considered to be completed when there is no node that can be splitted that significantly adds value to the prediction. In the decision tree, an instance is classified by sorting them down the tree from the root to a particular leaf node, which provides the classifier Feature importance of the Random Forest Classifier is given below action of that instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by that particular node, then moving down the tree branch corresponding to the attribute's value. This process is repeated for the subtree rooted at the new node. Figure 4.1 shows the feature importance of the Decision Tree.

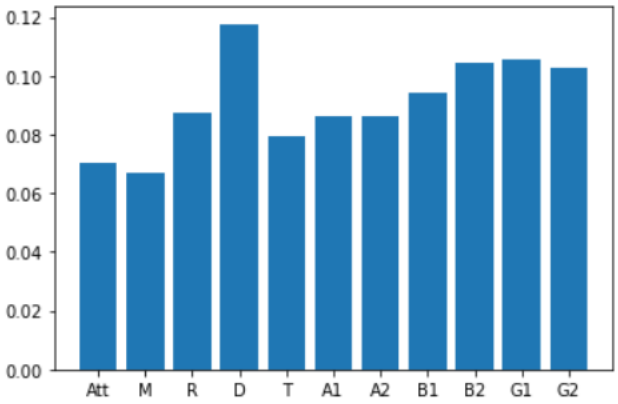


Figure 4.1: Feature importance of the Decision Tree

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 950 & 644 \\ 630 & 979 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.602, 0.603, 0.608, 0.606, respectively.

Figure 4.2 shows ROC curve for Dataset 1 when using Decision Tree Classifier.

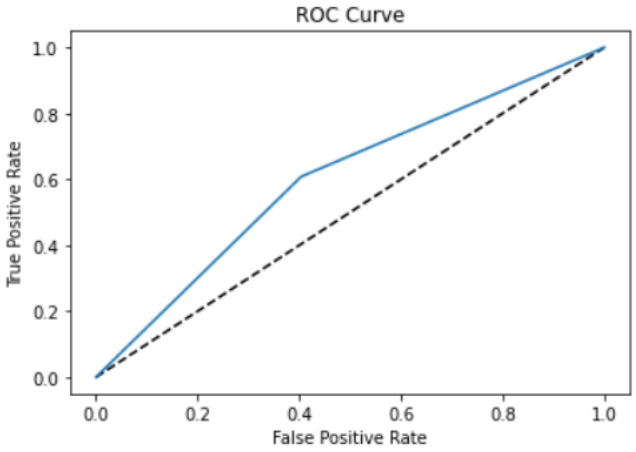


Figure 4.2: ROC curve of Decision Tree Classifier for Dataset 1

For the ROC curve in figure 4.2, AUC score of 0.602 was obtained which means that Decision Tree has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1128 & 466 \\ 404 & 1205 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.728, 0.721, 0.749, 0.735, respectively.

Figure 4.3 shows ROC curve for Dataset 2 when using Decision Tree Classifier.

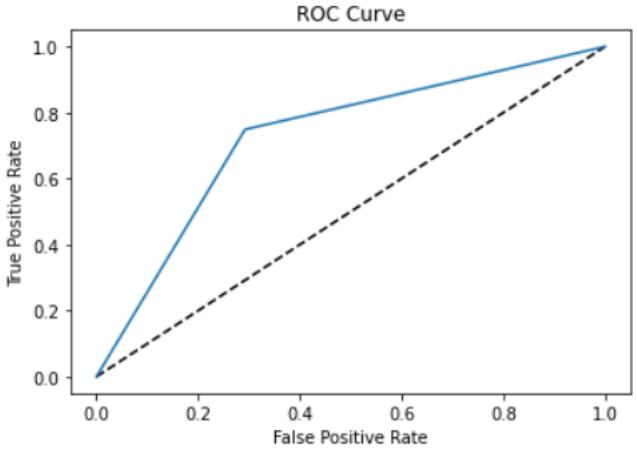


Figure 4.3: ROC curve of Decision Tree Classifier for Dataset 2

For the ROC curve in figure 4.3, AUC score of 0.728 was obtained which means that Decision Tree has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.2.2 Random Forest

Random Forest is a supervised learning algorithm that produces fairly acceptable results even without hyperparameter tuning. It has been used in various peer-reviewed journals and conference papers. In [28], the authors had applied the random forest on EEG data to classify the human mental state. They managed to identify concentration and meditation with an accuracy of 75%. In [29], the authors used the random forest classifier for sleep stage identification and extracted features from the time-frequency representation of the subject's EEG signal using Renyi's entropy. They achieved promising results with an accuracy of 83%. If we were to put it simply, a random forest creates a number of decision trees and merges them together to achieve a more accurate prediction compared with a single decision tree. We have already explained how a decision tree works. However, the random forests are a modification of the decision tree algorithm making it more robust. Now, there are a number of steps that take place in a random forest classifier. For example, the Dataset 1 and 2 have 9608 training examples, the algorithm repeatedly samples subsets of the training data of size m1 and m2 (separate variables for the two datasets), which is less than 9608. Since the training examples for the two datasets have 11 and 21 features,

the subset of them have n1 and n2 features which are less than the 11 and 21 features, respectively. So, none of the decision trees see the full training data. The decision trees, or the n_estimators, are trained on a different set of n1 and n2 features as well as m1 and m2 training examples for Dataset 1 and 2, respectively. We have chosen 93 decision trees, or n_estimators. Since there are 93 decision trees, there will be 93 predictions which combine to form the overall prediction of the random forest. Since our problem is a classification task, the final outcome of whether a student is confused or not is decided through majority voting [30].

For our classification task, a train-test split of 75 percent for train and 25 percent for test was made. As it has been stated previously, the n_estimators or decision trees have been set to 93. Figure 4.4 shows the feature importance of the Random Forest Classifier.

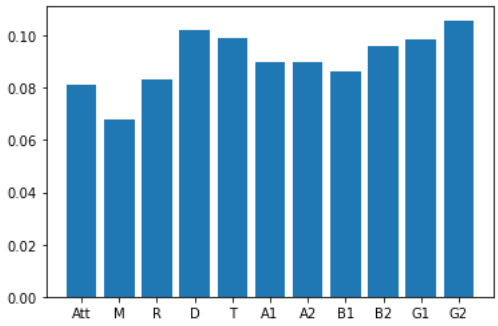


Figure 4.4: Feature importance of the Random Forest Classifier

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 1032 & 562 \\ 496 & 1113 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.670, 0.664, 0.692, 0.678, respectively.

Figure 4.5 shows ROC curve for Dataset 1 when using Random Forest Classifier.

For the ROC curve in figure 4.5, AUC score of 0.734 was obtained which means that Random Forest has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

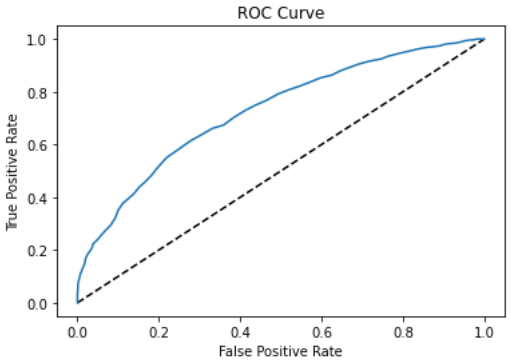


Figure 4.5: ROC curve of Random Forest Classifier for Dataset 1

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1198 & 396 \\ 244 & 1365 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.800, 0.775, 0.848, 0.810, respectively.

Figure 4.6 shows ROC curve for Dataset 2 when using Random Forest Classifier.

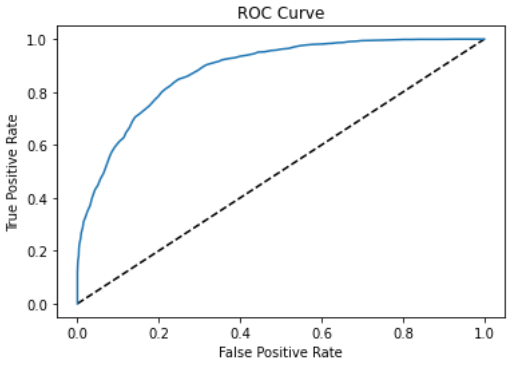


Figure 4.6: ROC curve of Random Forest Classifier for Dataset 2

For the ROC curve in figure 4.6, AUC score of 0.881 was obtained which means that Random Forest has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.2.3 Bagging with Random Forest

A Bagging classifier is an ensemble meta-estimator which means it relies on a collection of classifiers that is fitted on random subsets of the original dataset where the individual predictions are aggregated either by voting or averaging

to form a final prediction [31]. The training set for each classifier is generated randomly by drawing, with replacement, N examples from the original training dataset, where N is the size of the training dataset, in our case, N will be 9608. There will be cases where many of the original data may be repeated in the resulting training sets while others could be dropped. Hence, each individual classifier in the ensemble is generated with a different random sampling of the training set [32].

For our classification task, a train-test split of 75 percent for train and 25 percent for test was made as it has already been stated. The base estimator that we have chosen for our Bagging classifier is Random Forest Classifier where the number of decision trees are set to 93. The number of base estimators for bagging has been set to 20. As a result, 20 random forest classifiers are generated with different random samples of the original training data for each of the two datasets.

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 1048 & 546 \\ 502 & 1107 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.673, 0.670, 0.688, 0.678, respectively.

Figure 4.7 shows ROC curve for Dataset 1 when using Bagging with Random Forest Classifier.

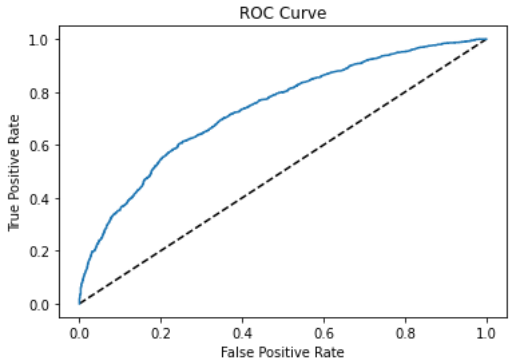


Figure 4.7: ROC curve of Bagging with Random Forest Classifier for Dataset 1

For the ROC curve in figure 4.7, AUC score of 0.736 was obtained which means that Bagging with Random Forest has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1188 & 406 \\ 235 & 1374 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.800, 0.772, 0.854, 0.811, respectively.

Figure 4.8 shows ROC curve for Dataset 2 when using Bagging with Random Forest Classifier.

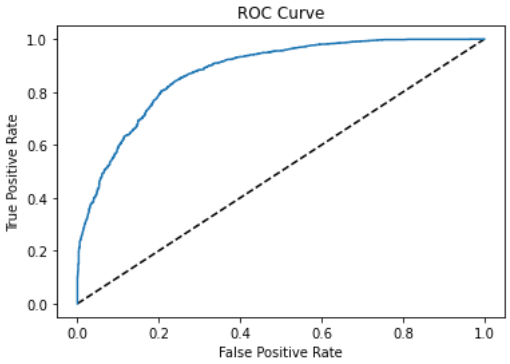


Figure 4.8: ROC curve of Bagging with Random Forest Classifier for Dataset 2

For the ROC curve in figure 4.8, AUC score of 0.876 was obtained which means that Bagging with Random Forest has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.2.4 Gaussian Naive Bayes

Gaussian Naive Bayes is a supervised machine learning classification algorithm. It is a variant of Naive Bayes but follows Gaussian normal distribution while working with continuous data. Naive Bayes is a useful classification technique of supervised machine learning classification or data mining algorithms based on the Bayes theorem, a very simple classification technique with high functionality that allows us to calculate conditional probability. Naive Bayes Classifiers can implement simple and complex classification problems since it is not necessary to have a large training data to approximate the parameters required for classifying the labels. They are mainly found to be very helpful when inputs have high dimensionality. These classifiers make the assumption that the value of the features are not affected by the value of others. Hence, Naive bayes classifiers are efficient in training supervised learning data. As the Gaussian Naive Bayes Classifier is based on this, it shares these properties. As it complies with attributes and models with continuous values that follow Gaussian distribution, it can be ideal to be applied to real life situations. Assuming that Gaussian distribution describes data without any co-variance between dimensions is a way to generate a basic model. The distribution can be described by fitting the model through the calculation of the standard deviation and mean of the data points associated with each label. In

the paper [33], the authors used the Gaussian Naive Bayes algorithm for cancer classification. Their evaluation results achieved a solid 98% accuracy in predicting breast cancer and 90% predicting lung cancer. In the paper [34], the authors apply Genetic Algorithms and Gaussian Naive Bayes Classifier to select an optimal set of attributes and classify different cognitive states, respectively, achieving an accuracy of 96.46%.

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 926 & 668 \\ 640 & 969 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.592, 0.592, 0.602, 0.597, respectively.

Figure 4.9 shows ROC curve for Dataset 1 when using Gaussian Naive Bayes Classifier.

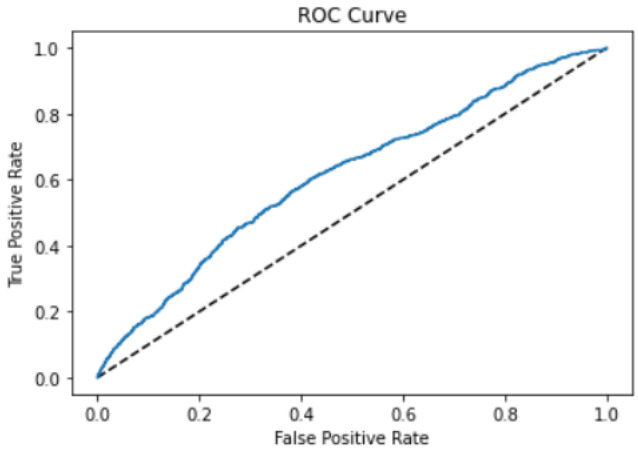


Figure 4.9: ROC curve of Gaussian Naive Bayes Classifier for Dataset 1

For the ROC curve in figure 4.9, AUC score of 0.613 was obtained which means that Gaussian Naïve Bayes has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 996 & 598 \\ 567 & 1042 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.636, 0.635, 0.648, 0.641, respectively.

Figure 4.10 shows ROC curve for Dataset 2 when using Gaussian Naive Bayes Classifier.

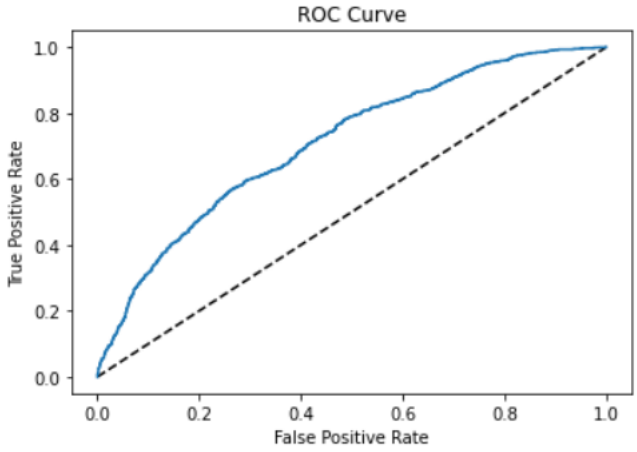


Figure 4.10: ROC curve of Gaussian Naive Bayes Classifier for Dataset 2

For the ROC curve in figure 4.10, AUC score of 0.710 was obtained which means that Gaussian Naïve Bayes has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.2.5 K-Nearest Neighbors

K Nearest Neighbour is a classification algorithm widely used in machine learning for its simple and straightforward implementation. In simple terms, it is used to classify a particular data point by evaluating its distance from its k nearest neighbours, where k determines the number of neighbours it would take into consideration [35]. For example, let us consider that there are 2 clusters, A and B, and there is a single data point, X, which needs to be classified as a member of either A or B and that the value of k in this particular situation is 1. We would assign X to a group by evaluating the single nearest neighbour of X. If that specific neighbour is a member of group A, X would be classified as a member of group A, and vice versa [36]. Now, if the value of k is 2, the algorithm would take into consideration of 2 nearest neighbours of X. If the first neighbour is a member of group A, and the second neighbour is a member of group B, the distance between X and the 2 neighbours would be evaluated. The group of the neighbour, which has the shortest distance from X, would be assigned as the group of X.

For our classification, we ran our KNN model 50 times. In each iteration, we increment the value of K by 1 to see which value of K, ranging from 1 to 50, gave the best accuracy. A train-test split of 75 percent for train and 25 percent for test was made.

Dataset 1

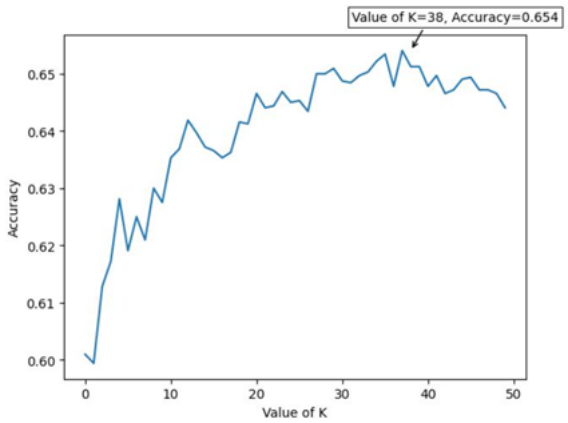


Figure 4.11: Accuracy for different values of K for Dataset 1

The graph in figure 4.11 shows that when the value of K is 38, our model yielded the best result while using Dataset 1.

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 1038 & 556 \\ 552 & 1057 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.654, 0.655, 0.657, 0.656, respectively.

Figure 4.12 shows ROC curve for Dataset 1 when using K-Nearest Neighbors Classifier.

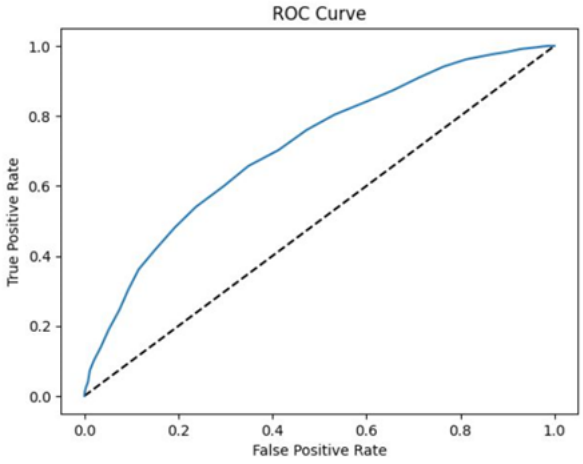


Figure 4.12: ROC curve of K-Nearest Neighbors Classifier for Dataset 1

For the ROC curve in figure 4.12, AUC score of 0.711 was obtained which means that K-Nearest Neighbors has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not

confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The graph in figure 4.13 shows that when the value of K is 13, our model yielded the best result while using Dataset 2.

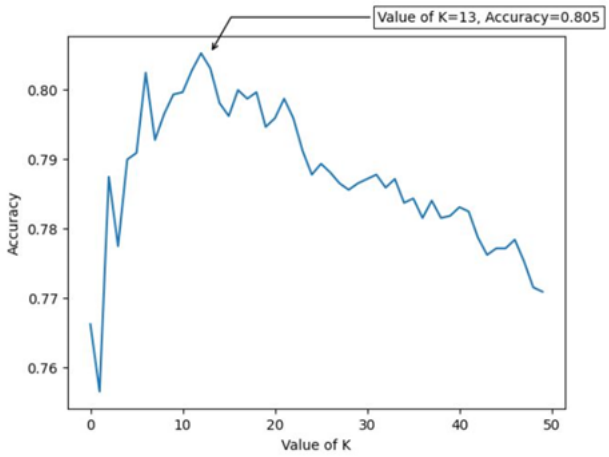


Figure 4.13: Accuracy for different values of K for Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1216 & 378 \\ 246 & 1363 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.805, 0.783, 0.847, 0.814, respectively.

Figure 4.14 shows ROC curve for Dataset 2 when using K-Nearest Neighbors Classifier.

For the ROC curve in figure 4.14, AUC score of 0.881 was obtained which means that K-Nearest Neighbors has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

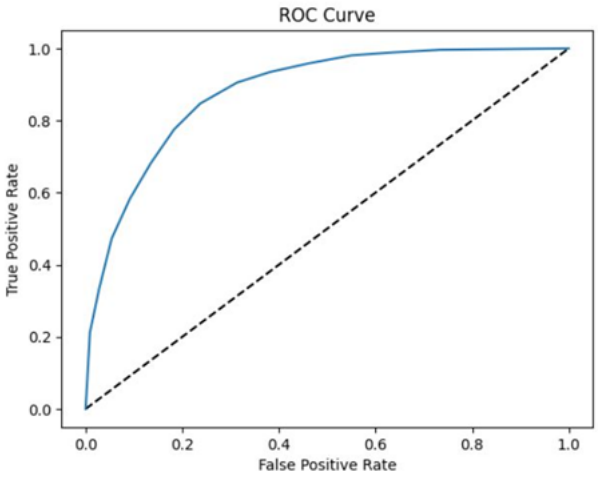


Figure 4.14: ROC curve of K-Nearest Neighbors Classifier for Dataset 2

4.2.6 Gradient Boosting

Gradient Boosting Classifier is an algorithm from one of the many Boosting techniques used in Machine Learning. In a boosting technique, a weak learner is converted into a strong learner by fitting a modified version of the original dataset into each new classification tree. Ensembles are derived from decision tree models, and then a tree is gradually added to the ensemble and it is fit in order to correct the error made by the previous tree. In gradient boosting, the model will identify the shortcomings of the weak learner by using gradients of a differentiable loss function, and gradient descent optimization algorithm. Similar to a neural network, the loss gradient is minimized in every step. The loss function is a measure of how well the coefficients of the model are fitting the data and then performing the classification task.

For our classification, we have set the following parameters: there were 200 boosting stages to be performed, maximum depth of 35 trees, learning rate of 0.01, and 10% of the samples were to be used for fitting the prior learners. A train-test split of 75 percent for train and 25 percent for test was made. Figure 4.15 shows the feature importance of the Gradient Boosting Classifier.

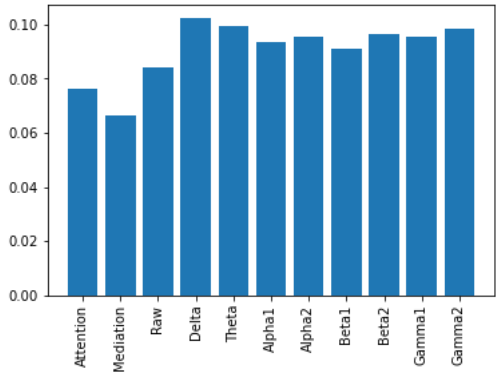


Figure 4.15: Feature importance of the Gradient Boosting Classifier

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 1004 & 590 \\ 501 & 1108 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.659, 0.653, 0.689, 0.670, respectively.

Figure 4.16 shows ROC curve for Dataset 1 when using Gradient Boosting Classifier.

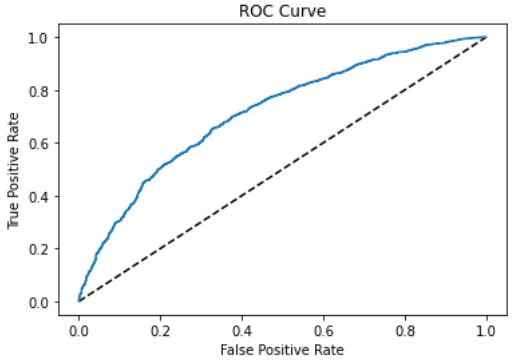


Figure 4.16: ROC curve of Gradient Boosting Classifier for Dataset 1

For the ROC curve in figure 4.16, AUC score of 0.714 was obtained which means that Gradient Boosting has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1140 & 454 \\ 274 & 1335 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.773, 0.746, 0.830, 0.786, respectively.

Figure 4.17 shows ROC curve for Dataset 2 when using Gradient Boosting Classifier.

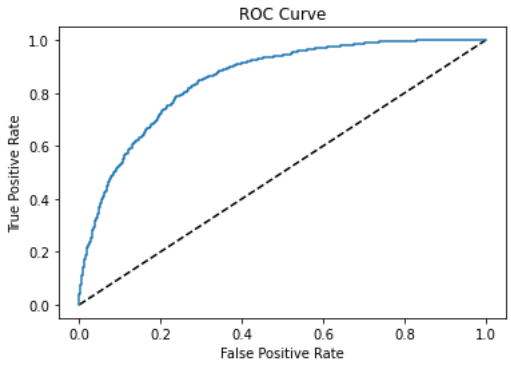


Figure 4.17: ROC curve of Gradient Boosting Classifier for Dataset 2

For the ROC curve in figure 4.17, AUC score of 0.851 was obtained which means that Gradient Boosting has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.2.7 XGBoost

Extreme Gradient Boosting Classifier, more commonly known as XGBoost or XGB algorithm was first discovered by Tianqi Chen and Carlos Guestrin in 2016 and the algorithm has since been widely used in Machine Learning. Similar to other boosting algorithms, XGB also boosts a weak learner into a strong learner using the gradient descent architecture. However, it uses a more regularized model formalization to prevent over-fitting which ultimately results in a better performance. In XGBoost, the tree formation is implemented in a parallel manner which gives it a lead in computation too.

For our classification, we have set the following parameters: maximum depth of 12 trees, learning rate of 0.05, and gamma was set to 0.3 which defined the minimum loss reduction required to make a further partition on a leaf node of the tree. A train-test split of 75 percent for train and 25 percent for test was made. Figure 4.18 shows the feature importance of the XGBoost Classifier.

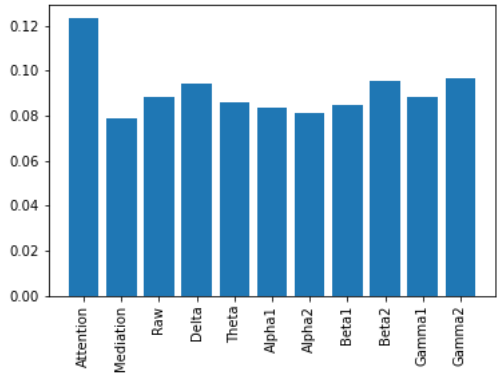


Figure 4.18: Feature importance of the XGBoost Classifier

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 1003 & 591 \\ 517 & 1092 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.654, 0.649, 0.679, 0.663, respectively.

Figure 4.19 shows ROC curve for Dataset 1 when using XGBoost Classifier.

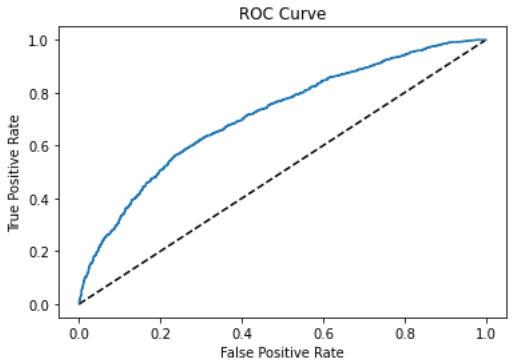


Figure 4.19: ROC curve of XGBoost Classifier for Dataset 1

For the ROC curve in figure 4.19, AUC score of 0.714 was obtained which means that XGBoost has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1203 & 391 \\ 250 & 1359 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.801, 0.777, 0.845, 0.809, respectively.

Figure 4.20 shows ROC curve for Dataset 2 when using XGBoost Classifier.

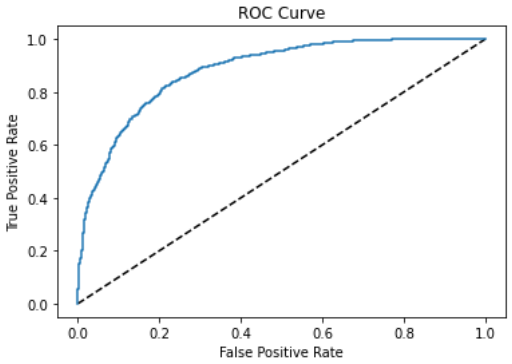


Figure 4.20: ROC curve of XGBoost Classifier for Dataset 2

For the ROC curve in figure 4.20, AUC score of 0.883 was obtained which means that XGBoost has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.2.8 Bidirectional LSTM

To understand how Bidirectional LSTM works, let us first understand the concept of Recurrent Neural Networks (RNN). RNN takes the output of a particular node and forwards it as the input of the very next node. This is done so that the past information is preserved. It is not like traditional neural networks where the input and output are independent to each other. The mechanics of RNN allows the network to preserve its previous information through time. Figure 4.21 shows the RNN structure [37].

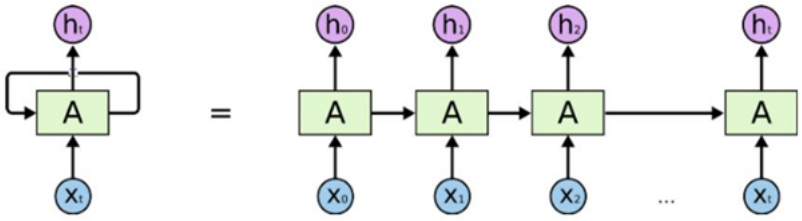


Figure 4.21: RNN structure

However, this configuration of RNN is susceptible to problems known as vanishing gradient or the exploding gradient problem. Vanishing gradient problem arises when the small randomized weights initially assigned to the network get multiplied through the layers of the network. Every neuron should get their weights updated

so that the error is minimized. For the case of RNN, it is not only the layer previous to the output layer which contributes to the error produced at the output, but all the layers of neurons that have been produced back in time. This causes a problem because each time the weights are multiplied, the numbers keep getting smaller and smaller throughout time. This leads to the vanishing gradient problem. If a network suffers from the vanishing gradient problem, the network cannot train properly because it would forget all of the past information [38].

A solution to this problem is the implementation of LSTM. LSTM is a special kind of RNN, which allows the network to retain the past information for a longer period of time. Figure 4.22 shows the structure of a LSTM [39].

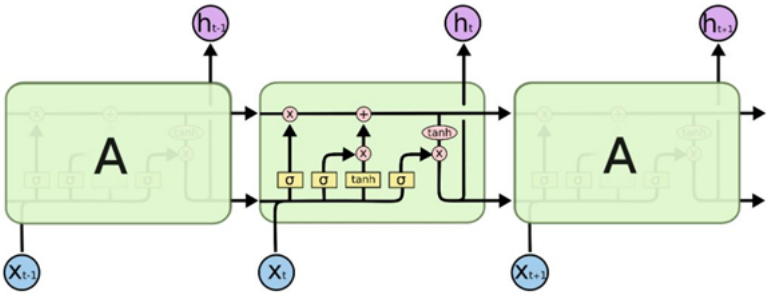


Figure 4.22: LSTM structure

LSTM incorporates output gate, input gate, forget gate and a cell state. The cell state acts as the container which stores the information from the previous states. The forget gate looks at the past information and tells the cell state how much of the previous information it should forget. The input gate determines how much of the new information should enter the cell state. The output gate determines how much of the current information preserved in the cell state should go to the next hidden state [40]. This current configuration allows better performance than traditional RNNs. However, the implementation of Bidirectional LSTM allows the achievement of even better results.

Bidirectional LSTM allows the network to run in two ways, one way is from the past to the future, while the other way is from the future to the past. This does not only allow the past information to be preserved, the network can now also predict what the future information may be like. Figure 4.23 shows the structure of a Bidirectional LSTM [37].

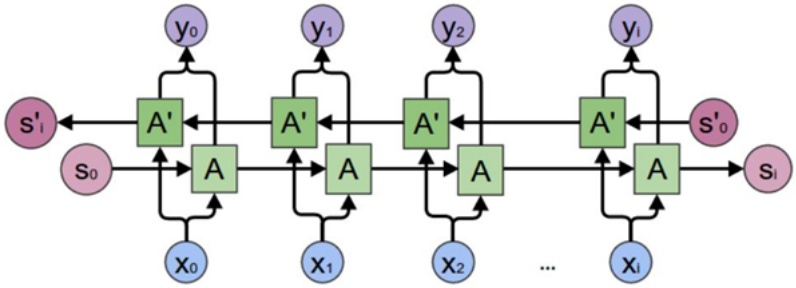


Figure 4.23: Bidirectional LSTM structure

Dataset 1

The following confusion matrix is constructed when using Dataset 1.

$$\begin{bmatrix} 1034 & 560 \\ 535 & 1074 \end{bmatrix}$$

For Dataset 1, the accuracy, precision, recall, F1 scores are 0.658, 0.657, 0.667, 0.662, respectively.

Figure 4.24 shows ROC curve for Dataset 1 when using Bidirectional LSTM.

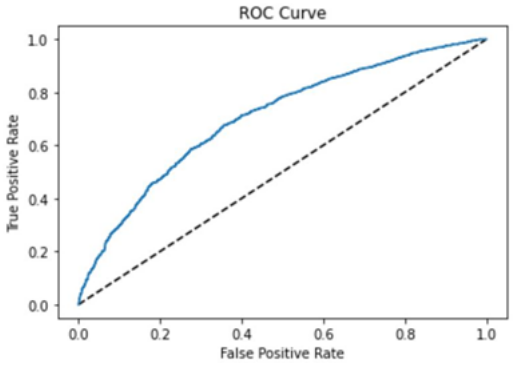


Figure 4.24: ROC curve of Bidirectional LSTM for Dataset 1

For the ROC curve in figure 4.24, AUC score of 0.704 was obtained which means that Bidirectional LSTM has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

Dataset 2

The following confusion matrix is constructed when using Dataset 2.

$$\begin{bmatrix} 1231 & 363 \\ 249 & 1360 \end{bmatrix}$$

For Dataset 2, the accuracy, precision, recall, F1 scores are 0.809, 0.789, 0.845, 0.816, respectively.

Figure 4.25 shows ROC curve for Dataset 2 when using Bidirectional LSTM.

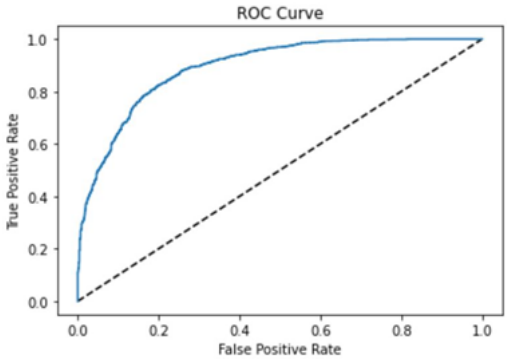


Figure 4.25: ROC curve of Bidirectional LSTM for Dataset 2

For the ROC curve in figure 4.25, AUC score of 0.893 was obtained which means that Bidirectional LSTM has a relatively higher probability that it would successfully differentiate the students who are confused from the students who are not confused, since the AUC value lies between 0.5 and 1. This signifies that the TP and TN values would be higher than the FP and FN values.

4.3 Results and Analysis

After building all the models, they were evaluated to determine how successful the models have been to identify confusion states in students based on their EEG signals. The confusion matrices mentioned before for each model in section 4.2 was our primary source of information for calculating the metrics required for evaluation. TThe parameters of a confusion matrix are False Positive (FP), True Postive (TP), False Negative (FN), True Negative (TN). Figure 4.26 represent the layout of the confusion matrix.

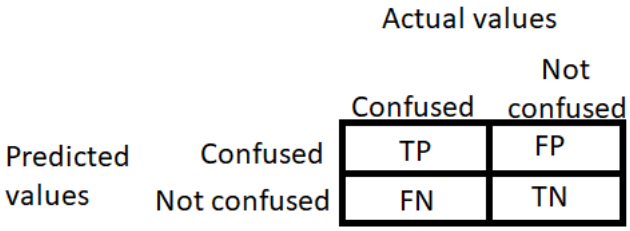


Figure 4.26: Confusion Matrix Layout

The true positive and true negatives are indicators of the predictions the model made correctly and the false positive and false negatives are the wrong predictions

made of the model.

The accuracy is calculated by computing the ratio of the correctly classified predictions to the total number of predictions. Equation 4.1 gives the accuracy of the model from the components of a confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

The precision is the ratio of the number of correctly predicted positive outcomes to the total number of total positive predictions from the test data. Equation 4.2 gives the precision of the model from the components of a confusion matrix.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Another metric used for evaluation was the recall score, which for tasks of binary classifications are more commonly known as the sensitivity. It measures how accurately the model predicted the outcomes of the test data compared to all the samples of that actual outcome in the test data. For our research, the recall score would give a measure of how many confused students were actually classified as being confused. Equation 4.3 is used for calculating the recall score.

$$Recall\ score = \frac{TP}{TP + FN} \tag{4.3}$$

The last metric used was the F1-score, which is a weighted average of the recall and the precision. Along with all the previous metrics, the F1-score is considered to be a better measure of evaluation because it indicates how well the model performed despite having uneven class distribution. Equation 4.4 is used for calculating the F1-score.

$$F1\text{-}score = \frac{2 \times (Sensivity \times Precision)}{(Sensivity + Precision)} \tag{4.4}$$

Confusion Matrix Summarization

Algorithm	**TP**	**FP**	**FN**	**TN**
Decision Tree	950	644	630	979
Random Forest	1032	562	496	1113
Bagging with Random Forest	1048	562	502	1107
Gaussian Naive Bayes	926	668	640	969
K-Nearest Neighbors	1038	556	552	1057
Gradient Boosting	1004	590	501	1108
XGBoost	1003	591	517	1092
Bi-LSTM	1034	560	535	1074

Table 4.1: Confusion matrix models for Dataset 1

Algorithm	**TP**	**FP**	**FN**	**TN**
Decision Tree	1128	466	404	1205
Random Forest	1198	396	244	1365
Bagging with Random Forest	1188	406	235	1374
Gaussian Naive Bayes	996	598	567	1042
K-Nearest Neighbors	1216	378	246	1363
Gradient Boosting	1140	454	274	1335
XGBoost	1203	391	250	1359
Bi-LSTM	1231	363	249	1360

Table 4.2: Confusion matrix models for Dataset 2

In table 4.1, we have summarized all the information available from a confusion matrix, i.e., TP, FP, TN, FN, for all the Machine Learning models applied on Dataset 1. It can be seen that Bagging with Random Forest is overall the best performer while Gaussian Naïve Bayes had the lowest TP/TN values, making it the overall worst performer.

In table 4.2, we have summarized all the information available from a confusion matrix, i.e., TP, FP, TN, FN, for all the Machine Learning models applied on Dataset 2. It can be seen that Bi-Directional LSTM is overall the best performer while Gaussian Naïve Bayes had the lowest TP/TN values, making it the overall worst performer.

Performance Evaluation Summarization

Algorithm	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1 Score(%)**
Decision Tree	60.2	60.3	60.8	60.6
Random Forest	67.0	66.4	69.2	67.8
Bagging with Random Forest	67.3	67.0	68.8	67.8
Gaussian Naive Bayes	59.2	59.2	60.2	59.7
K-Nearest Neighbors	65.4	65.5	65.7	65.6
Gradient Boosting	65.9	65.3	68.9	67.0
XGBoost	65.4	64.9	67.9	66.3
Bi-LSTM	65.8	65.7	66.7	66.2

Table 4.3: Performance evaluation of models for Dataset 1

Algorithm	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1 Score(%)**
Decision Tree	72.8	72.1	74.9	73.5
Random Forest	80.0	77.5	84.8	81.0
Bagging with Random Forest	80.0	77.2	85.4	81.1
Gaussian Naive Bayes	63.6	63.5	64.8	64.1
K-Nearest Neighbors	80.5	78.3	84.7	81.4
Gradient Boosting	77.3	74.6	83.0	78.6
XGBoost	80.1	77.7	84.5	80.9
Bi-LSTM	80.9	78.9	84.5	81.6

Table 4.4: Performance evaluation of models for Dataset 2

In table 4.3, we have summarized all the information we obtained for performance evaluation, i.e., Accuracy, Precision, Recall and F1-Score, for all the Machine Learning models applied on Dataset 1. It can be seen that Bagging with Random Forest is overall the best performer while Gaussian Naïve Bayes had the lowest values for all the metrics, making it the overall worst performer.

In table 4.4, we have summarized all the information we obtained for performance evaluation, i.e., Accuracy, Precision, Recall and F1-Score, for all the Machine Learning models applied on Dataset 1. It can be seen that Bi-Directional LSTM is overall the best performer while Gaussian Naïve Bayes had the lowest values for all the metrics, making it the overall worst performer.

Accuracy Comparison for Scaling

Dataset 1

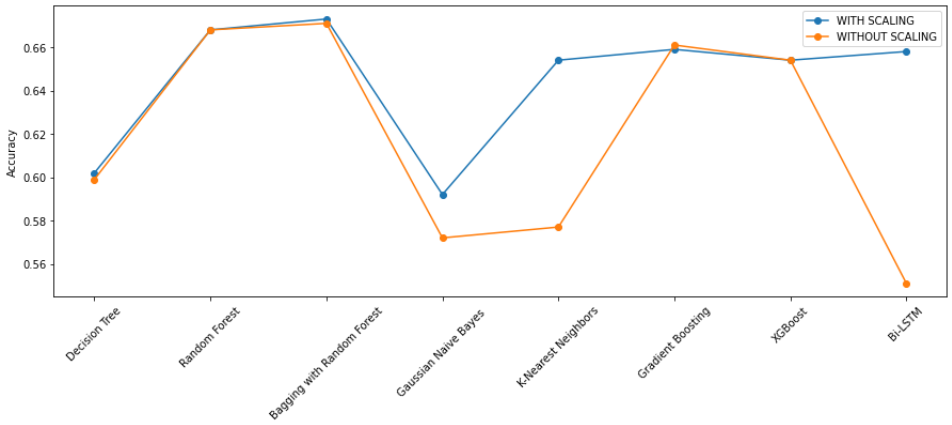


Figure 4.27: Accuracy for Dataset 1 with Scaling Vs. without Scaling

Dataset 2

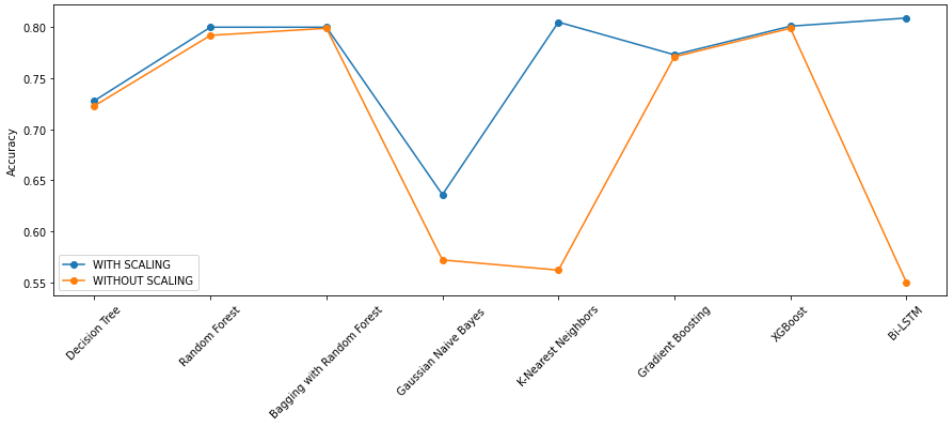


Figure 4.28: Accuracy for Dataset 2 with Scaling Vs. without Scaling

Figures 4.27 and 4.28 show how the Machine Learning model accuracies respond to scaling with PowerTransformer in both Dataset 1 and Dataset 2, respectively. It can be seen that K-Nearest Neighbor algorithm and the Bi-Directional LSTM model show the best response to scaling, with Gaussian Naïve Bayes classifier showing a slight increase only. The increase in K-Nearest Neighbor model accuracy for Dataset 1 was by approximately 8% and 16% for Dataset 2. On the other hand, the increase in model accuracy for Bi-Directional LSTM for Dataset 1 was by approximately 11% and 25% for Dataset 2.

Accuracy Comparison for Datasets

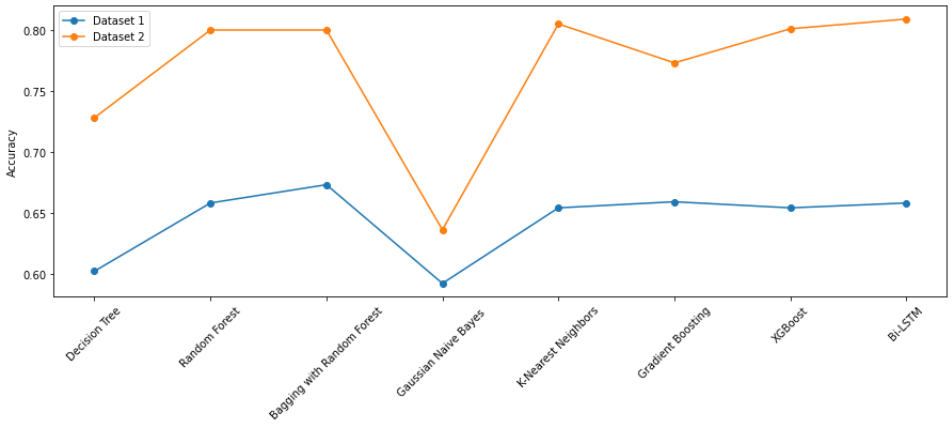


Figure 4.29: Accuracy for Dataset 1 Vs. Dataset 2

Figure 4.29 illustrates the accuracies of the various Machine Learning models used for both Dataset 1 and 2. It can be seen that Dataset 2 has better accuracies overall, a result of what we believe is including the CourseID in our modified dataset which led to better classification when taken the content being viewed into account.

Chapter 5

Conclusion and Future Work

In our research, we have attempted to apply Machine Learning techniques on EEG brainwave data extracted from students as they viewed Massive Open Online Courses (MOOC) of various difficulty levels, and establish whether the student was in a state of confusion or not. Our work was inspired from the ongoing COVID-19 pandemic where learning has been forced to take a more online approach than the world had ever seen before. We discovered that in our research, as well as in some related works, using the EEG brainwave values directly from the confused dataset [1], yields unsatisfactory model accuracies, which could partly be because of the extremely high skewness in the brainwave values. Scaling the data, in this case applying the PowerTransformer scaler yields the best results. It improves the accuracy by a great deal in case of some classifiers such as the K-Nearest Neighbors algorithm, compared to the results of the related works. However, it can be said that the results still are not satisfactory. This is where our findings from our synthesized dataset play a critical role. An alternate approach can be used to train the classifiers to classify the confusion state of students. Previously, we only resorted on using purely EEG brainwave signals to determine the confusion level of the student. However, if we add a tag of which particular video a student is watching along with the EEG signals, our classifiers can perform significantly better. It is because our classifiers can learn which particular video tends to make a student prone to more confusion. So together with the EEG signals and the CourseID, we see a performance increase of about 10 to 15 percent. We believe that since this synthesized dataset only has data from 10 students, the models could be further improved by using data from more students. As a part of our future work, we would like to construct our own dataset with the help of the EMOTIV EPOC+ EEG headset available in our Computer Science and Engineering Department. This would allow us to have more control over the data and ensure that the EEG brainwave signals are extracted in a controlled environment of our preference. We can also gather more volunteers of different age groups to enrich our data so that the dataset size is no longer a limiting factor and furthermore, we can explore the impact of age groups on the confusion state. This had been our initial plan which could not be carried out due to lockdown measures because of the COVID-19 pandemic. Our research demonstrates that with the aid of EEG signals and Machine Learning, it is possible to have models which determine confusion states, and subsequently can be applied to discover the effectiveness of online learning.

Bibliography

[1] H. Wang, *Confused student EEG brain wave dataset*, 2017. [Online]. Available: https://www.kaggle.com/wanghaohan/confused-eeg.

[2] H. Wang, Y. Li, X. Hu, Y. Yang, Z. Meng, and K.-M. Chang, "Using EEG to improve massive open online courses feedback interaction," *CEUR Workshop Proceedings*, vol. 1009, pp. 59–66, Jan. 2013.

[3] A. Ng and M. Jordan, "On Discriminative vs. Generative Classifiers: A comparison of Logistic Regression and Naive Bayes," *Adv. Neural Inf. Process. Sys*, vol. 2, Apr. 2002.

[4] H. Wang, Z. Wu, and E. Xing, "Removing Confounding Factors Associated Weights in Deep Neural Networks Improves the Prediction Accuracy for Healthcare Applications," Jan. 2019, pp. 54–65. DOI: 10.1142/9789813279827_0006.

[5] A. Tahmassebi, A. Gandomi, and A. Meyer-Base, "An Evolutionary Online Framework for MOOC Performance Using EEG Data," Jul. 2018, pp. 1–8. DOI: 10.1109/CEC.2018.8477862.

[6] B. Schildkrout, *Unmasking psychological symptoms: How therapists can learn to recognize the psychological presentation of medical disorders*. 2011, ISBN: 9781118083598. DOI: 10.1002/9781118083598.

[7] B. Kumar, D. Gupta, and S. Goswami Rajat, "Classification of Student's Confusion Level in E-Learning using Machine Learning," 2019, pp. 346–351. DOI: 10.35940/ijitee.B1092.1292S19.

[8] M. Teplan, "Fundamental Of EEG Measurement‖ Measurement Science Review," *Measurement Science Review*, vol. 2, 2002.

[9] C. P. Niemic, "Studies of Emotion: A Theoretical and Empirical Review of Psychophysiological Studies of Emotion," pp. 15–18, Sep. 2004.

[10] Z. Ni, A. Yuksel, X. Ni, M. Mandel, and L. Xie, "Confused or not Confused?: Disentangling Brain Activity from EEG Data Using Bidirectional LSTM Recurrent Neural Networks," vol. 2017, Aug. 2017, pp. 241–246. DOI: 10.1145/3107411.3107513.

[11] M. Yeo, X. Li, K. Shen, and E. Wilder-Smith, "Can SVM be used for automatic EEG detection of drowsiness during car driving?" *Safety Science*, vol. 47, pp. 115–124, Jan. 2009. DOI: 10.1016/j.ssci.2008.01.007.

[12] A. Subasi and M. Gürsoy, "EEG signal classification using PCA, ICA, LDA and support vector machines," *Expert Systems with Applications*, vol. 37, pp. 8659–8666, Dec. 2010. DOI: 10.1016/j.eswa.2010.06.065.

[13] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015.

[14] S. D. Puthankattil, P. Joseph, U. R. Acharya, and C. Lim, "EEG signal analysis: A survey," *Journal of medical systems*, vol. 34, pp. 195–212, Apr. 2010. DOI: 10.1007/s10916-008-9231-z.

[15] S. Siuly, Y. Li, and Y. Zhang, *EEG Signal Analysis and Classification: Techniques and Applications*. Jan. 2016, ISBN: 978-3-319-47652-0. DOI: 10.1007/978-3-319-47653-7.

[16] *Lobes of the brain*. [Online]. Available: https://qbi.uq.edu.au/brain/brain-anatomy/lobes-brain.

[17] *5 Types Of Brain Waves Frequencies: Gamma, Beta, Alpha, Theta, Delta*. [Online]. Available: https://mentalhealthdaily.com/2014/04/15/5-types-of-brain-waves-frequencies-gamma-beta-alpha-theta-delta/.

[18] J. S. Kumar and P. Bhuvaneswari, "Analysis of Electroencephalography (EEG) Signals and Its Categorization–A Study," *Procedia Engineering*, vol. 38, pp. 2525–2536, 2012.

[19] *NeuroSky's eSense™ Meters and Detection of Mental State*. [Online]. Available: http://www.brainathlete.jp/pdf/WP-lee-neurosky-esense.pdf.

[20] Z. Jaadi, *A Step-By-Step Explanation of Principal Component Analysis*, 2019. [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[21] R. Bhadauria, *Linear Discriminant Analysis*, 2019. [Online]. Available: https://www.geeksforgeeks.org/ml-linear-discriminant-analysis/.

[22] A. Ye, *Linear Discriminant Analysis*, 2020. [Online]. Available: https://medium.com/analytics-vidhya/linear-discriminant-analysis-explained-in-under-4-minutes-e558e962c877.

[23] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, pp. 210–229, 1959.

[24] T. Mitchell, *Machine Learning*. 1997, ISBN: 0070428077.

[25] A. Burkov, *The Hundred-Page Machine Learning Book (Hard Cover ed.)* 2019, ISBN: 978-1999579500.

[26] Y.-Y. Song and Y. Lu, "Decision tree methods: Applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, pp. 130–5, Apr. 2015. DOI: 10.11919/j.issn.1002-0829.215044.

[27] P. Gupta, *Decision Trees in Machine Learning*, 2017. [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052.

[28] D. Edla, K. Mangalorekar, G. Dhavalikar, and S. Dodia, "Classification of EEG data for human mental state analysis using Random Forest Classifier," *Procedia Computer Science*, vol. 132, pp. 1523–1532, Jan. 2018. DOI: 10.1016/j.procs.2018.05.116.

[29] L. Fraiwan, K. Lweesy, N. Khasawneh, H. Wenz, and H. Dickhaus, "Automated sleep stage identification system based on time-frequency analysis of a single EEG channel and random forest classifier," *Computer methods and programs in biomedicine*, vol. 108, pp. 10–9, Dec. 2011. DOI: 10.1016/j.cmpb.2011.11.005.

[30] T. Wood, *Random Forests*. [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/random-forest.

[31] D. Dey, *Bagging Classifier*, 2019. [Online]. Available: https://www.geeksforgeeks.org/ml-bagging-classifier/.

[32] D. Optiz, *Bagging Classifier*, 1999. [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume11/opitz99a-html/node3.html.

[33] H. Kamel, D. Abdulah, and J. Al-Tuwaijari, "Cancer Classification Using Gaussian Naive Bayes Algorithm," Jun. 2019, pp. 165–170. DOI: 10.1109/IEC47844.2019.8950650.

[34] S. Parida, S. Dehuri, and S.-B. Cho, "Application of genetic algorithms and Gaussian Naïve Bayesian approach in pipeline for cognitive state classification," Feb. 2014, pp. 1237–1242, ISBN: 978-1-4799-2572-8. DOI: 10.1109/IAdCC.2014.6779504.

[35] D. Subramanian, *A Simple Introduction to K-Nearest Neighbors Algorithm*, 2019. [Online]. Available: https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e.

[36] A. Uberoi, *K-Nearest Neighbors*, 2019. [Online]. Available: https://www.geeksforgeeks.org/k-nearest-neighbours/.

[37] R. Aggarwal, *Bi-LSTM*, 2019. [Online]. Available: https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0.

[38] K. Eremenko, *Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem*, 2018. [Online]. Available: https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem.

[39] C. Olah, *Understanding LSTM Networks*, 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[40] E. Kang, *Long Short-Term Memory (LSTM): Concept*, 2017. [Online]. Available: https://medium.com/@kangeugine/long-short-term-memory-lstm-concept-cb3283934359.