Course: Foundation of Cyber-Physical Systems CS460
(AY 2022-23 Term 1)

Project Title: Music and Dementia Innovation

Project Members:	**Matriculation Number:**
Song Yu Xiang	01372166
Wong Jie Peng	01370290
Luqman Juzaili Bin Muhammad Najib	01384917
Tan Rui Trina	01411344

Section 1.0 Project Goals

Section 1.1: Project Context

Photo 1 (Alzheimer's Patient reaction to music)

Our project partner, Apex Harmony Lodge (AHL), is a community committed to empowering lives affected by dementia. In week 2, AHL has showcased to us how Alzheimer's patients react to music that is considered stimulating. Before music is introduced to the patient, they are only able to answer simple yes or no questions. However, after subjecting them to music that they are familiar with, their cognitive ability has improved drastically where some of them are able to recall their childhood times. Thus, it can be seen that music therapy activates cognitive and speech centres in the brain of these dementia patients.

Section 1.2: Project Problem

While music therapy has a positive impact on the dementia patient as they regain some cognitive ability, it is difficult to know exactly which of these songs are considered stimulating to the patient.

- A person's music taste could be different from another
- Time consuming to track which songs are stimulating by having volunteers to sit with the patients and observe their physical reactions
- Physical reactions are also not always the best indicator for stimulation, which make it difficult to distinguish songs that are actually stimulating.

Section 1.3: Project Goal

In order to make this process more efficient and effective, we aim to produce a classification model which would be able to predict whether a song is considered stimulating or non-stimulating given a set of brain waves. Data that is collected from the EmotivePro headset would be used as input to various machine learning models. From the models used, we want to provide insights into any possible correlation between brain waves and music.

Section 2.0 Initial Research

Section 2.1: Research into Brain Waves present in EmotivPro 3.0

<table>
<tr><td>Beta</td><td>Beta waves can be split into 3 main sections:

<table><tr><td>Low Beta Waves (12.5 Hz to 16 Hz)</td><td>Associated with quiet, focused, introverted concentration.</td></tr><tr><td>Beta Waves</td><td>Associated with increase in energy, anxiety and performance.</td></tr><tr><td>High Beta Waves (16.5 Hz - 20 Hz)</td><td>Associated with significant stress, anxiety, paranoia, high energy, and high arousal.</td></tr></table>
With regards to the headset we were using (EmotivPro 3.0) during our data collection, only two main Beta Waves (Beta L and Beta H) were recorded. For Beta L, it covers the Low Beta Waves from 12-16 Hz while Beta H covers both Beta Waves and high Beta waves, ranging from 16 to 25 Hz.</td></tr>
<tr><td>Alpha</td><td>Alpha Waves are usually measured between 8 Hz - 12 Hz in the EmotivPro 3.0, which is mainly associated with daydreaming, meditating, and a more relaxed state of mind.</td></tr>
<tr><td>Theta</td><td>Theta waves are usually measured between 4 Hz - 8 Hz in the EmotivPro 3.0. The Theta waves are a common marker of drowsiness and they are often present as a transition between an awake state and a sleep state.</td></tr>
<tr><td>Gamma</td><td>Gamma waves are usually measured between 25 Hz - 45 Hz in the EmotivPro 3.0. The Gamma waves are the fastest brain waves produced in the brain, a person is likely to produce Gamma waves when they are intensely focused or actively engaged in solving a problem.</td></tr>
</table>

Section 2.2: Research into potential models

We ultimately wanted to produce a classification model, which is able to take a set of brain waves as input and return us a value of either Not-Stimulating or Stimulating.

To develop our classification model, we researched and found several algorithms that are applicable in the context of this project:

- Logistic Regression
- Support Vector Machines
- K-Nearest Neighbors
- Decision Tree
- Gaussian Naives Bayes

Since ensemble learning models tend to have a higher predictive accuracy, compared to individual models, our team also explored the different ensemble learning models that are applicable in the context of this project:

- Random Forest
- Adaboost (Base Estimator: Decision Tree Classifier)
- Gradient Boosting
- Extreme Gradient Boosting
- Bagging Classifier
- Extra Tree Classifier

We decided to have a mixture of single machine learning models as well as having ensemble learning models in order to investigate how the accuracies of the model perform. This is especially since ensemble learners, which aggregate the input of multiple models through methods such as bagging and boosting, are typically said to perform better than single machine learning models in terms of the accuracies produced. Overall, we will be using both the individual models and ensemble learning models to aid us in finding the better accuracies in our exploratory combination in Section 5.

Section 3.0 Data Collection Process

Section 3.1 Preliminary Testing

Section 3.1.1 Data collection

We decided that it was best to start off the first preliminary testing with the 6 songs that Justin has requested for us to try out. 4 songs would be the standard songs as provided in the spotify playlist, with 2 random songs based on the user's preference. We appointed a member - Yu Xiang, to try out the Emotiv Headset while listening to the 6 songs mentioned.

Song Number	Title
1	Happy birthday
2	Home
3	Unknown Song 1
4	Unknown Song 2
5	POP - Nayeon (Any Song of User's Preference #1)
6	想当年, 我们一起走音的歌 (Any Song of User's Preference #2)

Table 3.1: Songs used for preliminary testing on Yu Xiang

Our initial data collection process required two people - one to wear the headset, another one to collect the data. The tracking of the brain waves was done through the retrieval of the timestamps of when the song is being played. When the Emotiv Headset application was tracking the brainwaves, we set up a timer to synchronise with the data's timestamp, aiming to retrieve a brainwave of a song

based on the length of the song. The stopwatch kept track of the start and the end time of a song. At the end of each song, we provided a 20 sec buffer time to get the listener to indicate - through thumbs up or down, whether a song was stimulating or not.

Prior to this, we also compiled a list of additional songs that we would like to carry out more testing on. The list included over 50 English songs with songs from each genre - Pop, Rock, Jazz, R&B, Hip Hop, Alternative, Indie, EDM. This was done on a google sheet, where each of us filled in songs from the various genres mentioned above, providing the Spotify links to the song and also general information such as the song title, language and genre. Initially, we planned to test out the 6 songs, together with as many songs that we can from the list that we compiled on the same day. However, due to some of the challenges faced, we did not use any of these songs during our preliminary test.

Section 3.1.2 Challenges faced

During our initial testing, we faced several challenges:

- **Battery problems:** When we first set up the Emotiv Headset, we realised that the battery was at 3%. We tried to carry out the testing while the Emotiv Headset was being charged, but realised that we were unable to do so because it was unable to connect to the computer while charging. Hence, we had to account for some time out of our designated 4-hour slot to let the headset charge.

- **Unfamiliarity with setting up headset:** We took awhile to set up the hardware as we had to constantly adjust the sensors to achieve a desirable Contact Quality and EEG Quality Value of > 90 before starting. We tried to obtain a good connection value through adding an extra saline solution towards the buds of each sensor to ensure that it is moist enough. Additionally, we also tried to adjust the orientation of the headset being put on the head.

Image 3.2: Unfamiliarity with setting up headset (application of additional saline solution)

- **Uncertain inputs for model training**: Our team was uncertain about what exactly the inputs were for the classification model. The possible inputs that we had in mind were:
    a) Brainwaves of a person when listening to the songs
    b) The traits of the song e.g. genre, song length, artist, etc. or a
    c) Combination of both the traits of the song and the brain waves that corresponds to it.

- **Confusion with Emotive Metrics:** We thought that the FFT values provided by the headset was with respect to the songs as they were measured in decibels (dB) to which we assumed to be relating to the song. We then consulted our professor and instructor to which we found out that the FFT values were actually for the brain waves as the headset only measures brain activity.

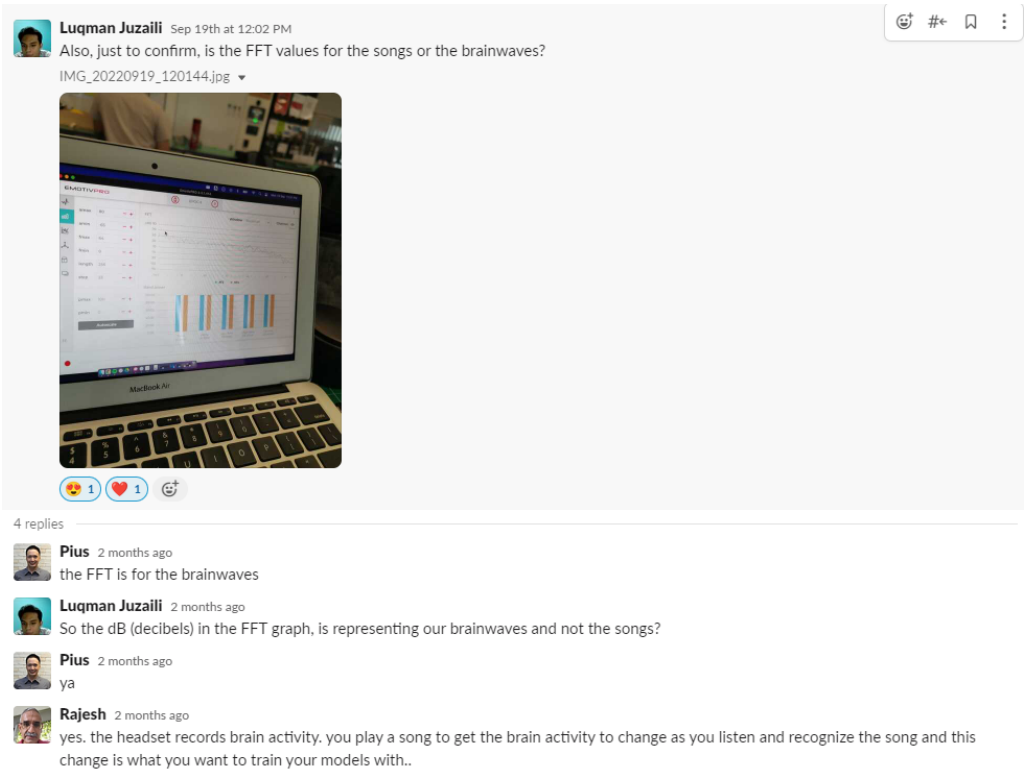


Image 3.1: Clarification with Rajesh and Pius regarding about FFT measurements

The challenges presented above have somewhat caused a delay in our overall data collection process, but it was useful in helping us move forward because we had a clearer idea of what should be input into the model for training.

Section 3.2: Actual Testing

After our preliminary testing, we thought more about the process and realised that our initial method did not account for differences in stimulation within songs themselves and only allowed for an overall label. Given this, we decided to carry out a new way of collecting data, taking into account all the challenges we faced during the preliminary testing. Instead of labelling whether a song is stimulating or not as a whole, we labelled specific sections of the song as stimulating, non-stimulating, or neutral. To facilitate our data collection process, keybinds were assigned to those labels to create markers to track the parts of the song it corresponds to.

Key bindings	Value associated
Num 8	888 - Neutral
Num 9	999 - Stimulating
Num 0	666 - Non-stimulating
Letter S	200 - Start Song
Letter E	404 - End Song

The values in the table proved to be rather useful to us when the data is exported into a CSV format. It simplified things rather easily as the filtering of songs can be done rather easily through the different values identified

Section 3.2.1 Finalised data collection process

The overall data collecting process will be done by minimally 2 people where one will be listening to the songs (listener) and another one will be collecting the data (recorder) can be listed as follows:

1. At the start of each song, the recorder will set a marker value of “200” to indicate the start of the song by pressing “S”.
2. Throughout the song, the listener will indicate a label for the current part of the song.
    a. If the part of the song is stimulating, the listener will indicate a thumbs up
    b. If the part of the song is neutral, the listener will indicate a thumbs up, 90 degrees to the left.
    c. If the part of the song is non-stimulating, the listener will indicate a thumbs down
3. The recorder will then press the respective markers to correspond with the signal given by the listener.
    a. If the part of the song is stimulating, recorder would press “9” which would set a marker value “999”
    b. If the part of the song is neutral, recorder would press “8” which would set a marker value “888”
    c. If the part of the song is non-stimulating, recorder would press “6” which would set a marker value “666”
4. Once the song has ended, the recorder would press “E” which would set a marker value of “404” to indicate the end of the song.

Image 3.3: Data collection process

Though our newly improved process was more effective, we had to discard the data collected from the preliminary testing as the methods used to label the data were different and recollect it later just to ensure that our data format remained consistent.

For a start, we collected data only on a single member of the group, Yu Xiang, collecting a total of 36 songs. However, we felt that focusing on only one member may cause the model to overfit and so we decided to expand further to the other members of the group to investigate whether our model would be more generalisable. Data was collected on Luqman and Jie Peng just to ensure we have even more data should we need to carry out further testing on the models. The EEG was unable to stabilise when collected on Trina, so we were unable to do data collection. Out of the two members, we found out that only the data collected from Luqman could be used as it is the only one which contains the desirable contact and EEG quality.

Section 4.0: Data Preprocessing

Section 4.1: Preparation of excel data

Newly Created Data Columns	
EQ.OVERALL.New	**Reason for creation:** EQ.OVERALL measures EEG Quality which is necessary for us to determine which rows to drop, but it is only populated once every 63 rows, leaving many blank cells in between one value and the next. **Assumption:** EQ.OVERALL is the same for all the rows between one recorded value and the next **Method of creation:** We populated the new column, using an Excel Formula with a IF

	function: ● IF Current Cell in EQ.Overall is blank ● THEN follow the previous cell in EQ.Overall.NEW ● ELSE follow the value in Current cell in EQ.Overall With this new column, we are able to drop using NaN more effectively, without removing any important rows.
Target	**Reason for creation:** Purpose of "Target" is to know if our Subject is currently feeling the song is "Stimulating", "Neutral", "Non-Stimulating". **Method of creation:** Made use of the Column: MarkerValueInt. EmotivPro 3.0 provided a keystroke function, which allows us to mark exact points within the dataset itself by pressing keys (as mentioned in section 3.2.1) These are our input MarkerValueInt that we have keyed into EmotivPro 3.0: ● 200 refers to the start of a Song ● 404 refers to the end of a Song ● 888 refers to a Neutral (0) ● 666 refers to Non-Stimulating ● 999 refers to Stimulating Thus, we have populated our Target Column by referencing this MarkerValueInt with a Nested IF Function in Excel: ● If MarkerValueInt is 666 ● Then Target Column will be 1 ● If MarkerValueInt is 888 ● Then Target Column will be 0 ● If MarkerValueInt is 999 ● Then Target Column will be 1 ● If MarkerValueInt is blank ● Then Target Column will follow the previous Target Column Row ● Else Input a Blank Column (This is to account for 200 and 404 MarkerValueInt)
Song	**Reason for creation:** Track which rows of data belong to which song, similar to a Song ID. **Method of creation:** Made use of the Column: MarkerValueInt. When there is a 200 value (refers to the Start of a Song), we will increment the Song accordingly using a Excel Formula: ● If MarkerIntValue is 200 ● Then Song ID will increment by 1 ● Else Song ID remains the same

Section 4.2: Initial Data Collection

In our data collection process, initially we decided to split it into 3 possible target values:

- "-1" for Non-Stimulating
- "0" for Neutral
- "1" for Stimulating

However, we ran a Preliminary analysis using our different combinations in Section 5 using a multiclass as our target and our models performed poorly with F1 scores ranging from 10% to 35%. So, we researched the matter and found out that not all Machine Learning algorithms support multiclass classification. For instance, Logistic Regression and Support Vector Machines do not. It may also be more difficult for models to predict multiple classes instead of a binary class division.

```
def dummy(charvalue):
    if charvalue == -1:
        return 0
    elif charvalue == 0:
        return 0
    elif charvalue == 1:
        return 1

df['Target'] = df['Target'].apply(dummy)
df4['Target'] = df4["Target"].apply(dummy)
df5['Target'] = df5['Target'].apply(dummy)
df6["Target"] = df6["Target"].apply(dummy)
```

Figure 4.1.2: Code to change rows where Target = -1 to Target = 0

As such, we have scoped down our initial data collection into just 2 main Target values:

- "0" for Non-Stimulating (Covers both -1 and 0 from the dataset)
- "1" for stimulating

Section 4.2.1: Selection of Relevant Dataset

Since there are a total of 181 Columns in the dataset, we have to filter out and take out the columns that are relevant to our project goals. We have identified 3 main categories which are: Performance Metrics (PM), Beta Waves (both H and L) and all the Brain waves.

Information about the Data Object	
Time Stamp & Original Time Stamp	This time stamp is currently in Epoch time. However, this row will be dropped afterwards in the combination portion in Section 5.0.
EQ.OVERALL.New	This is a newly created column as EmotivPro 3.0 only fills in the data
Target	It will tell us if the person is feeling "Neutral", "Stimulated", "Non-Stimulating" for the current data object.
Song	It will tell us the Song ID, the current data object belongs to.

Data for the different combination	
PM.Scaled	We have selected PM Scaled and not either PM.Raw or PM.MinMax. This is because Scaling ensures that high data points do not dominate the smaller data points.
BetaL & BetaH	We have selected Beta waves in our combination as Low Beta Waves and High Beta waves are able to tell us if the person is highly active after listening to the song.
All POW.<sensor>.<band>	We have selected all brain waves POW sensors for exploratory purposes as the data as a whole might provide us with better prediction that PM, and just Beta Waves are unable to pick up.
Data Columns that are not relevant [Dropped]	
Mental Commands	This column is not part of our project goals.
Facial Expression	This column is not part of our project goals. In addition, Facial Expression might potentially become noise in the dataset as our test subject could potentially have a facial expression that does not correctly indicate the Target value.
Motion	This column is not part of our project goals. In addition, Facial Expression might potentially become noise in the dataset as our test subject could potentially move during the collection of dataset that does not correctly indicate the Target value.

Section 4.2.2: Cleaning of Dataset (EQ Quality)

We have two main step in our Cleaning of Dataset:

1. In the first step, since there are multiple NaN intervals in the dataset, we will drop all the columns that are NaN.

```
training_yuxiang_30_songs_PM.dropna(how="any", axis=0, inplace=True)
```

2. In the second step, we will drop data objects that EQ Quality is below 90%.

```
(training_yuxiang_30_songs_PM[training_yuxiang_30_songs_PM["EQ.OVERALL.New"] < 90].index)
```

Hence, with these two steps, we ensure that our dataset does not contain any missing values and they are data that are accurate which can be used in our Combination Models.

Section 4.3: Preparation of Data Sets

In the preparation of Dataset, we have prepared the dataset into multiple brain waves:

	Dataset Name
YuXiang's 30 Song Brain Waves	1. pm_df [Performance Metrics Scaled] 2. Beta_df [POW Beta H and Beta L] 3. other_brain_df [All Brain Waves]
YuXiang's 6 Song Brain Waves	1. pm_df_yuxiang_6 [Performance Metrics Scaled] 2. beta_yuxiang_6_df [POW Beta H and Beta L] 3. other_brain_yuxiang_6_df [All Brain Waves]
Luqman'6 Song Brain Waves	1. pm_df_luqman_6 [Performance Metrics Scaled] 2. beta_luqman_6_df [POW Beta H and Beta L] 3. other_brain_luqman_6_df [All Brain Waves]
JiePeng's 6 Song Brain Waves	1. pm_df_jiepeng_6 [Performance Metrics Scaled] 2. beta_jiepeng_6_df [POW Beta H and Beta L] 3. other_brain_jiepeng_6_df [All Brain Waves]

Section 5.0 Dataset Analysis

Section 5.1: Exploratory Data Analysis (EDA)

We decided to investigate the relationship between certain attributes and the Target variable to understand which attributes may be more crucial in predicting stimulation and non-stimulation. Yu Xiang's dataset of 36 songs was used in this analysis.

Section 5.1.1: Performance metrics and Target Values

First, we explored the relationship between Performance metrics (PM) (Engagement, Excitement, Stress, Relaxation, Interest, Focus) and Target Values (0, 1). Exploration will be done through:

1. Barplot (mean of PM values)

2. Boxplot (to understand the distribution of PM values)

PM values used are scaled instead of raw, min or max as provided from the Emotiv csv.

Barplot of PM values

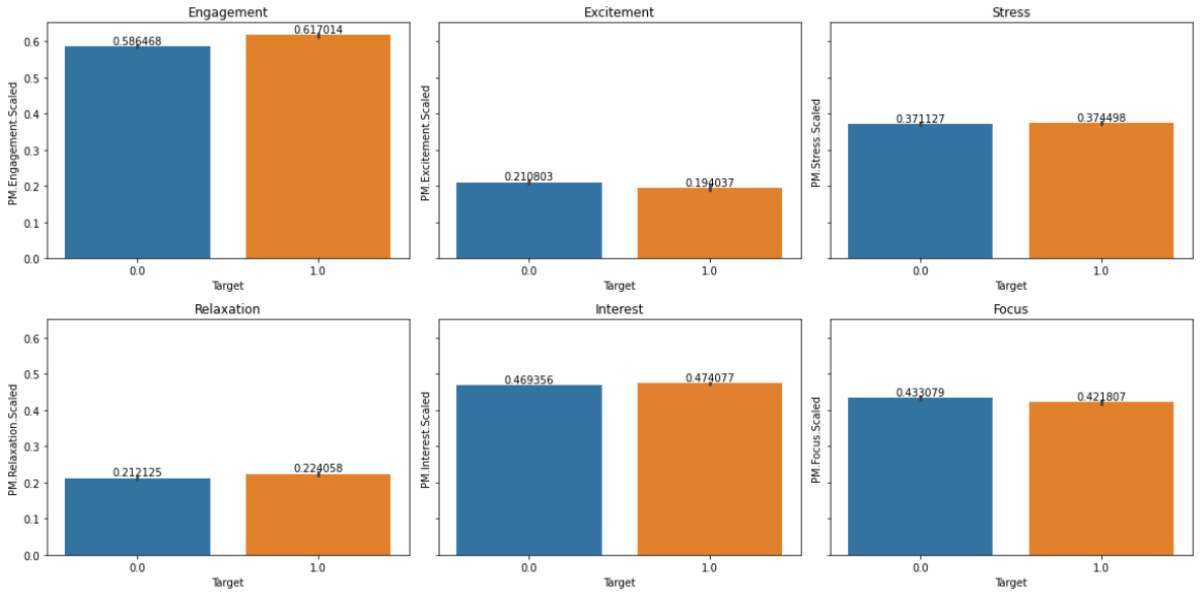


Figure 5.1.1: Barplot of average PM values split based on Target

Key observations:

- Average Engagement higher when Target = 1 compared to Target = 0 (5.2% higher)
- Average Excitement lower when Target = 1 compared to Target = 0 (8% lower)
- Average Stress, Relaxation, Interest and Focus values were very similar between Target = 1 and Target = 0 .

It makes sense that more stimulation leads to higher Engagement, though it is interesting that this is the opposite for Excitement. Other PMs show little to no difference between stimulation and non-stimulation, possibly indicating that they will not be important for our predictions. However, it may be difficult to conclude anything concrete as the values only differ slightly. Barplots also provide limited information as the mean may not truly be representative of the data.

Boxplot of PM values

With the limited information provided from the barplot, we attempted to investigate the validity of the possible patterns found by checking the distribution of the PM data using a boxplot.

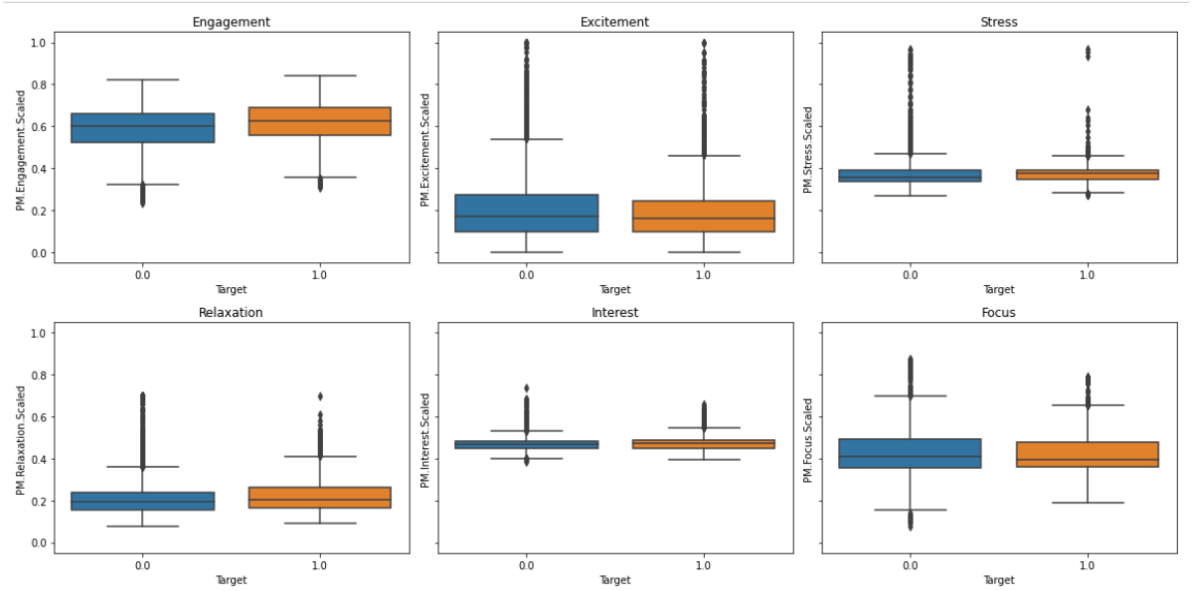


Figure 5.1.2: Boxplot of PM values split based on Target

Key observations:

- High variation in values for Excitement, Stress: Significant number of values which are more than 1.5 times the Interquartile Range (IQR) away from the 3rd Quartile
- Little variation for Engagement, Interest, Focus

Key insights from this section:

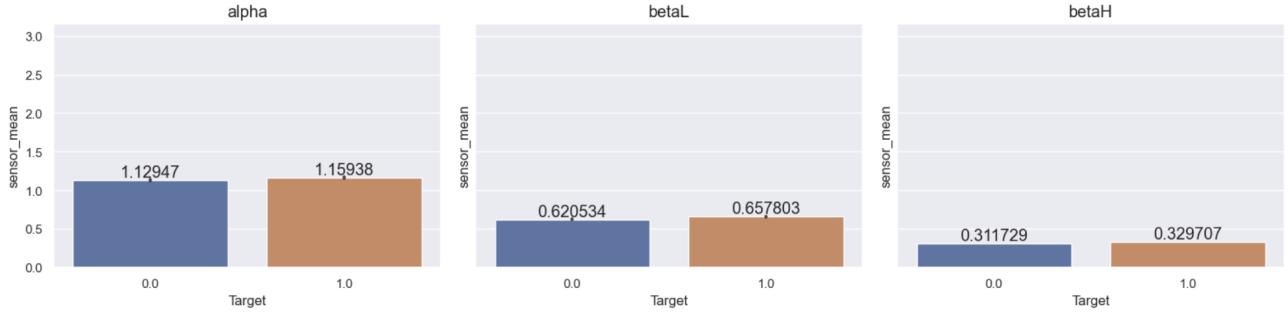
- It is possible that greater stimulation translates to higher Engagement values and lower Excitement values, but it cannot be fully confirmed as the difference between stimulated and non-stimulated values is not large and boxplots show a larger range of values.
- Due to this, it is uncertain if PM waves will be useful in predicting Target values and determining whether it is stimulating or not.

Section 5.1.2: Brainwaves and Target values

We explored the relationship between all brainwaves (Alpha, BetaL, BetaH, Gamma, Theta) and Target Values (0, 1). Exploration will be done through:

1. Barplot (mean values for each type of wave, split by Target)
2. Boxplot (to understand the distribution of the wave values)

Barplot



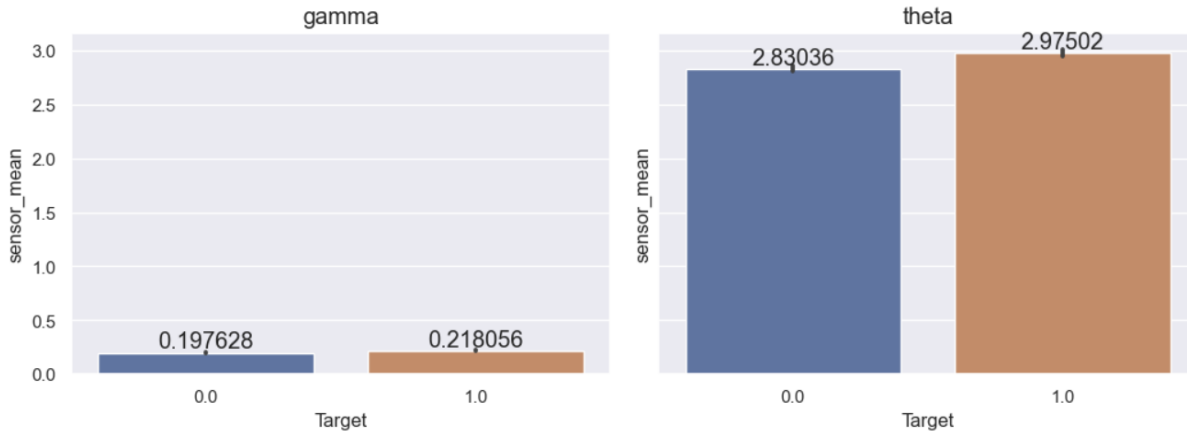


Figure 5.1.2.1: Barplot of average wave values (mean across all sensors) split based on Target

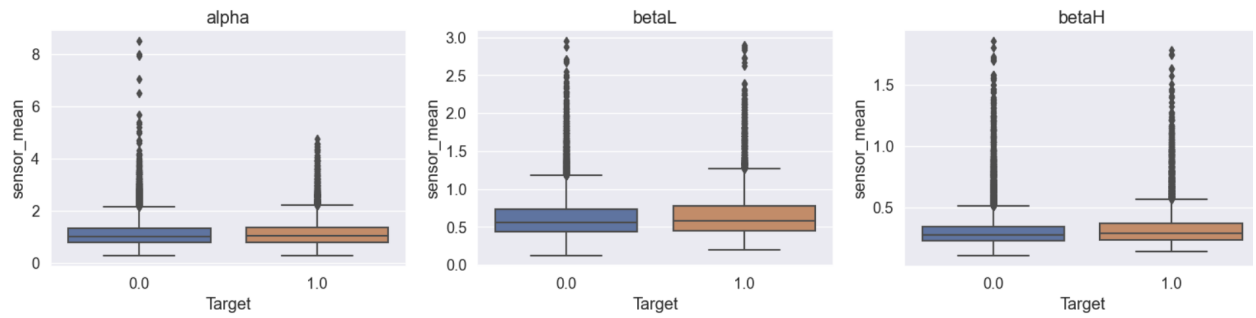
Observations:

- All average wave values (Alpha, BetaL, BetaH, Gamma, Theta) are higher when Target = 1 compared to Target = 0. They are higher by:
    - Alpha: 2.6%
    - BetaL: 6%
    - BetaH: 5.8%
    - Gamma: 12.3%
    - Theta: 5.1%
- Theta shows the largest increase in average value for Target = 1, followed by BetaL, Alpha, Gamma, BetaH

This suggests that more stimulation leads to higher Alpha, BetaL, BetaH, Gamma and Theta waves. Relating these to what each of the waves are generally related with as stated in section 2.1, it could mean an increase in daydreaming/relaxation (Alpha), drowsiness (Theta), intense focus (Gamma), energy (BetaH), and concentration (BetaL). Since an increase in everything at once does not seem to make sense as the waves imply contradictory states, it is likely that within each Target value, the different wave values peaked at different timings. However, since the mean may not necessarily be a good representation of the data, we continued our investigation by plotting the boxplot.

Boxplot

In order to plot the boxplot, the mean sensor values in each row for each type of wave were found. As the values for Theta are much larger than the other waves, in order to visualise the distribution for each wave clearly, the y-axis range was not standardised.



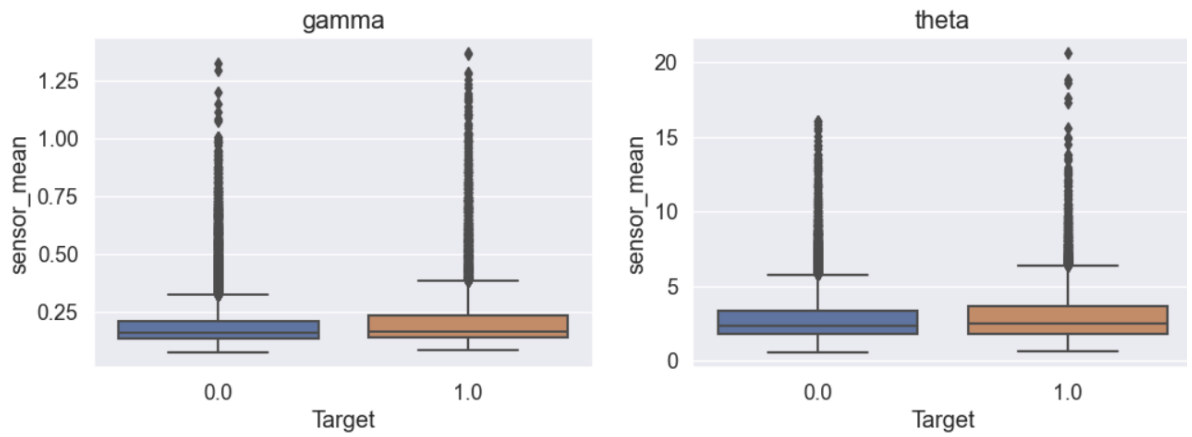


Figure 5.1.2.2: Boxplot of average wave values (average per row) split based on Target

Observations

- Compared to PM, the data for the waves seems to have a much higher variation, with many values more than 1.5 times the IQR away from the 3rd Quartile.

Large variations and only slight differences in the average values for the waves make it difficult to conclude that the observations from figure 5.1.2.1 hold true. After finding this quite inconclusive, we realised that a possible reason could be that the range of values may differ from sensor to sensor and any changes in values from Target = 0 to Target = 1 may not be obvious once aggregated. A preliminary boxplot of the different sensors for Alpha waves proved that this may be the case.

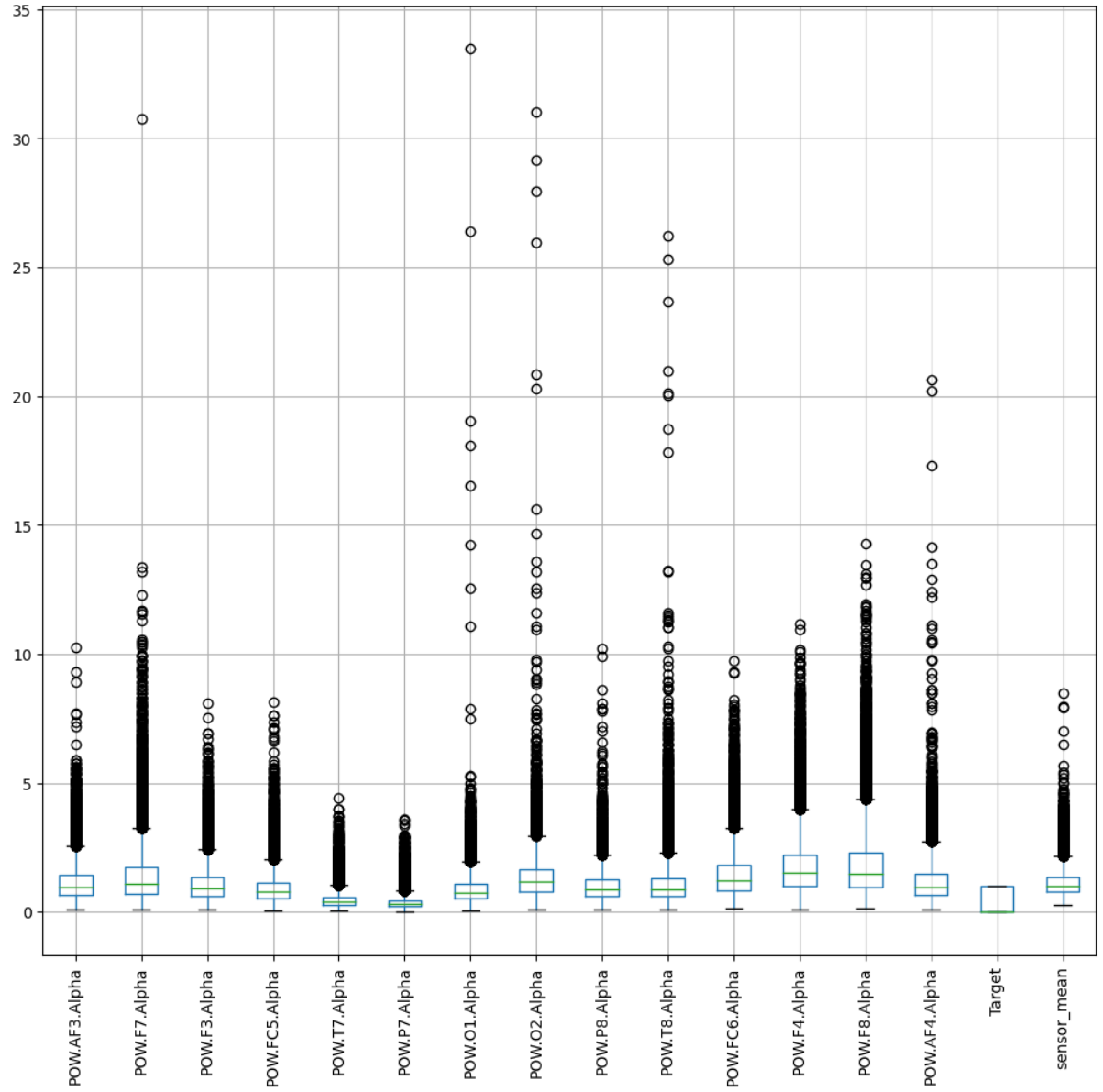


Figure 5.1.2.3: Boxplot for Alpha waves for different sensors

From here, we can see that the range of alpha wave values across different sensors do vary, suggesting that it may be useful to conduct the same analysis, but separated by each sensor.

Section 5.1.3: Brainwaves (by sensor) and Target values

We explored the relationship between all brainwaves (Alpha, BetaL, BetaH, Gamma, Theta) and Target Values (0, 1) *by sensor*. Exploration will be done through:

1. Barplot (mean values for each type of wave, split by Target)
2. Correlation heatmap (to determine how to use the wave values for classification later)

Alpha

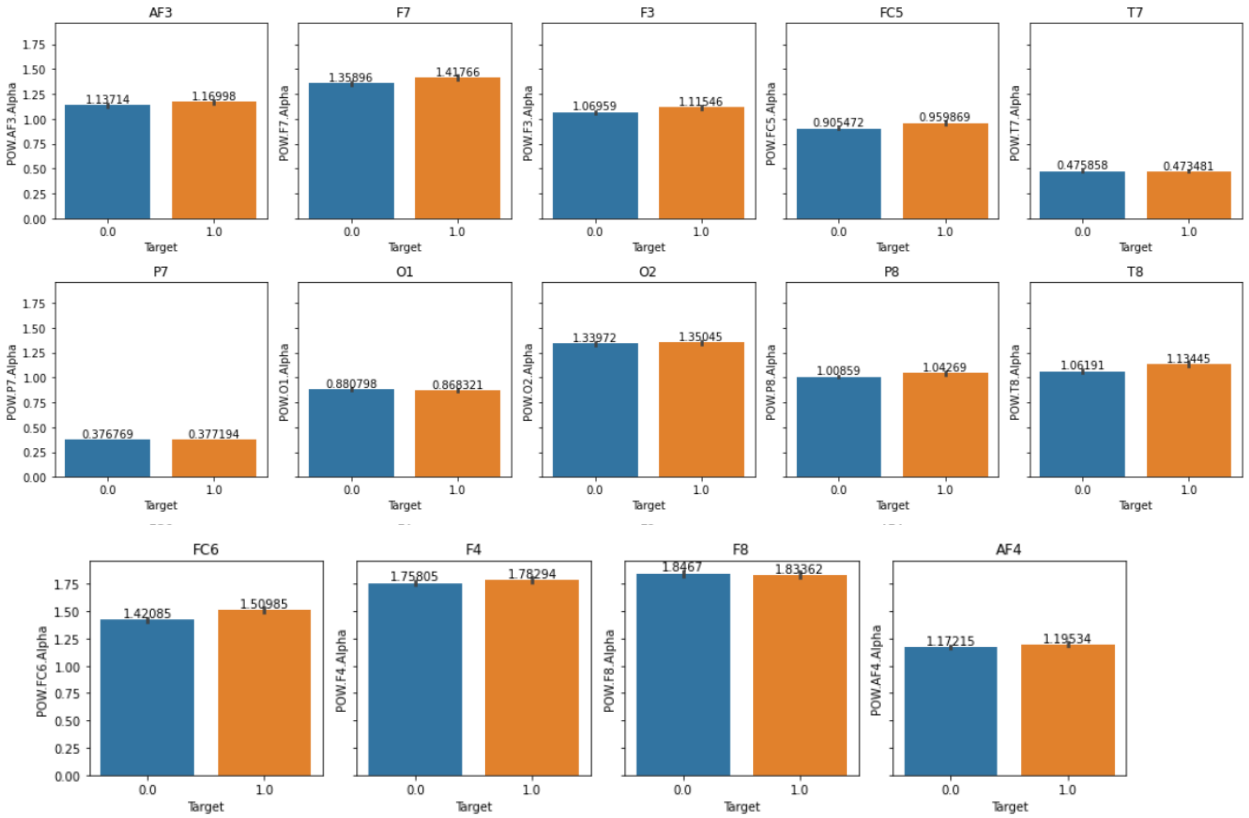


Figure 5.1.3.1: Barplot for average Alpha values for each sensor split based on Target

Observations:

- AF3, F7, F3, FC5, O2, P8, T8, FC6, F4 and AF4 show an increase in average Alpha value from Target = 0 to Target = 1
- T7 and P7 show around the same average Alpha value from Target = 0 to Target = 1
- O1 and F8 show a decrease in average Alpha value from Target = 0 to Target = 1

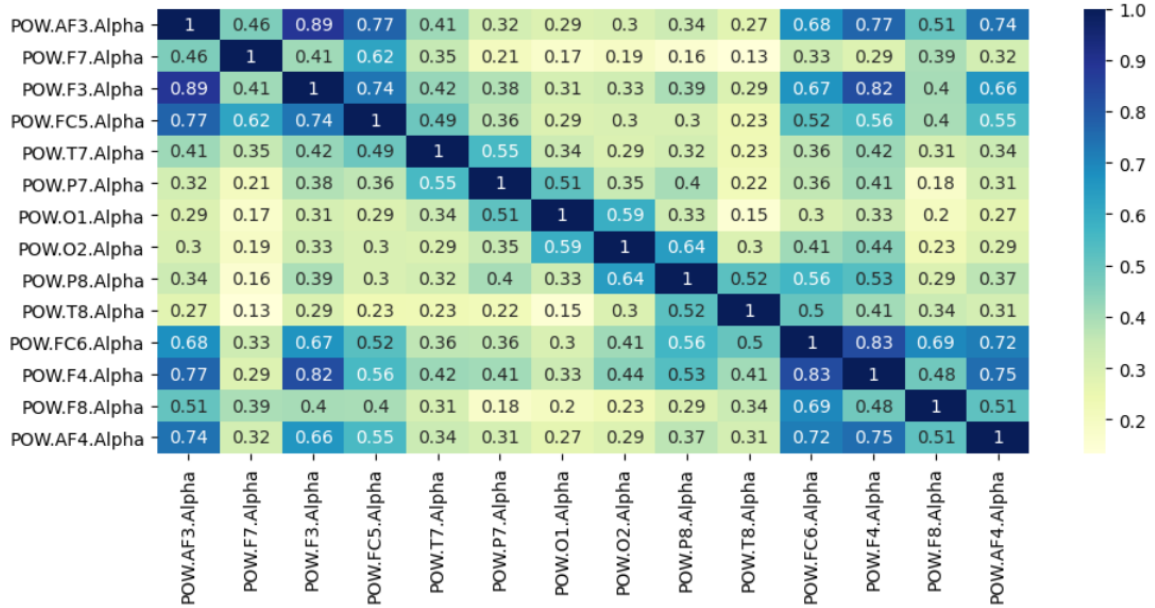


Figure 5.1.3.2: Correlation heatmap for Alpha values for each sensor

Observations:

- High correlation between:
    - F3 and AF3 (89%)
    - FC6 and F4 (83%)
    - F3 and F4 (82%)
- Many of the other sensors are not highly correlated.

BetaL

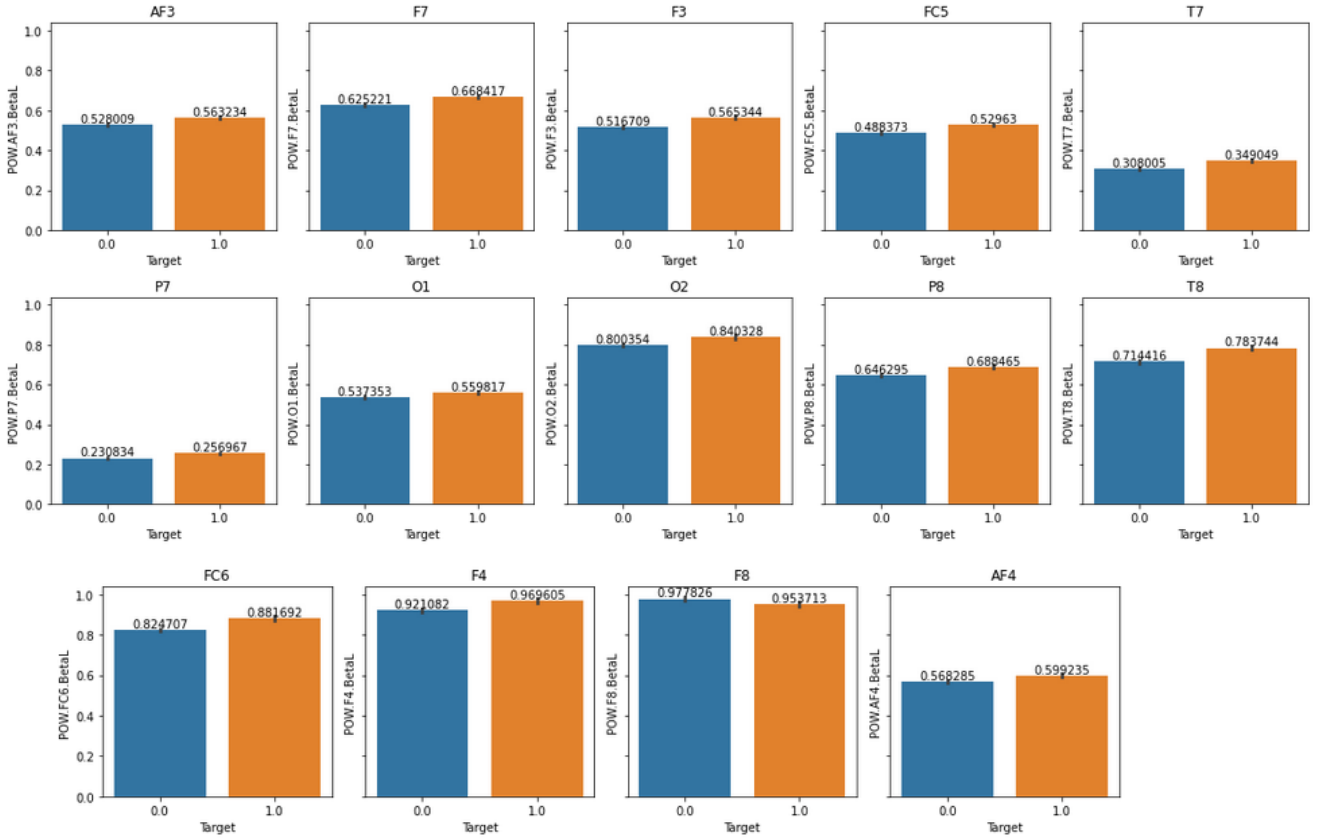


Figure 5.1.3.3: Barplot for average betaL values for each sensor split based on Target

Observations:

- AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4 and AF4 (all sensors except for F8) show an increase in average BetaL values from Target = 0 to Target = 1
- F8 shows a decrease in average BetaL value from Target = 0 to Target = 1

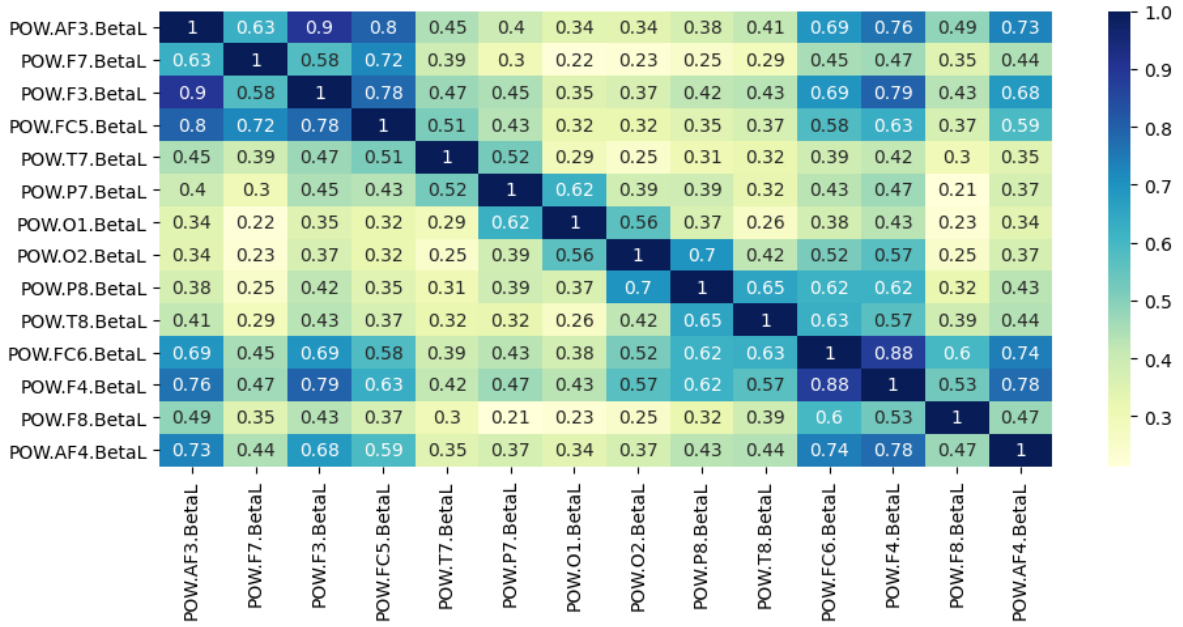


Figure 5.1.3.4: Correlation heatmap for BetaL values for each sensor

Observations:

- High correlation between:
  - F3 and AF3 (90%)
  - FC6 and F4 (88%)
  - FC5 and AF3 (80%)
- Many of the other sensors are not highly correlated.

BetaH

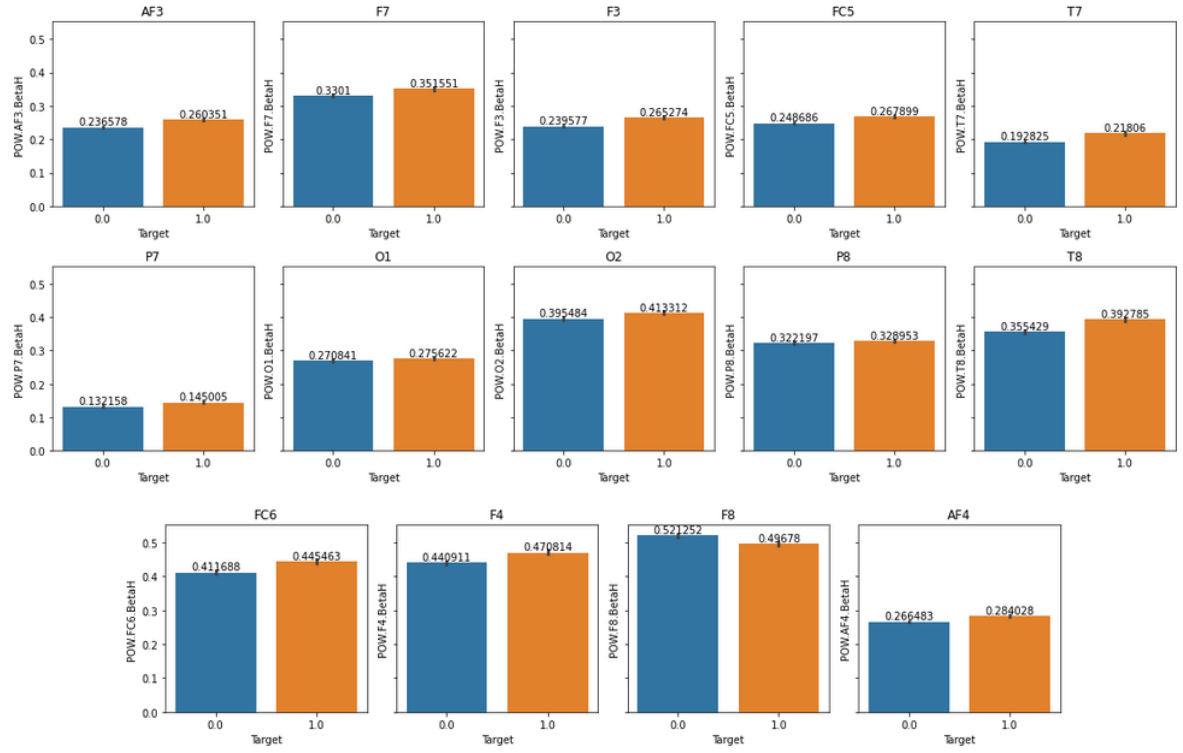


Figure 5.1.3.5: Barplot for average BetaH values for each sensor split based on Target

Observations:

- AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4 and AF4 (all sensors except for F8) show an increase in average BetaH values from Target = 0 to Target = 1
- F8 shows a decrease in average BetaH value from Target = 0 to Target = 1

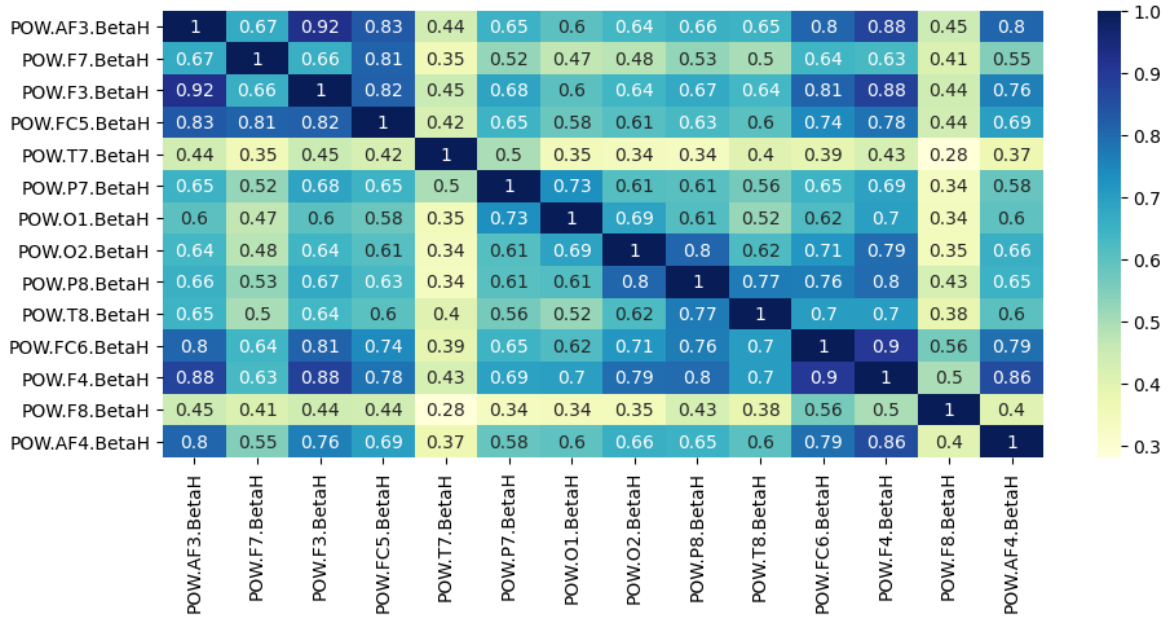


Figure 5.1.3.6: Correlation heatmap for BetaH values for each sensor

Observations:

- High correlation between BetaH values for:
    - F3 and AF3 (92%)
    - FC6 and F4 (90%)
    - F4 and AF3 (88%)
    - F4 and F3 (88%)
    - AF4 and F4 (86%)
    - FC5 and AF3 (83%)
    - FC5 and F3 (82%)
    - FC5 and F7 (81%)
    - FC6 and F3 (81%)
    - P8 and F4 (80%)
    - FC6 and AF3 (80%)
    - AF4 and AF3 (80%)

- Many of the other sensors are quite correlated, in the 60-70% range
- F8 and T7 show a lack of correlation with other sensors in BetaH values

Even though T7 shows an increase in average BetaH values like many of the other correlated sensors, from Figure 5.1.3.5, it seems like perhaps its values do not move in tandem with the others.

Gamma

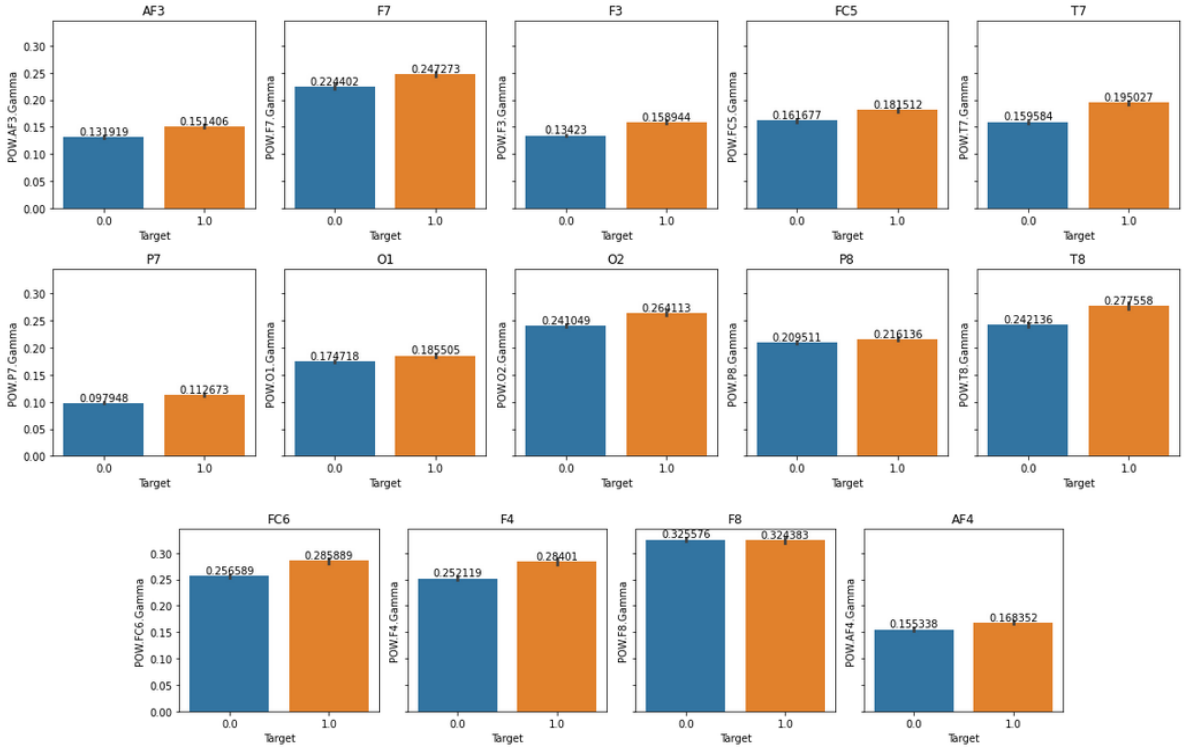


Figure 5.1.3.7: Barplot for average Gamma values for each sensor split based on Target

Observations:

- AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4 and AF4 (all sensors except for F8) show an increase in average Gamma values from Target = 0 to Target = 1
- F8 shows a decrease in average Gamma value from Target = 0 to Target = 1

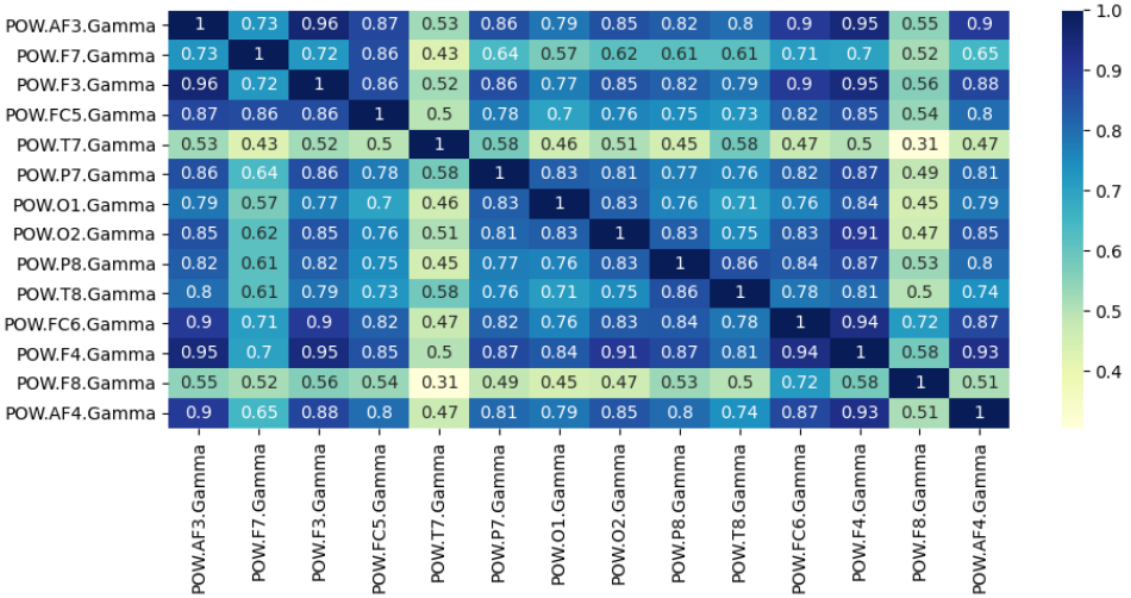


Figure 5.1.3.8: Correlation heatmap for Gamma values for each sensor

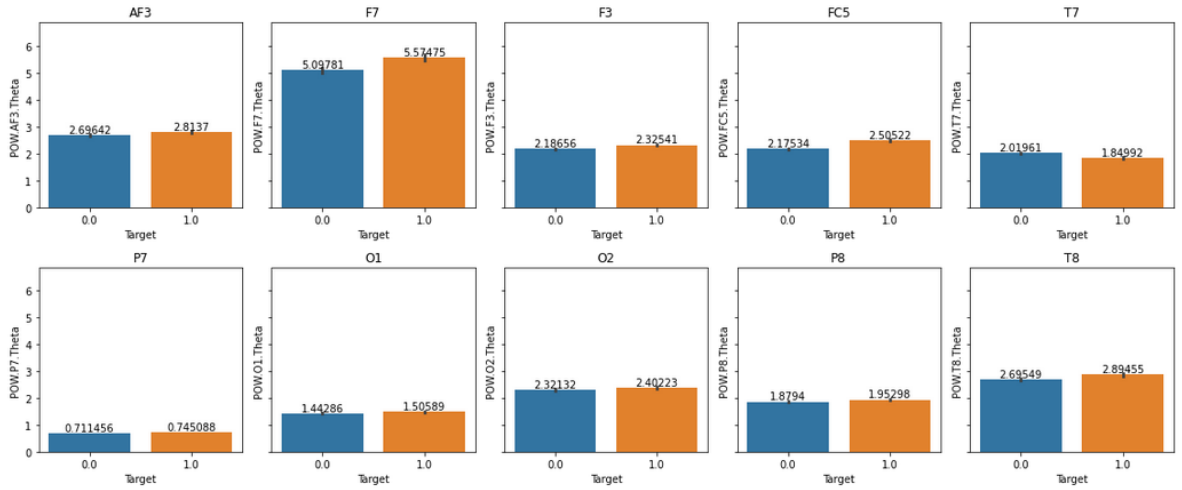
Observations:

- Highest correlations between Gamma values for:
    - F3 and AF3 (96%)
    - F4 and AF3 (95%)
    - FC6 and F4 (94%)
    - AF4 and F4 (93%)
    - F4 and O2 (91%)
    - F4 and F3 (90%)
    - FC6 and F3 (90%)
    - FC6 and AF3 (90%)
    - AF4 and AF3 (90%)

- Many of the sensors are highly correlated, in the 70-80% range
- F8 and T7 show a lack of correlation with other sensors in Gamma values

Even though T7 shows an increase in average Gamma values like many of the other correlated sensors, from Figure 5.1.3.7, it seems like perhaps its values do not move in tandem with the others.

Theta



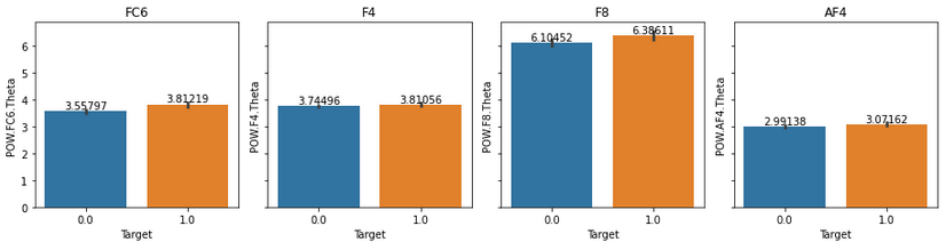


Figure 5.1.3.9: Barplot for average Theta values for each sensor split based on Target

Observations

- AF3, F7, F3, FC5, F8, P7, O1, O2, P8, T8, FC6, F4 and AF4 (all sensors except for T7) show an increase in average Theta values from Target = 0 to Target = 1
- T7 shows a decrease in average Theta value from Target = 0 to Target = 1

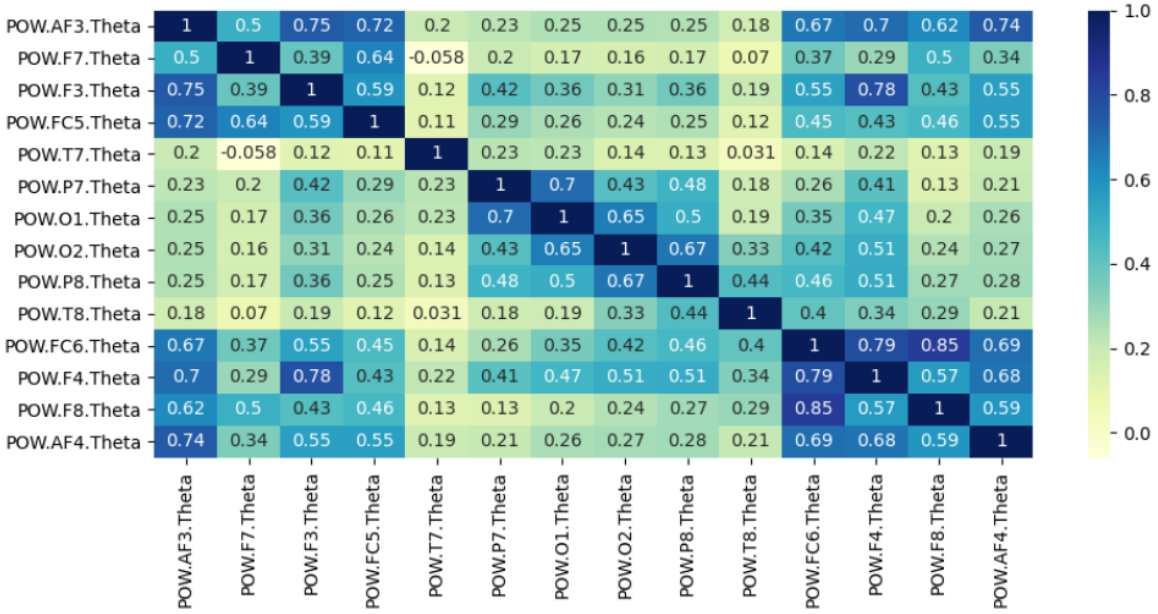


Figure 5.1.3.10: Correlation heatmap for Theta values for each sensor

Observations:

- High correlation between:
  - F8 and FC6 (85%)
- Many of the other sensors are not highly correlated, only with some in the 60-70% range, but most still lower.

Key insights

- Top few correlated sensors (correlation is frequently higher than other sensors)
  - F3 and AF3 (High correlation for Alpha, BetaL, BetaH, Gamma)
  - FC6 and F4 (High correlation for Alpha, BetaL, BetaH, Gamma)

- While the majority of the sensors showed a similar pattern with the increase in average Alpha, BetaL, BetaH, Gamma and Theta values, this does not always correspond to their correlation scores. This means that the increase in one type of wave value for each sensor from Target = 0 to Target = 1 is often not unanimous, which would make it harder to predict.

- There are a number of sensors who have a high correlation to other sensors for the different wave values. Due to this, we will be attempting Principal Component Analysis (PCA) to reduce the number of attributes into several new attributes which still capture most of the variation in the dataset. This will be explained further in the later sections.

Section 5.1.4: Comparison with Luqman's dataset

Just to gain some insights into whether the PM values or waves vary from individual to individual for stimulating and non-stimulating songs, we explored Luqman's dataset to do some comparisons:

1. PM barplot
2. All waves barplot

PM Barplot

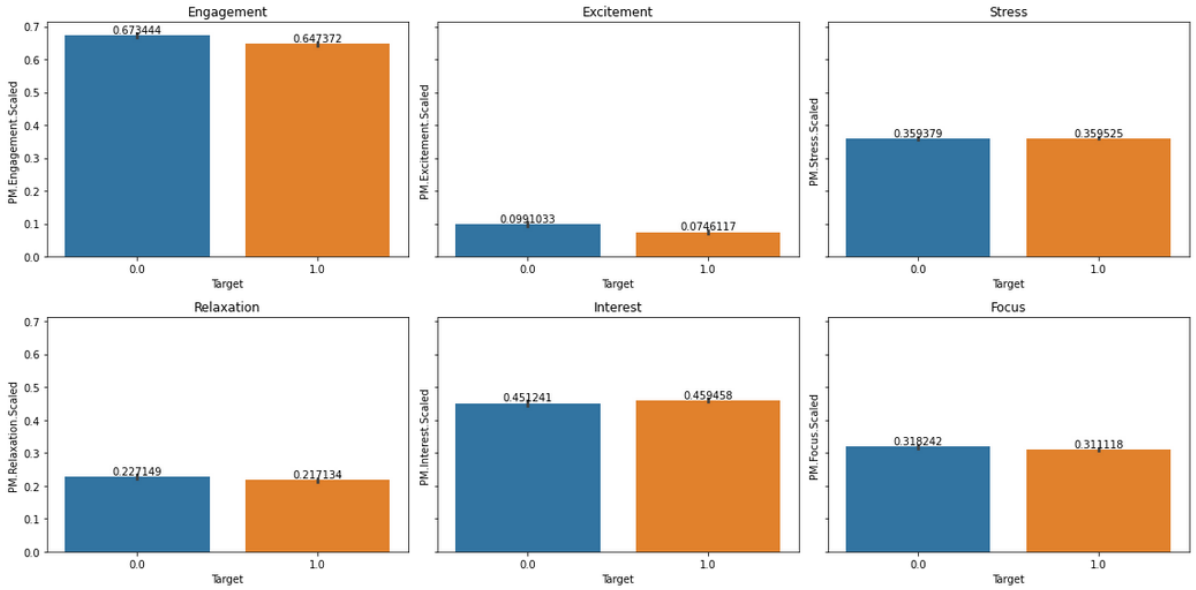


Figure 5.1.4.1: Barplot for average PM values split based on Target (Luqman's dataset)

Compared to Yu Xiang's dataset in Figure 5.1.1.1:

- The average Excitement values are lower (Target 0 - 53% lower, Target 1 - 61.5% lower)
- The average Focus values are lower (Target 0 - 26% lower, Target 1 - 26% lower)
- Average Engagement, Stress, Relaxation and Interest are around the same range

All waves barplot

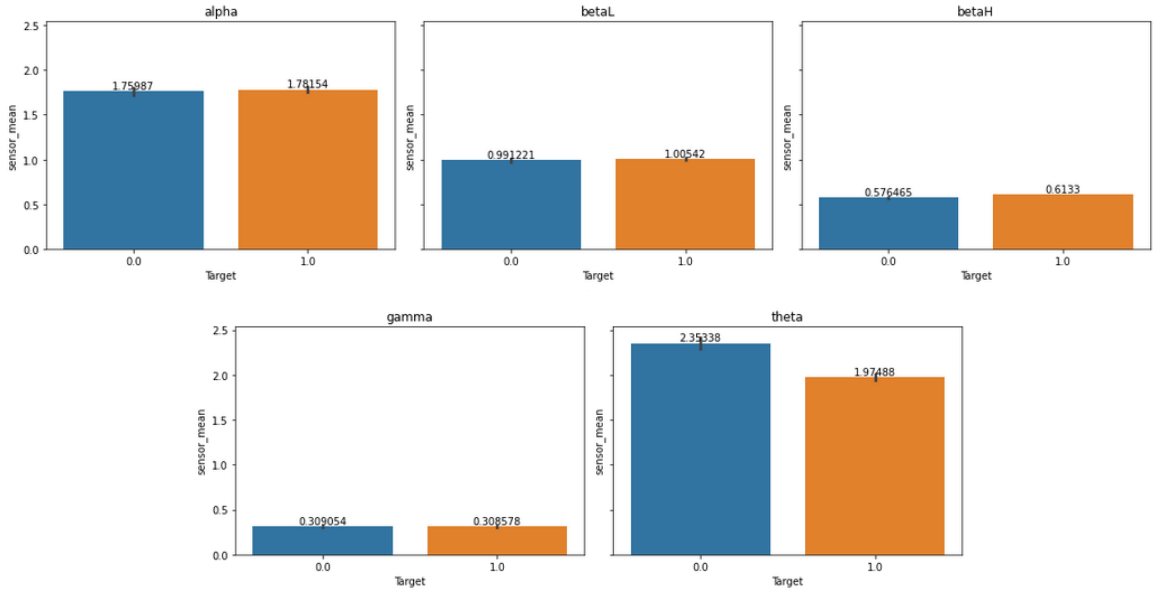


Figure 5.1.4.2: Barplot for average wave values split based on Target (Luqman's dataset)

Compared to Yu Xiang's dataset in Figure 5.1.2.1:

- The average Alpha values are much higher (Target 0 - 55.8% higher, Target 1 - 53.7% higher)
- The average BetaL values are much higher (Target 0 - 59.7% higher, Target 1 - 52.8% higher)
- The average BetaH values are much higher (Target 0 - 84.9% higher, Target 1 - 86% higher)
- The average Gamma values are much higher (Target 0 - 56.4% higher, Target 1 - 41.5% higher)
- The average Theta values are lower (Target 0 - 16.8% lower, Target 1 - 33.6% lower)
- Theta shows an opposite trend where it drops significantly from Target = 0 to Target = 1

All waves (by sensor)

After plotting all the bar plots and correlation heatmaps for luqman's dataset, we compared the values with Yu Xiang's dataset and found that there were differences in the average values for some sensors, and correlation between sensors. For example, the average values recorded by the F4 sensor for Alpha waves were higher for Yu Xiang compared to Luqman (shown in Figures 5.1.4.3 and 5.1.4.4). Most sensors were also not highly correlated for Gamma waves in Luqman's dataset, while they were in Yu Xiang's (shown in Figures 5.1.4.5 and 5.1.4.6).

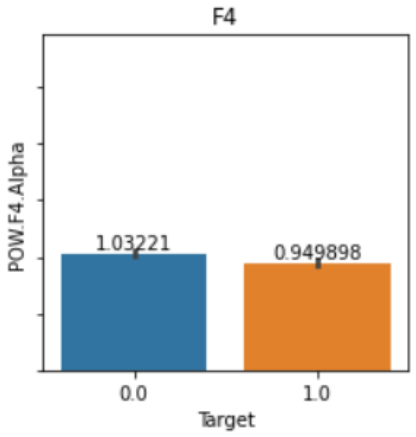


Figure 5.1.4.3: Luqman's dataset sensor F4 average value for Alpha

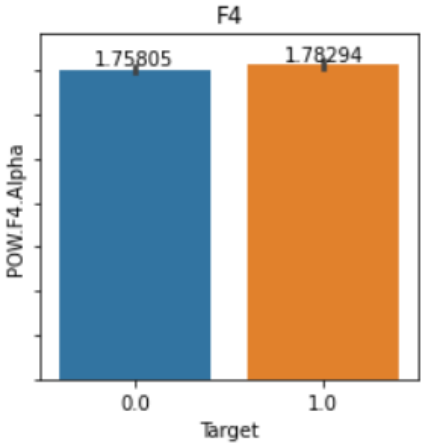


Figure 5.1.4.4: Yu Xiang's dataset F4 average value for Alpha

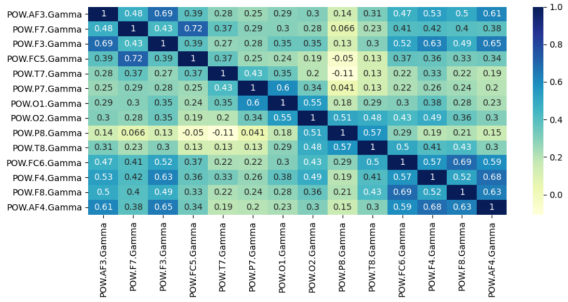


Figure 5.1.4.5: Luqman's dataset sensor value correlations for Gamma

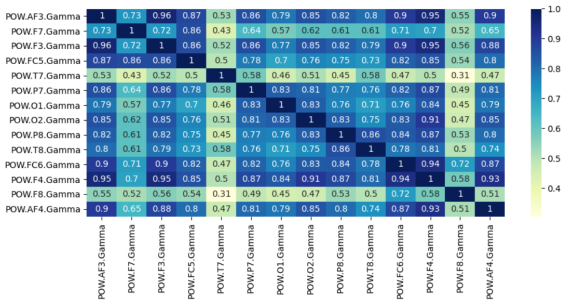


Figure 5.1.4.6: Yu Xiang's dataset sensor value correlations for Gamma

This shows that there are likely to be significant differences between one person's brain waves and sensor values, which may make models hard to generalise. However, there were still many similarities present between the two, which may still be sufficient for the model to make generalised predictions.

Section 6.0 Model Building

Section 6.1: Considerations for Model Building

As mentioned above, the team planned to implement 11 classification models. All of these are implemented in our jupyter notebook, under the runmodels function.

Before training the dataset, there were a few things that we had to account for before the actual training of the model:

(1) Datasets which the model should be trained on (different team member's datasets)
(2) Applicability of feature extraction/engineering
(3) How the training and test datasets will be separated

For (1), we needed to decide on the different combinations of datasets we wanted to run our models on. Ultimately, different combinations could lead to different accuracies of the model and may provide certain insights. For (2), if a larger number of attributes were to be input into the model, this could result in long running times and unnecessary attributes may also add to the noise in the data. For (3), we needed to decide on how to separate the data from different songs between the training and test datasets.

Section 6.1.1: Dimensionality reduction of the pre-processed dataset

As a team, we decided that the training of the models should be done once on each of the datasets prepared: the Performance Metrics (PM), Beta waves as well as all the other waves.

Due to the large number of features present on the beta waves dataset and all other waves dataset (33 columns for beta and 77 for all other waves), we searched for a dimensionality reduction technique, and found one known as the Principle Component Analysis (PCA). PCA is often used as a method to reduce the large dataset due to the large number of features/columns present, by accounting for the largest possible variance in the dataset and at the same time reducing the number of features available. It is applicable when there are many correlated attributes, which applies to our case as shown in section 5.1.3. This dimensionality reduction will help to reduce the complexity and computational time required to develop our models, while reducing noise from irrelevant attributes.

Through the PCA library on sk-learn, PCA was performed on the training data of the beta waves and other waves. For all datasets, we decided to select relevant Principle Components that explained up to 95% variance of the dataset to train the model, capturing most of the important patterns which will help predictions.

Section 6.1.2: Model Validation

Model validation was key in order to prevent the model from overfitting especially to the training dataset, making the models employed not generalisation as a whole.

Train-test-split

Initially our team performed a train-test-split validation on the relevant combinations provided based on a 80% to 20% split. Songs were split randomly by the train-test-split validation based on the rows of the dataset. As such, there is a possibility that a particular song could have some of its rows being in the training set and some of it being in the testing set. Our team decided to branch out into manual song validation (which would be discussed in the next part), where we aggregated songs and ensure that songs that were in the training set did not have rows in the testing set.

Manual validation testing via song aggregation

Based on the combinations of dataset which is used to train the model (which we will talk about in section 5.2), the training data will be chosen based on the song. Songs belonging to train will only belong to train and would not reappear again in the test set. The training data of the model has to account for approximately 80% of the rows used in the dataset, based on the particular combination chosen. Likewise, this is the same for the test data as well, and this would be approximately 20% of the entire dataset chosen. As it is hard to split the dataset into a 80-20 ratio (For training and testing data), our team decided to set a threshold of about +- 5% range within 80% and 20% . This means that if the training data accounts for 85% of the rows in the dataset, this was fine because each song had different rows to it. By doing this, it granted us much needed flexibility, which made the testing of

the models much easier. At the same time, the way which we aggregate and split the songs would ensure that there would be no spillover of rows between the testing and the training set, and this would ensure that our test set is completely new

Section 6.2: Model Training

The training of the 11 baseline models was first trained with the train-test-split validation technique followed by a manual validation of the model via song aggregation.

Section 6.2.1: Model training with Train-test-split validation

A total of 2 models are trained with different combinations of dataset from our team members, with accuracy and F1-score as the overall evaluation metric being used and the validation method used here being a train-test-split validation. Datasets are being validated a subjected to a train-test-split of 80% training and 20% testing

Model 1 (Combination 4): 36 Songs (Yu Xiang)
Initially the team first wanted to test how the baseline models would perform with a train-test-split conducted only on Yu Xiang. The results from our findings can be seen in the table below.

<table>
<tr><th colspan="3">Performance</th></tr>
<tr><td>Performance metric</td><td>Beta waves (BetaH and BetaL)</td><td>All Other Waves</td></tr>
<tr>
<td>

	Accuracy Score	F1 Score
ExtraTreesClassifier	0.940249	0.916084
Random Forest	0.917012	0.881797
ExtremeGradientBoost	0.885477	0.839161
BaggingClassifier	0.882988	0.825279
KNN	0.863900	0.811927
Decision Tree Classifier	0.827386	0.760369
Gradientboost	0.730290	0.512744
Gaussian NB	0.596680	0.494802
Adaboost	0.655602	0.383358
SVM	0.643154	0.107884
Logistic Regression	0.633195	0.034934

</td>
<td>

	Accuracy Score	F1 Score
ExtraTreesClassifier	0.910507	0.859196
KNN	0.898048	0.853477
Random Forest	0.874377	0.793445
BaggingClassifier	0.823505	0.704656
ExtremeGradientBoost	0.813123	0.700997
Decision Tree Classifier	0.736296	0.632948
Gradientboost	0.686254	0.336408
Adaboost	0.660507	0.330193
Gaussian NB	0.612334	0.306206
SVM	0.691030	0.283927
Logistic Regression	0.649502	0.173359

</td>
<td>

	Accuracy Score	F1 Score
ExtraTreesClassifier	0.843231	0.731603
KNN	0.794228	0.693474
Random Forest	0.822051	0.689830
BaggingClassifier	0.775540	0.615986
Decision Tree Classifier	0.702658	0.582264
ExtremeGradientBoost	0.724045	0.523485
Gradientboost	0.668812	0.230584
Adaboost	0.646387	0.196319
SVM	0.666321	0.182188
Gaussian NB	0.638912	0.173086
Logistic Regression	0.647425	0.094883

</td>
</tr>
<tr>
<td>Best accuracy score
● Extra Trees Classifier (94.0%)
Best F1-score
● Extra Trees Classifier (91.6%)</td>
<td>Best accuracy score
● Extra Trees Classifier (91.0%)
Best F1-score
● Extra Trees Classifier (85.9%)</td>
<td>Best accuracy score
● Extra Trees Classifier (84.3%)
Best F1-score
● Extra Trees Classifier (73.1%)</td>
</tr>
</table>

Insights

- In general, ensemble learning techniques seem to outperform single machine learning classifiers
- From the baseline modelling, ExtraTreesClassifier performed the best amongst the 3 datasets performed
- Random forest is always within the top 3 classifier for each of the different datasets used

The accuracy and F1-scores for the models were good, particularly so for the ExtraTreesClassifier model. However, since this is only tested based on one person, which may be overfitting in a sense, our team wanted to include Luqman's dataset in, in order to test the generalizability of baseline models being trained. This led to the testing of our next model.

Model 2 (Combination 5): 6 songs Yu Xiang, 6 songs Luqman

Performance

Performance metric	Beta waves (BetaH and BetaL)	All Other Waves

	Accuracy Score	F1 Score
ExtraTreesClassifier	0.953757	0.962963
Random Forest	0.927746	0.941995
ExtremeGradientBoost	0.916185	0.933025
KNN	0.910405	0.926714
BaggingClassifier	0.893064	0.911271
Gradientboost	0.861272	0.888889
Decision Tree Classifier	0.858382	0.884161
Adaboost	0.736994	0.797327
SVM	0.638728	0.763705
Logistic Regression	0.618497	0.754647
Gaussian NB	0.598266	0.734226

	Accuracy Score	F1 Score
ExtraTreesClassifier	0.929964	0.942570
Random Forest	0.872202	0.897983
KNN	0.854152	0.876225
ExtremeGradientBoost	0.820217	0.852924
BaggingClassifier	0.797112	0.829179
SVM	0.685199	0.770285
Decision Tree Classifier	0.707581	0.759072
Gradientboost	0.673646	0.755146
Logistic Regression	0.620217	0.729145
Gaussian NB	0.612996	0.710270
Adaboost	0.628881	0.710259

	Accuracy Score	F1 Score
ExtraTreesClassifier	0.907581	0.926267
Random Forest	0.864260	0.892816
ExtremeGradientBoost	0.821661	0.858453
KNN	0.826715	0.855247
BaggingClassifier	0.796390	0.826141
Decision Tree Classifier	0.727798	0.774387
Gradientboost	0.693863	0.773987
SVM	0.677978	0.770576
Logistic Regression	0.627437	0.742515
Adaboost	0.638989	0.732334
Gaussian NB	0.615162	0.726807

Best accuracy score
- Extra Trees Classifier (95.3%)

Best F1-score
- Extra Trees Classifier (96.2%)

Best accuracy score
- Extra Trees Classifier (92.9%)

Best F1-score
- Extra Trees Classifier (94.2%)

Best accuracy score
- Extra Trees Classifier (90.7%)

Best F1-score
- Extra Trees Classifier (92.6%)

Insights
- In general, ensemble learning techniques seem to outperform single machine learning classifiers
- From the baseline modelling, ExtraTreesClassifier performed the best amongst the 3 datasets performed
- Random forest is always within the top 3 classifier for each of the different datasets used
- Additionally, the team noted that Model 2 performed better than Model 1, in terms of its overall F1-score and accuracy score

The accuracy and F1-scores for the models were good, particularly so for the ExtraTreesClassifier model. At the same time, our team considered branching out into manual validation testing, as one reason which we deduced why the models performed better was mainly due to the fact that some rows in the training dataset could be in the testing dataset as well. Our team wanted to figure out a way to aggregate songs, for training and testing sets, with no overlap in between.

Section 6.2.2: Model training with Manual Validation Testing

As mentioned above in section 5.1.2, manual validation testing is being used to provide a clear segregation between the training and testing dataset. We wanted to prevent an over spillage of training and testing data (E,g having rows training data in the testing data), as we thought that this could potentially overfit our models that are being trained.

A total of 4 models were subjected to the manual validation. Likewise just like in section 5.2.1, different combinations of datasets were used from members in our team, with accuracy and F1-score as the overall evaluation metric being used

Model 3 (Combination 1): 30 songs Yu Xiang, 6 songs Yu Xiang

<table>
<tr><th colspan="3">Performance</th></tr>
<tr><td>Performance metrics</td><td>Beta waves (BetaH and BetaL)</td><td>All Other Waves</td></tr>
<tr>
<td>

	Accuracy Score	F1 Score
Gaussian NB	0.559456	0.460472
KNN	0.522084	0.407303
Decision Tree Classifier	0.468856	0.356653
ExtremeGradientBoost	0.515289	0.347561
Adaboost	0.499434	0.338323
Random Forest	0.506229	0.278146
ExtraTreesClassifier	0.505096	0.265546
BaggingClassifier	0.467724	0.234528
Gradientboost	0.510759	0.231317
Logistic Regression	0.516421	0.000000
SVM	0.516421	0.000000

</td>
<td>

	Accuracy Score	F1 Score
Decision Tree Classifier	0.511224	0.387468
KNN	0.522704	0.355494
ExtremeGradientBoost	0.548980	0.298969
BaggingClassifier	0.552806	0.231477
Random Forest	0.552296	0.168640
ExtraTreesClassifier	0.561735	0.143569
Adaboost	0.550510	0.142162
SVM	0.558673	0.098958
Gradientboost	0.553061	0.068085
Gaussian NB	0.553316	0.044735
Logistic Regression	0.554337	0.026741

</td>
<td>

	Accuracy Score	F1 Score
Decision Tree Classifier	0.506122	0.381074
KNN	0.522704	0.355494
ExtremeGradientBoost	0.548980	0.298969
BaggingClassifier	0.537500	0.224882
Random Forest	0.557143	0.167785
Adaboost	0.550510	0.142162
ExtraTreesClassifier	0.562500	0.128114
SVM	0.558673	0.098958
Gradientboost	0.553061	0.068085
Gaussian NB	0.553316	0.044735
Logistic Regression	0.554337	0.026741

</td>
</tr>
<tr>
<td>
Training rows:
• 5141 (86.4%)
Testing rows
• 883 (14.6%)

Best accuracy score
• Gaussian NB (55.9%)
Best F1-score
• Gaussian NB (46.0%)
</td>
<td>
Training rows:
• 20157 (84.8%)
Testing rows
• 3920 (16.2%)

Best accuracy score
• Extra Trees Classifier (56.1 %)
Best F1-score
• Decision Tree Classifier (38.7%)
</td>
<td>
Training rows:
• 20157 (84.8%)
Testing rows
• 3920 (16.2%)

Best accuracy score
• Extra Trees Classifier (56.2%)
Best F1-score
• Decision Tree Classifier (38.1%)
</td>
</tr>
</table>

Insights

- It is hard to say which dataset performed the best here, both evaluation metrics of F1-score and accuracy were roughly similar across all datasets
- Ensemble techniques seem to be better in terms of accuracy but this is not the case for the F1-score. An example such would be the ExtraTreesClassifier model that was trained using the Other Waves dataset. While it had the best accuracy, the F1-score for this classifier was clearly not the best in this case, yielding a score of 12.8%.

As the models built based on Yu Xiang data of Performance Metric, Beta Waves and Other Waves were bad in terms of accuracy and F1-scores, our team wanted to see how the models would be

generalisable if it was tested on another person's dataset. In the next model, we decided to train the baseline models on Yuxiang data, and test it on Luqman data.

Model 4 (Combination 2): 36 songs Yu Xiang, 6 songs Luqman

In this combination, the baseline models are trained on the dataset of 36 songs of Yu Xiang (Performance Metrics, Beta Waves, All Other Waves), and tested on 6 songs of Luqman. The purpose of this is that we wanted to investigate and find out whether the baseline model train on one particular person is generalisable on another person. The performance of the baseline model on the 3 different datasets of features can be found below.

Performance

Performance metrics

	Accuracy Score	F1 Score
Gaussian NB	0.555468	0.643982
Decision Tree Classifier	0.481511	0.474062
KNN	0.475216	0.451931
BaggingClassifier	0.458694	0.397548
ExtremeGradientBoost	0.490165	0.390977
Random Forest	0.474430	0.386029
Adaboost	0.394965	0.370188
ExtraTreesClassifier	0.468135	0.356190
Gradientboost	0.430370	0.309160
SVM	0.419355	0.068182
Logistic Regression	0.410700	0.036036

Training rows:
- 6024 (82.6%)

Testing rows
- 1271 (17.4%)

Best accuracy score
- Gaussian NB (55.5%)

Best F1-score
- Gaussian NB (64.3%)

Beta waves (BetaH and BetaL)

	Accuracy Score	F1 Score
Logistic Regression	0.654590	0.773243
SVM	0.546933	0.671443
KNN	0.496706	0.582266
ExtremeGradientBoost	0.518320	0.580194
Decision Tree Classifier	0.454920	0.488606
BaggingClassifier	0.439893	0.425343
Adaboost	0.428160	0.395035
Random Forest	0.428983	0.361712
Gradientboost	0.412310	0.286072
Gaussian NB	0.396871	0.244066
ExtraTreesClassifier	0.385755	0.224935

Training rows:
- 24077 (83.3%)

Testing rows
- 4858 (16.7%)

Best accuracy score
- Logistic Regression (65.4%)

Best F1-score
- Decision Tree Classifier (77.3%)

All Other Waves

	Accuracy Score	F1 Score
Decision Tree Classifier	0.510704	0.584803
ExtremeGradientBoost	0.478386	0.506043
KNN	0.470152	0.503472
BaggingClassifier	0.423425	0.386418
Random Forest	0.420338	0.349654
SVM	0.403252	0.246034
Gradientboost	0.384726	0.209886
ExtraTreesClassifier	0.374846	0.186009
Logistic Regression	0.388843	0.185906
Adaboost	0.370935	0.145892
Gaussian NB	0.363936	0.132510

Training rows:
- 24077 (83.3%)

Testing rows
- 4858 (16.7%)

Best accuracy score
- Decision Tree Classifier (51.0%)

Best F1-score
- Decision Tree Classifier (58.4%)

Insights

- Model 4 performed better overall as compared to model 3 in terms of its baseline machine learning models accuracy and F1-score
- The team felt puzzled as to why this was the case (Better accuracy and F1-score) as we felt that the baseline models in model 4 should be less generalizable, and it should overall obtain a lower accuracy and F1-score, since different people would produce different varying lengths of brainwaves.
- Additionally, we noted that models which produced the best F1-score were actually single machine learning models, and not ensemble models, which we anticipated to perform slightly better as well, which was not the case here.

Currently, the baseline models are being trained to the 36 songs by Yu Xiang, which could potentially be overfitted with Yu Xiang during the training of the baseline models. This would lead us to the training of our next model, which we will try to overcome this form of biasness.

Model 5 (Combination 3): 6 songs Yu Xiang, 6 songs Luqman

In this combination, the baseline models are trained on the dataset of 5 songs of Yu Xiang together with 5 songs of Luqman (Performance Metrics, Beta Waves, All Other Waves), and tested on 1 song from Luqman and another on Yu Xiang. While the training of the dataset is much smaller, we wanted to see whether the effect of having training data would potentially help to improve the evaluation metrics used in over models

<table>
<tr><th colspan="3">Performance</th></tr>
<tr><td>Performance metrics</td><td>Beta waves (BetaH and BetaL)</td><td>All Other Waves</td></tr>
<tr>
<td>

	Accuracy Score	F1 Score
Decision Tree Classifier	0.532258	0.647202
KNN	0.525806	0.614173
BaggingClassifier	0.496774	0.573770
Logistic Regression	0.393548	0.520408
ExtremeGradientBoost	0.400000	0.502674
ExtraTreesClassifier	0.377419	0.437318
Gradientboost	0.358065	0.426513
Random Forest	0.374194	0.408537
SVM	0.303226	0.341463
Gaussian NB	0.248387	0.339943
Adaboost	0.296774	0.318750

</td>
<td>

	Accuracy Score	F1 Score
Random Forest	0.556489	0.648928
ExtraTreesClassifier	0.549020	0.648216
SVM	0.545285	0.633007
Logistic Regression	0.542484	0.628225
Decision Tree Classifier	0.523810	0.615964
Gradientboost	0.521942	0.609160
KNN	0.517274	0.595778
ExtremeGradientBoost	0.507937	0.594927
Adaboost	0.507003	0.581616
BaggingClassifier	0.477124	0.551282
Gaussian NB	0.432306	0.444241

</td>
<td>

	Accuracy Score	F1 Score
Gaussian NB	0.672269	0.760082
SVM	0.557423	0.650957
Logistic Regression	0.546218	0.644217
ExtraTreesClassifier	0.533147	0.635036
Random Forest	0.535948	0.628827
Gradientboost	0.526611	0.622487
Adaboost	0.513539	0.610904
KNN	0.527544	0.599049
ExtremeGradientBoost	0.500467	0.596835
Decision Tree Classifier	0.492997	0.576112
BaggingClassifier	0.478992	0.547812

</td>
</tr>
<tr>
<td>

Training rows:
- 1299 (75.1%)

Testing rows
- 431 (24.9%)

Best accuracy score
- Decision Tree Classifier (53.2%)

Best F1-score
- Decision Tree Classifier (64.7%)

</td>
<td>

Training rows:
- 5850 (84.6%)

Testing rows
- 1071 (15.4%)

Best accuracy score
- Random Forest (55.6%)

Best F1-score
- Random Forest (64.8%)

</td>
<td>

Training rows:
- 5850 (84.6%)

Testing rows
- 1071 (15.4%)

Best accuracy score
- Gaussian NB (67.2%)

Best F1-score
- Gaussian NB (76.0%)

</td>
</tr>
</table>

Insights

- It would seem like the F1-scores and accuracies for model 5 is rather similar or close to F1-scores and accuracies in model 4
- Baseline models that were trained with All Other Waves dataset performed the best here in terms of both F1-score and accuracy score, obtaining values of 76.0% and 67.2% respectively with the Gaussian NB model. This is rather similar to the baseline models that ere Beta Waves dataset performed in model 4, with its Logistic Regression Model having the highest accuracy and F1-score of about 65.4% accuracy 77.3% F1-score.

As both Models 4 and 5 were rather similar in scores, and tying it the problem of not understanding why Model 3 performed worse amongst all our models, our team came out with a relevant hypothesis.

Hypothesis formulation

- Model 4 (36 Yuxiang; 6 Luqman) and model 5 (6 Yuxiang; 6 Luqman) had an overall higher accuracy and F1-score than model 3.
- As Luqman and Yu Xiang have the relatively similar overall taste in songs, it could be possible that brain waves produced by them could be similar in wavelength, which resulted in better test accuracy results
- One theory could be that the baseline models could be trained with similar brainwave values which resulted in an overall higher than expected accuracy and F1-score.

To confirm this hypothesis, our team decided to investigate this by training another model to account for the significance of these 6 songs being used.

Model 6 (Combination 6): 30 songs Yu Xiang, 6 songs Luqman

As mentioned in the model 5, model 6 If our thesis were to be correct, the performance of the models should be overall much worse as compared to model 4 and 5. The results of our findings are as shown below.

<table>
<tr><th colspan="3">Model 6</th></tr>
<tr><td>Performance metrics</td><td>Beta waves (BetaH and BetaL)</td><td>All Other Waves</td></tr>
<tr>
<td>

	Accuracy Score	F1 Score
Gaussian NB	0.550747	0.642902
Decision Tree Classifier	0.509835	0.492258
KNN	0.453973	0.421667
ExtremeGradientBoost	0.484658	0.376784
Random Forest	0.453186	0.323272
BaggingClassifier	0.446892	0.296296
ExtraTreesClassifier	0.435090	0.294695
Adaboost	0.372935	0.290294
Gradientboost	0.423289	0.280667
SVM	0.416994	0.067925
Logistic Regression	0.409127	0.035944

</td>
<td>

	Accuracy Score	F1 Score
Logistic Regression	0.652326	0.770921
SVM	0.545080	0.670248
Decision Tree Classifier	0.513998	0.583965
KNN	0.495060	0.581042
ExtremeGradientBoost	0.480445	0.519787
BaggingClassifier	0.429189	0.379225
Gaussian NB	0.425895	0.327141
Random Forest	0.411692	0.305299
Adaboost	0.387402	0.258595
Gradientboost	0.383697	0.230334
ExtraTreesClassifier	0.372787	0.163601

</td>
<td>

	Accuracy Score	F1 Score
Gaussian NB	0.650679	0.781568
Logistic Regression	0.656237	0.761156
KNN	0.481886	0.516983
Decision Tree Classifier	0.429807	0.421712
Adaboost	0.413956	0.319388
BaggingClassifier	0.385961	0.286193
Random Forest	0.396048	0.239896
ExtremeGradientBoost	0.383697	0.227155
SVM	0.397900	0.214343
Gradientboost	0.368053	0.172060
ExtraTreesClassifier	0.367641	0.156507

</td>
</tr>
<tr>
<td>
Training rows:
● 5883 (78.4%)
Testing rows
● 1271 (21.6%)

Best accuracy score
● Gaussian NB (55.0%)
Best F1-score
● Gaussian NB (64.2%)
</td>
<td>
Training rows:
● 23513 (79.4%)
Testing rows
● 4858 (20.6%)

Best accuracy score
● Logistic Regression (65.2%)
Best F1-score
● Logistic Regression (77.0%)
</td>
<td>
Training rows:
● 23513 (79.4%)
Testing rows
● 4858 (20.6%)

Best accuracy score
● Gaussian NB (65.0%)
Best F1-score
● Gaussian NB (78.1%)
</td>
</tr>
</table>

Insights

- It would seem like the F1-scores and accuracies for model 6 performed slightly better as compared to model 5
- Baseline modelling seemed to be rather generalisable to Luqman's dataset, especially for the Logistic Regression Model, which yielded the best F1-Score of 77%.

Section 7.0 Analysis of results

Section 7.1 Comparison between different dataset and training models

			Top F1 Score		
Model	Training Condition	Aggregation Method	PM	Beta Waves	All POW Waves
1	Yu Xiang's 36 Song	Train Test Split (80% - 20%)	0.916	0.859	0.732
			ETC	ETC	ETC
2	Combine both Yu Xiang's and Luqman 6 songs each	Train Test Split (80% - 20%)	0.958	0.943	0.926
			ETC	ETC	ETC
3	Train on Yuxiang's 30 songs, and test on Yuxiang's 6 songs (Aggregated by songs)	Manual Train Test Split (80%-20%)	0.460	0.379	0.381
			Gaussian NB	Decision Tree	Decision Tree
4	Train on 36 Yuxiang's song and test on 6 Luqman's songs (Aggregated by songs)	Manual Train Test Split (80%-20%)	0.644	0.773	0.585
			Gaussian NB	Logistic Reg	Decision Tree
5	Combine both Yuxiang and Luqman's 5 songs each and test on each one of our song (Aggregated by songs)	Manual Train Test Split (80%-20%)	0.647	0.649	0.760
			Decision Tree	Random Forest	Gaussian NB
6.	Train on 30 YuXiang's Songs and test on Luqman's 6 Songs	Manual Train Test Split (80% - 20%)	0.643	0.771	0.782
			Gaussian NB	Logistic Reg	Gaussian NB

The results will be analysed in pairs, according to differences in the combinations and aggregation method:

Comparisons	Analysis
Comparison 1: Model 1 & 3	**Main difference:** Aggregation method Model 1: Train-test Model 3: Manual train-test (aggregated by song) Both use the same set of data consisting of Yu Xiang's 36 songs so differences in performance should be due to the aggregation method. **Expected result: Model 1 performs better** In a train-test split, the model will likely be trained on at least a few rows from each song, while in the manual train test the model would not have been trained on any rows from the test songs. Model should be better at predicting the rows if it has seen other rows of the same song as they may be more similar in terms of what is stimulating and non-stimulating. **Actual result: Model 1 performs better (Matches expected)** Model 1 performs better (PM, 91.6%) than Model 3 (PM, 46%) Significant decrease in f1 score (45.6%). Suggests that it is possible the wave values that determine what is stimulating and non-stimulating are relative, differing from song to song and it becomes a lot harder to predict one song from another.
Comparison 2: Model 2 & 5	**Main difference:** Aggregation method Model 2: Train-test Model 5: Manual train-test (aggregated by song) Both use the same set of data consisting of Yu Xiang's 6 songs and Luqman's 6 songs. **Expected result: Model 2 performs better** *(justification similar to comparison 1)* **Actual result: Model 2 performs better (Matches expected)** ● Model 2 performs better (PM, 95.8%) than Model 5 (All POW waves, 76%) ● Decrease in f1 score (19.8%). Reinforces insights gained from comparison 1.

Comparison 3: Model 3 & 6	**Main difference:** Testing dataset Model 3: Train on Yu Xiang, tested on Yu Xiang Model 6: Train on Yu Xiang, tested on Luqman Both use the same set of data consisting of Yu Xiang's 30 songs for training so differences in performance should be due to the testing dataset used and whether it is generalisable. **Expected result: Model 3 performs better** This is mainly because we are expecting Yuxiang's brain waves to be more similar to his own brain waves compared to Luqman's. It should be easier to identify what brain waves determine stimulation. Luqman's dataset may have different patterns for stimulating and non-stimulating (refer to sample differences explored in EDA section 5.1.4). **Actual result: Model 6 performs better (Does not match expected)** Model 6 performs better (All POW waves, 78.2%) than Model 3 (PM, 46%). Possible reasons: ● Model that is trained solely on Yu Xiang's data primarily is actually generalisable, which means that the model is able to do well on datasets belonging to other people ● Yu Xiang 6 song dataset has existing outliers/noise present, which resulted in a lower test accuracy
Comparison 4 Model 4 & 6	**Main difference:** Amount of data used for training before testing on another individual Model 4: Yu Xiang's 36 songs tested on 6 Luqman songs Model 6: Yu Xiang's 30 songs tested on 6 Luqman songs **Expected result: Model 4 performs better** Since Model 4 is trained on more data, Model 4 would have seen more instances where brain waves are stimulating and non-stimulating, so it should be better able to differentiate between the two. **Actual result: Model 6 performs better (Does not match expected)** Model 6 (All POW waves, 78.2%), Model 3 (Beta waves, 77.3%) 19.1% increase in F1-score in the baseline models with lesser Yu Xiang data trained. Possible reasons: ● Overfitted on Yu Xiang's data, unable to generalise as well to Luqman's dataset ● Yu Xiang's 6 song dataset may have existing outliers/noise present, which resulted in a lower f1 score. ● Model 6 and 4 do not differ much for PM and Beta waves, with only up to a 0.2% difference in f1 score. The score only drops significantly for all POW waves. There could be subtle differences in alpha, gamma and theta brainwaves unique to this dataset, which resulted in the F1-score to be lower as result.

Section 7.2 Top performing classifiers and attribute sets

We analysed which of the classifiers and attribute sets are the most effective in helping the model make important predictions.

	F1 Scores of the Top 3 Classifier for each Models and Brain Waves		
Model	PM	POW Beta Waves	All POW waves
1	ExtraTreeClassifier - 91.6% RandomForest - 88.2% ExtGradientBoost - 83.9%	ExtraTreeClassifier - 85.9% KNN - 85.3% RandomForest - 79.3%	ExtraTreeClassifier - 73.2% KNN - 69.3% RandomForest - 69.0%
2	ExtraTreeClassifier - 95.8% RandomForest - 94.2% ExtGradientBoost - 93.3%	ExtraTreeClassifier - 94.3% RandomForest - 89.8% KNN - 87.6%	ExtraTreeClassifier - 92.6% RandomForest - 89.3% ExtGradientBoost - 85.8%
3	GaussianNB - 46% KNN - 40.7% DecisionTree - 35%	DecisionTree - 37% KNN - 35.5% ExtGradientBoost - 29.9%	DecisionTree - 38.1% KNN - 35.5% ExtGradientBoost - 29.8%
4	GaussianNB - 64.4% DecisionTree - 47.4% KNN - 45.1%	LogReg - 77.32% SVM - 67.1% KNN - 58.2%	DecisionTree - 58.5% ExtGradientBoost - 50.6% KNN - 50.3%
5	DecisionTree - 64.7% KNN - 61.4% BaggingClass - 57.4%	RandomForest - 64.9% ExtraTrees - 64.8% SVM - 63.3%	GaussianNB - 76.0% SVM - 65.1 LogReg - 64.4%
6	GaussianNB - 64.3% DecisionTree - 49.2% KNN - 42.2%	LogReg - 77.1% SVM - 67.0% DecisionTree - 58.4%	GaussianNB - 78.2% LogReg - 76.1% KNN - 51.7%

Figure 7.1: Top 3 Classifiers for each model.

Observations:

- Train-Test-Split Models (Typically 70-90% f1-score) performed better than Manual Train-Test-Split Models (Typically 60-70% f1-score, apart from Model 3 with 30-40%)
- For Train-Test-Split Models:
    - Best classifier: Extra Tree Classifier
    - Commonly top performing classifiers: Random Forest, Extra Gradient Boosting, KNN
    - Best attribute set: Performance Metrics (mostly > 90% f1-score)
- For Manual Train-Test-Split Models:
    - Best classifier: Inconclusive
    - Commonly top performing classifiers: GaussianNB, KNN, DecisionTree
    - Best attribute set: All POW Waves (not all, but majority of Models)

Section 7.3 Summary of Results

Overall, the model with the best results was Model 2 on Yu Xiang's 6 songs and Luqman's 6 songs.

- Validation technique: Train-Test-Split
- Best Classifier: Extra Trees Classifier
- Best performing attribute set: PM
- F1-score: 95.8%

However, for our goal of predicting whether new songs are stimulating or not, the train-test-split validation is likely to be unsuitable. In the use case of our model, the model will not be trained on any rows from the test song, thus the ability of our model can only be properly gauged through a manual train-test, aggregated by song.

The model with the best results with the Manual-Train-Test was Model 6 on Yu Xiang's 30 songs tested on Luqman's 6 songs.

- Validation technique: Manual-Train-Test (aggregated by song)
- Best Classifier: GaussianNB
- Best performing attribute set: All POW waves
- F1-score: 78.2%

Section 8.0 Conclusion and future works

Overall, the top performing models were done using the Train-Test-Split, achieving an f1-score of more than 90%. However, the Manual-Train-Test, which is a more accurate indicator of performance due to song aggregation, only achieved f1-scores of up to 70%. For Manual-Train-Test, there was no single model or attribute set that performed the best consistently. However, there were several more consistently top performing classifiers GaussianNB, KNN and DecisionTree, which can be explored further. Since Beta waves and all waves were the better performing attribute set, those atttributes should also be the focus of further research.

Apart from the performance of our Models, we derived possible insights:

(1) Aggregation hypothesis: Model cannot predict accurately when it has not been trained on any rows of the test song. This could be because stimulating and non-stimulating values may differ from song to song
(2) Generalisability: Models can actually be generalised from one person to another

For (1), we believe the comparison between Models 1 and 3 in section 7.1 may support this. For (2), this is somewhat contrary to our EDA in section 5.1.4, though the results from Models 3 and 6 seem to support it.

However, our results may have been affected by certain limitations:

1. Every song is not recorded in isolation, there are external variables such as possible noise and distraction in the testing environment which could have potentially triggered a change in brainwaves. This could result in instability of our machine learning model.
2. Inconsistency in EEQ quality of the Emotiv 3.0 headset hindered data collection process, which affected the amount of quality of data collected. This could have potentially resulted in our models to have a poor accuracy and f1-scores
3. Although we had implemented a buffer time of 20 seconds between each song, this timing might not be long, as the recency effect of the previous song could still affect the state of the individual. Hence, results for one song could possibly be affected by another.

Our insights cannot be entirely confirmed with our results, thus, we hope to propose further areas for research:

1. Trying on more combinations of attributes (combinations of sensors, types of waves)
    - With the limited time we have, we only performed baseline modelling on several combinations of the dataset. As such, we could potentially expand testing on other brainwaves by testing each of the brainwaves of Alpha, Gamma and Theta respectively though those waves were not emphasised as much as compared to Beta waves
2. Feature selection (from EDA, some sensors/attributes may not necessarily be useful in the predictions and may only add noise to the data)
    - Going forward, we could also possibly employ other feature selection techniques as well in order to select significant sensors and eliminate certain sensors that are less significant. This could potentially increase the stability and accuracy of our models, as well as to reduce the computational power which the models would take to run
3. Hyperparameter tuning for models
    - Hypertuning can also be employed to our best models, and this could potentially be used to potentially boost the accuracy of the model. Libraries to consider would be RandomSearch and GridSearch
4. Aggregation hypothesis: check on the values of stimulation, non-stimulation within different songs for the same person
    - Different songs could be tested on the same person and we can potentially compare the level of stimulation from one song to another, and the difference in values between the PM/Beta/All Other waves

Section 9.0 References

1. *http://www.operativeneurosurgery.com/doku.php?id=beta_wave#:~:text=Beta%20waves%20can%20be%20split,associated%20with%20normal%20waking%20consciousness.*

2. *https://www.sciencedirect.com/topics/medicine-and-dentistry/beta-wave#:~:text=Low%20beta%20waves%20(12%E2%80%9315,energy%2C%20anxiety%2C%20and%20performance.*

3. *https://www.mindbodygreen.com/articles/theta-waves*

4. *https://towardsdatascience.com/advanced-ensemble-learning-techniques-bf755e38cbfb#:~:text=Ensembling%20is%20less%20interpretable%2C%20the,accuracy%20than%20an%20individual%20model*

5. *https://practicalneurology.com/articles/2017-june/music-and-dementia-an-overview/pdf*

6. *https://machinelearningmastery.com/why-use-ensemble-learning/*

7. *https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6#:~:text=In%20order%20to%20avoid%20this,the%20last%20subset%20for%20test.*