
King Saud University

College of Computer and Information Science

Information Technology

# Bangalore EEG Epilepsy

## Section:56547



| Student | ID |
|---|---|
| Dana Sultan Alotaibi | 445203123 |
| Jana Saleh Obaid | 445200286 |
| Juri Thamer Alanazi | 445200260 |

## 1. Problem Definition

In this project, we worked with an EEG dataset containing multichannel brain signals, and our goal was to prepare this data for classification and clustering. EEG data is naturally noisy, contains sudden spikes, and varies widely across channels, so proper preprocessing is essential before applying machine learning techniques.

The main problem we addressed in this phase was cleaning, understanding, and preparing the EEG data so that it can be used in later stages for Decision Tree classification and K-Means clustering.
To achieve this, we analyzed the dataset, explored its structure, and applied several preprocessing techniques based on the issues we observed.

---

## 2. Data Mining Tasks

### 2.1 Classification Task

We prepared the dataset for a multi-class classification problem. The label column contains four classes (0, 1, 2, 3), and later in the project, we trained Decision Tree models using both Gini and Entropy criteria.
Our preprocessing work in this phase ensures that the classification models will have clean, normalized, and consistent inputs.

### 2.2 Clustering Task

We also prepared the dataset for unsupervised clustering using K-Means. Since clustering does not rely on labels, we focused on producing a high-quality, noise-reduced, and well-scaled feature set.
This preparation helps improve silhouette scores and cluster compactness later in the project.

---

## 3. Data Description

Before preprocessing, we examined the dataset to understand its structure and characteristics.

## 3.1 Overview of the Dataset

-The dataset contains 8000 EEG samples.

-Each sample includes 16 EEG channels (Channel1–Channel16).

-There is one label column with four classes.

-All columns are numeric, and we found no missing values.

## 3.2 Sample of Raw Data

We viewed the first rows of the dataset to get a sense of the raw values. The channels ranged between approximately −300 and +300, and the data appeared unscaled and noisy.

```
# Display first 5 rows
EEG.head()
```

| | Channel1 | Channel2 | Channel3 | Channel4 | Channel5 | Channel6 | Channel7 | Channel8 | Channel9 | Channel10 | Channel11 | Channel12 | Channel13 | Channel14 | Channel15 | Channel16 | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 7 | 18 | 25 | 28 | 27 | 20 | 10 | -10 | -18 | -20 | -16 | 13 | 32 | 12 | 10 | 0 |
| 1 | 87 | 114 | 120 | 106 | 76 | 54 | 28 | 5 | -19 | -49 | -85 | -102 | -100 | -89 | -61 | -21 | 0 |
| 2 | -131 | -133 | -140 | -131 | -123 | -108 | -58 | -51 | -70 | -77 | -76 | -76 | -73 | -57 | -40 | -14 | 0 |
| 3 | 68 | 104 | 73 | 34 | -12 | -26 | -38 | -36 | -67 | -88 | -25 | 31 | 18 | -4 | 6 | -29 | 0 |
| 4 | -67 | -90 | -97 | -94 | -86 | -71 | -43 | -11 | 23 | 46 | 58 | 50 | 39 | 19 | -9 | -41 | 0 |

---

## 3.3 Class Distribution

We checked the distribution of the classes and found that:

-Each class (0, 1, 2, 3) has exactly 2000 samples.

-This means the dataset is perfectly balanced, and we did not need to apply any oversampling or undersampling techniques.

---

## 3.4 Statistical Summary

We examined descriptive statistics and histograms for all 16 channels. We found that:

-Most features were centered around zero.

-Several features had long tails and extreme values.

-The wide range and presence of spikes confirmed the need for normalization and noise removal.

---

4. Data Preprocessing

After analyzing the dataset, we applied several preprocessing techniques to improve the quality of the data.

4.1 Missing Value Check

We started by checking for missing values in all columns.
Based on our analysis:

There were 0 missing values across all columns.
Because of that, we did not need to perform any imputation.

```
Missing values per column:
 Channel1     0
Channel2     0
Channel3     0
Channel4     0
Channel5     0
Channel6     0
Channel7     0
Channel8     0
Channel9     0
Channel10    0
Channel11    0
Channel12    0
Channel13    0
Channel14    0
Channel15    0
Channel16    0
Label        0
dtype: int64

 No missing values detected.
```

---

4.2 Converting Channels to Numeric

Even though the dataset appeared numeric, we ensured type consistency by converting all channel columns to numeric values using pd.to_numeric().
This step prevents errors during later calculations and preprocessing.

4.3 Outlier Detection (IQR Method)

We detected outliers using the IQR (Interquartile Range) method.
For each channel:

We calculated Q1, Q3, and IQR.

Any value outside [Q1 – 1.5×IQR, Q3 + 1.5×IQR] was considered an outlier.

Based on this process:

36.162% of rows contained at least one outlier.

Channels such as Channel11, Channel9, and Channel16 showed the highest outlier counts.

This confirmed that a noise-reduction step was necessary.

```
Rows: 8,000
Rows with any outlier: 2,893  (36.162%)
```

| | Outliers | Percent% |
|---|---|---|
| Channel11 | 1680 | 21.000 |
| Channel9 | 1613 | 20.162 |
| Channel16 | 1574 | 19.675 |
| Channel3 | 1564 | 19.550 |
| Channel1 | 1550 | 19.375 |
| Channel8 | 1519 | 18.987 |
| Channel7 | 1444 | 18.050 |
| Channel15 | 1406 | 17.575 |
| Channel2 | 1367 | 17.088 |
| Channel12 | 1336 | 16.700 |
| Channel10 | 1329 | 16.612 |
| Channel4 | 1319 | 16.488 |
| Channel14 | 1305 | 16.312 |
| Channel13 | 1272 | 15.900 |
| Channel6 | 1266 | 15.825 |
| Channel5 | 1241 | 15.513 |

4.4 Noise Removal (Median Binning)

To reduce noise without removing any rows, we applied median binning smoothing:

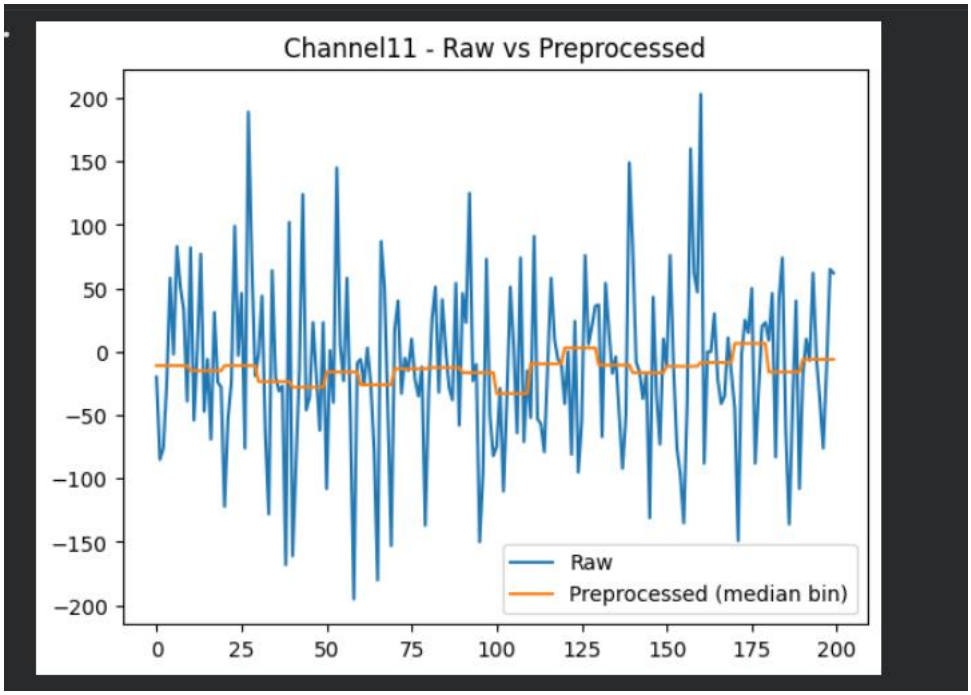
We divided the data into bins of size 10.

We replaced each bin’s values with the median of that bin.

After smoothing:

The percentage of rows containing outliers dropped from 36.162% to 32.375%.

This showed that binning successfully reduced noise and made the signal more stable.



4.5 Normalization (Min-Max Scaling):

Because the channels had a wide range (−300 to +300), we applied Min-Max normalization to scale all channel values to the range [0, 1].

We normalized all 16 channels using:

[X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}

After normalization, all features were on an equal scale, which is essential for both classification and clustering.

First 5 rows after preprocessing:

| | Channel1 | Channel2 | Channel3 | Channel4 | Channel5 | Channel6 | Channel7 | Channel8 | Channel9 | Channel10 | Channel11 | Channel12 | Channel13 | Channel14 | Channel15 | Channel16 | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.534709 | 0.507752 | 0.553753 | 0.560636 | 0.569201 | 0.582375 | 0.597586 | 0.508475 | 0.491228 | 0.513562 | 0.475836 | 0.492360 | 0.515411 | 0.554217 | 0.583624 | 0.557070 | 0 |
| 1 | 0.690432 | 0.715116 | 0.760649 | 0.721670 | 0.662768 | 0.634100 | 0.613682 | 0.499058 | 0.475439 | 0.457505 | 0.355019 | 0.346350 | 0.321918 | 0.345955 | 0.456446 | 0.504259 | 0 |
| 2 | 0.281426 | 0.236434 | 0.233266 | 0.250497 | 0.274854 | 0.323755 | 0.440644 | 0.393597 | 0.385965 | 0.406872 | 0.371747 | 0.390492 | 0.368151 | 0.401033 | 0.493031 | 0.516184 | 0 |
| 3 | 0.654784 | 0.695736 | 0.665314 | 0.578529 | 0.491228 | 0.480843 | 0.480885 | 0.421846 | 0.391228 | 0.386980 | 0.466543 | 0.572156 | 0.523973 | 0.492255 | 0.573171 | 0.490630 | 0 |
| 4 | 0.401501 | 0.319767 | 0.320487 | 0.324056 | 0.346979 | 0.394636 | 0.470825 | 0.468927 | 0.549123 | 0.629295 | 0.620818 | 0.604414 | 0.559932 | 0.531842 | 0.547038 | 0.470187 | 0 |

---

4.6 Feature Selection (Variance Threshold)

After normalization, we examined the variance of each channel.
We found that a few channels had very low variance, meaning they contributed very little information.

To address this, we applied Variance Threshold, using:

A threshold equal to 1.1 × minimum variance.

Based on this rule:

We removed 4 low-variance channels.

We kept 12 important and informative channels.

This step reduced dimensionality and improved the efficiency of later machine learning algorithms.

---

4.7 Final Preprocessed Dataset

By the end of preprocessing, we produced a clean and well-prepared dataset:

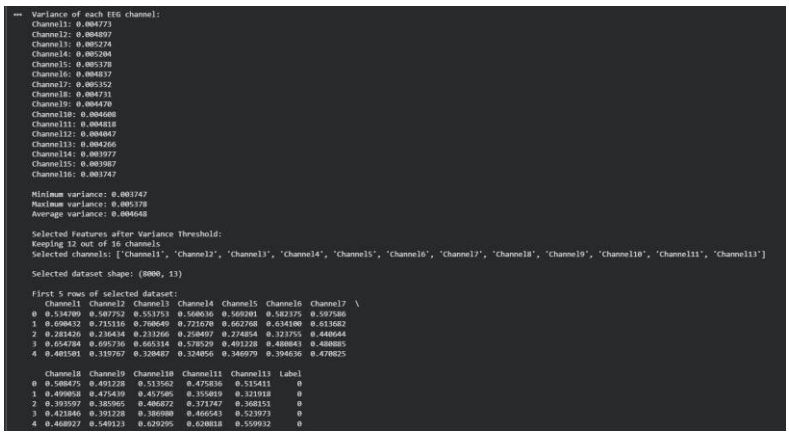
Rows: 8000

Selected features: 12 channels

Label column: included

File saved as: EEG_preprocessed.csv

This final dataset is now ready for classification and clustering in the next project phases.




## 5.1 Decision Tree Classification

### 5.1.1 Overview of Decision Trees:

A Decision Tree is a supervised classification model that predicts outcomes by splitting data based on feature values. In this project, the model was used to classify EEG signals, where each split reflects the value of an EEG channel. The most important features appear at the top of the tree because they reduce impurity the most.

Both the Gini and Entropy trees generated for our dataset consistently used Channel8, Channel11, and Channel7 as the first splitting features,

indicating these channels are the most informative for distinguishing EEG classes. To improve interpretability, the tree depth was limited to four levels, producing readable visualizations for both criteria gini and entropy.

---

5.1.2 Gini Impurity:

Gini Impurity measures how mixed the classes are inside a node. A value of 0.0 means the node is perfectly pure, while higher values indicate more class variation. The Decision Tree chooses splits that reduce Gini impurity the most, creating purer child nodes at each step.

```
DecisionTreeClassifier(max_depth=4, criterion="gini", random_state=123)
```

- Channel8 was chosen as the root split, meaning it provided the greatest reduction in impurity.

- Channel7 and Channel11 appeared repeatedly in upper levels, confirming they are highly informative EEG features.

- Many leaf nodes reached Gini = 0.0, showing perfectly pure class predictions.

- Limiting the tree to max_depth = 4 kept the structure readable while still capturing the key decision patterns.

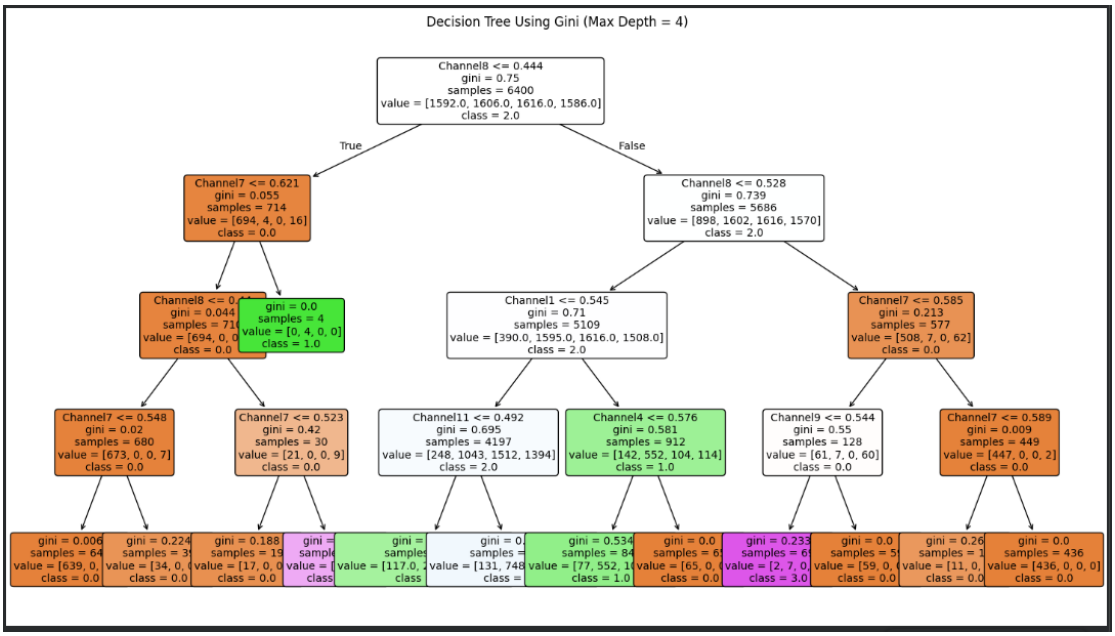
---

5.1.3 Entropy / Information Gain

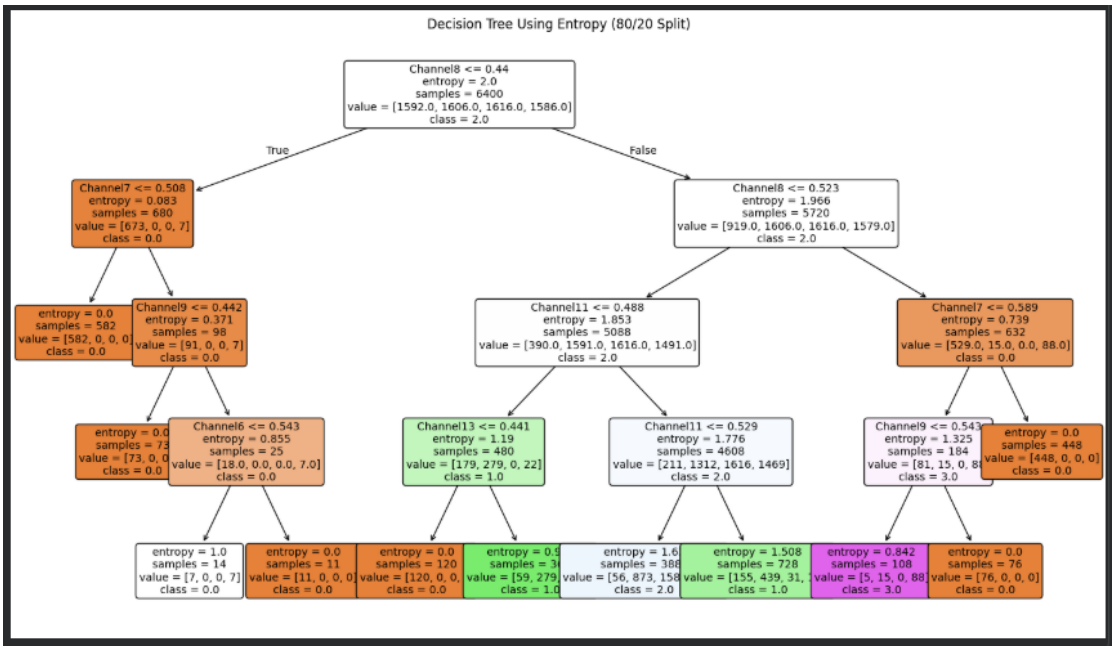
Entropy measures the disorder within a node, and the Decision Tree selects splits that give the highest Information Gain. In our entropy-based model (max_depth = 4), Channel8, Channel11, and Channel7 were the most influential features, appearing in the top splits and providing the greatest reduction in entropy. Several leaf nodes reached entropy = 0.0, indicating perfectly pure class predictions. The entropy model achieved strong performance across all splits, with its best accuracy (0.88125) at the 80/20 split.

5.1.4 Decision Tree Visualizations:

***Decision Tree (Gini) for split 80/20***




Decision Tree (Entropy) for split 80/20



## 5.2 K-Means Clustering:

**5**.2.1 Overview of the Technique :

K-Means partitions the dataset by minimizing the distance between each sample and the centroid of the cluster it belongs to. The algorithm works iteratively through the following steps:

Initialize k centroids randomly.

- Assign each sample to the nearest centroid based on Euclidean distance.

-Update each centroid based on the average of points assigned to it.

- Repeat the assignment–update cycle until the centroids no longer change.

This process leads to compact clusters that reflect underlying patterns within the numeric EEG features.

---

### 5.2.3 Python Packages and Methods Used:

The following tools were used to implement K-Means:

sklearn.cluster.KMeans

→ for building and fitting the model

*sklearn.metrics.silhouette_score*

→ for evaluating cluster quality

*matplotlib.pyplot*

→ for generating the cluster plots (shown later in Section 6)

*pandas / numpy*

→ for data handling and preprocessing

Also, Each model was run using *random_state = 42* to ensure consistent and reproducible clustering results.

---

### 5.2.4 Summary:

K-Means was used to explore whether the EEG dataset contains meaningful hidden groups. By testing multiple values of K and preparing the necessary outputs and evaluation metrics, this technique sets the foundation for the detailed clustering analysis presented in (Section 6:Evaluation and Comparison for Classification & Clustring).

---

## 6.1 Classification Evaluation

6.1.1: Accuracy Comparison Between Gini and Entropy Criteria:

Accuracy for gini:

| | Train/Test Split | Accuracy (Gini) |
|---|---|---|
| 0 | 90/10 | 0.875000 |
| 1 | 80/20 | 0.864375 |
| 2 | 70/30 | 0.849167 |

Accuracy for Entropy:

| | Train/Test Split | Accuracy (Entropy) |
|---|---|---|
| 0 | 90/10 | 0.873750 |
| 1 | 80/20 | 0.881250 |
| 2 | 70/30 | 0.857917 |

Combined Comparison:

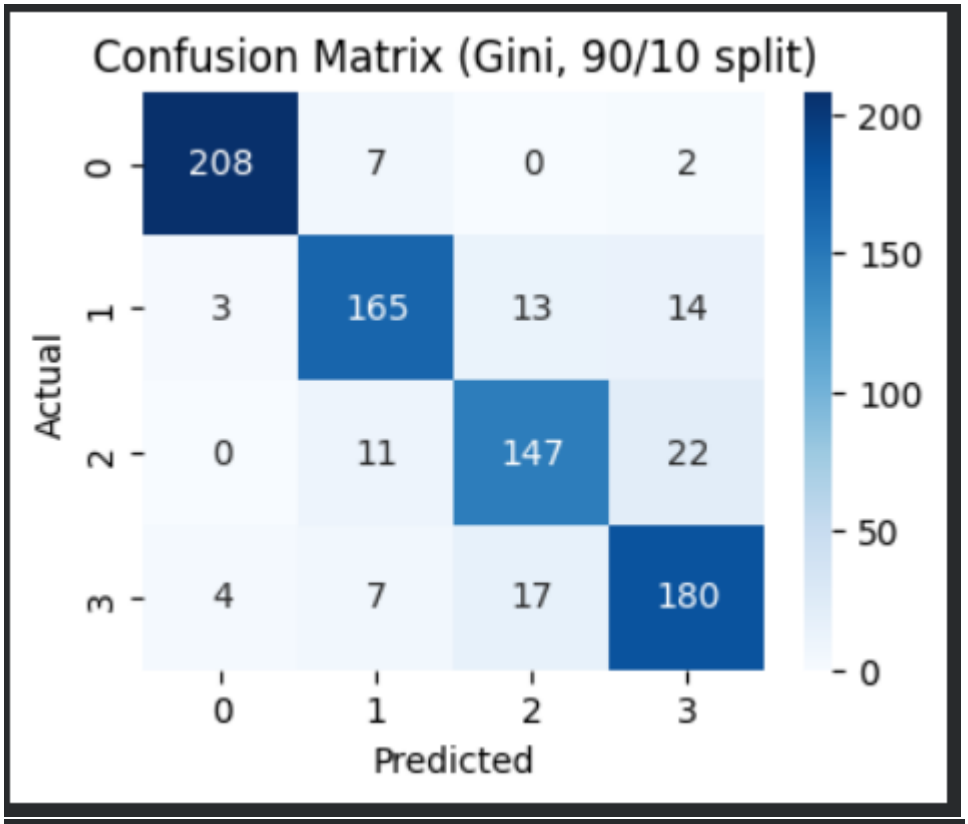
| | Train/Test Split | Accuracy (Gini) | Accuracy (Entropy) |
|---|---|---|---|
| 0 | 90/10 | 0.875000 | 0.873750 |
| 1 | 80/20 | 0.864375 | 0.881250 |
| 2 | 70/30 | 0.849167 | 0.857917 |

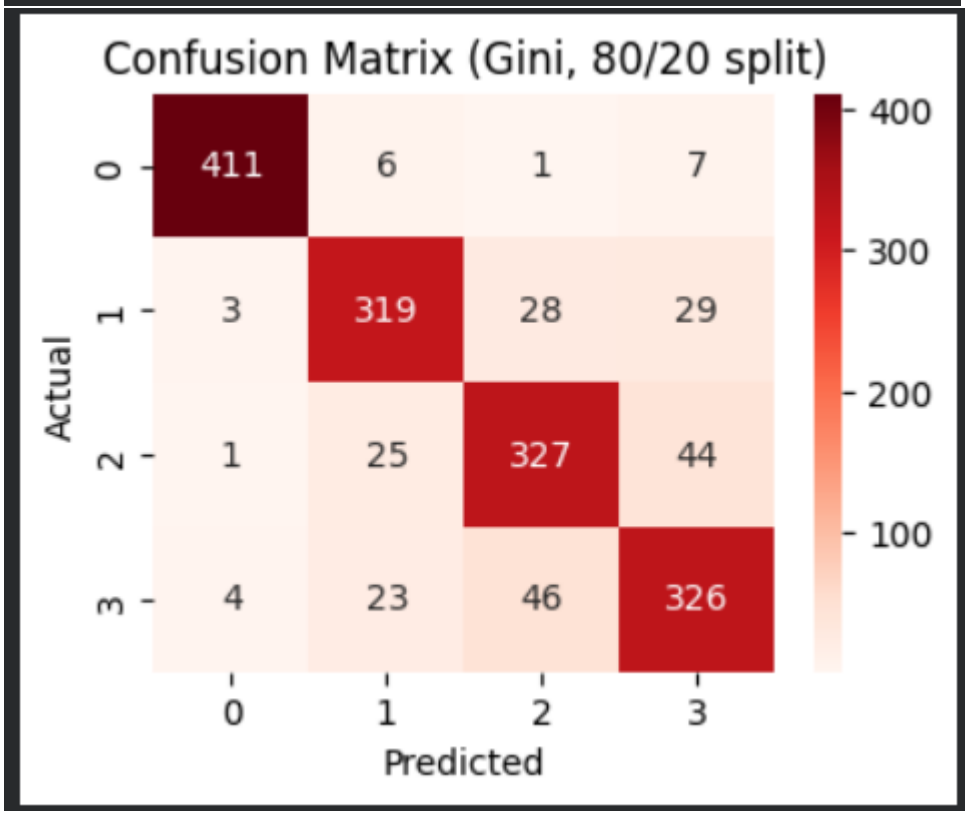
The accuracy results show that both Gini and Entropy perform well on the EEG dataset. Entropy achieved the highest accuracy (0.88125) using the 80/20 split, while Gini performed best on the 90/10 split. Overall, Entropy slightly outperformed Gini across the different data partitions.

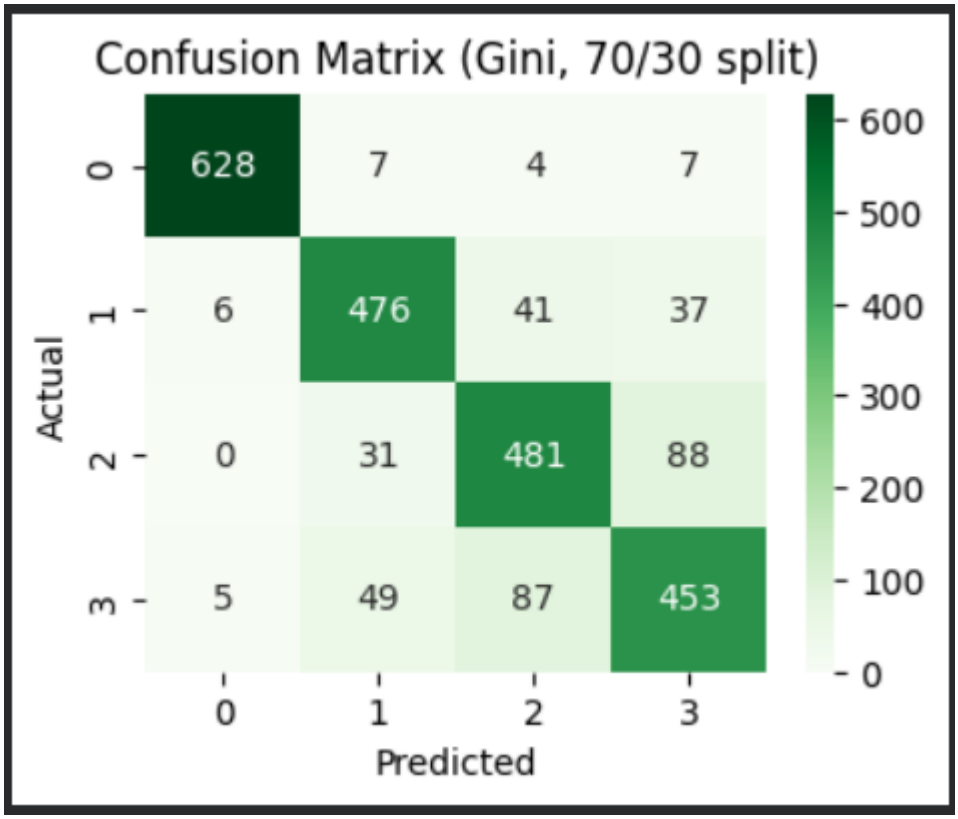
---

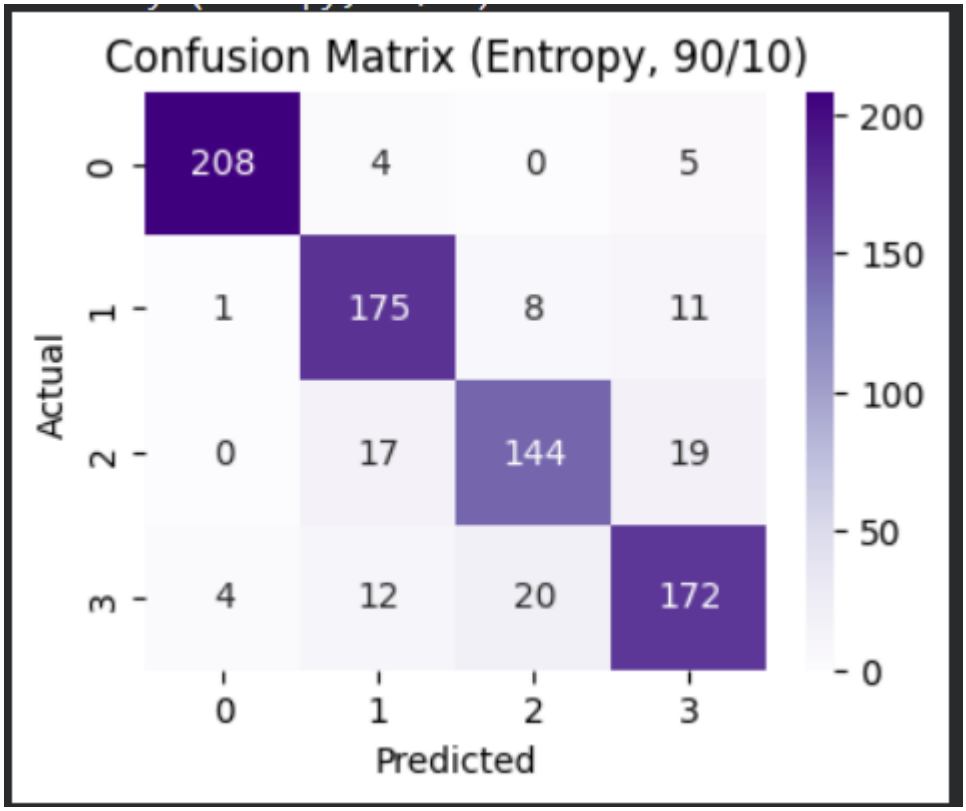
6.1.2 Confusion Matrices:

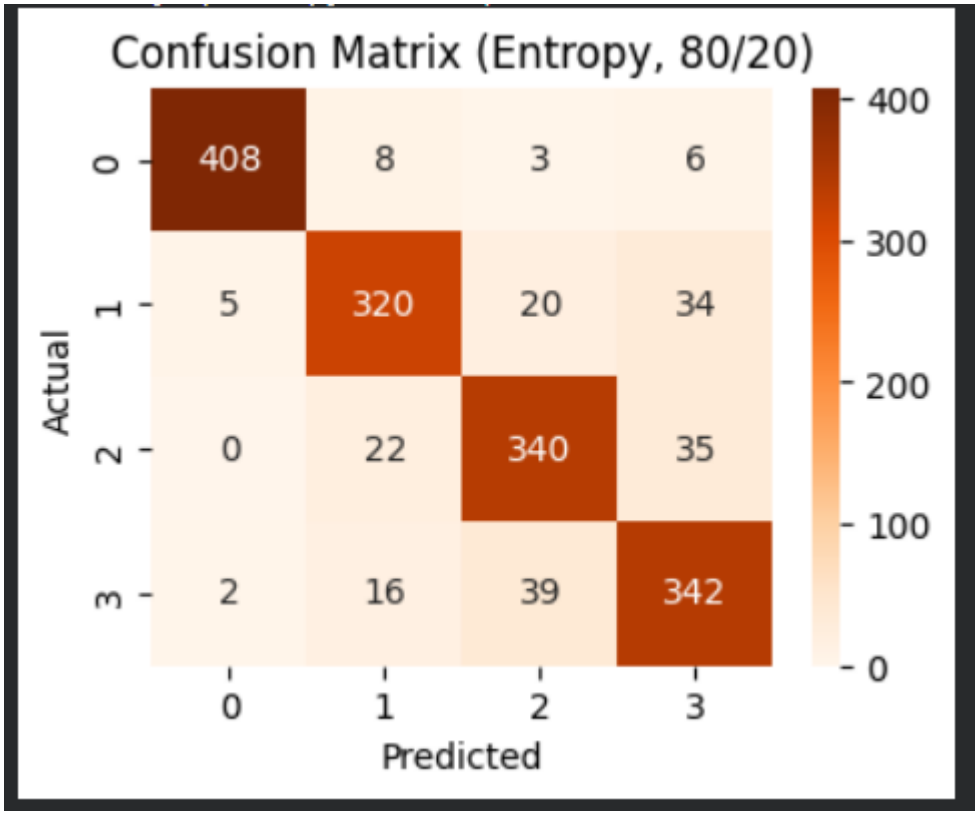
Each confusion matrix shows how well the model classified each EEG class.

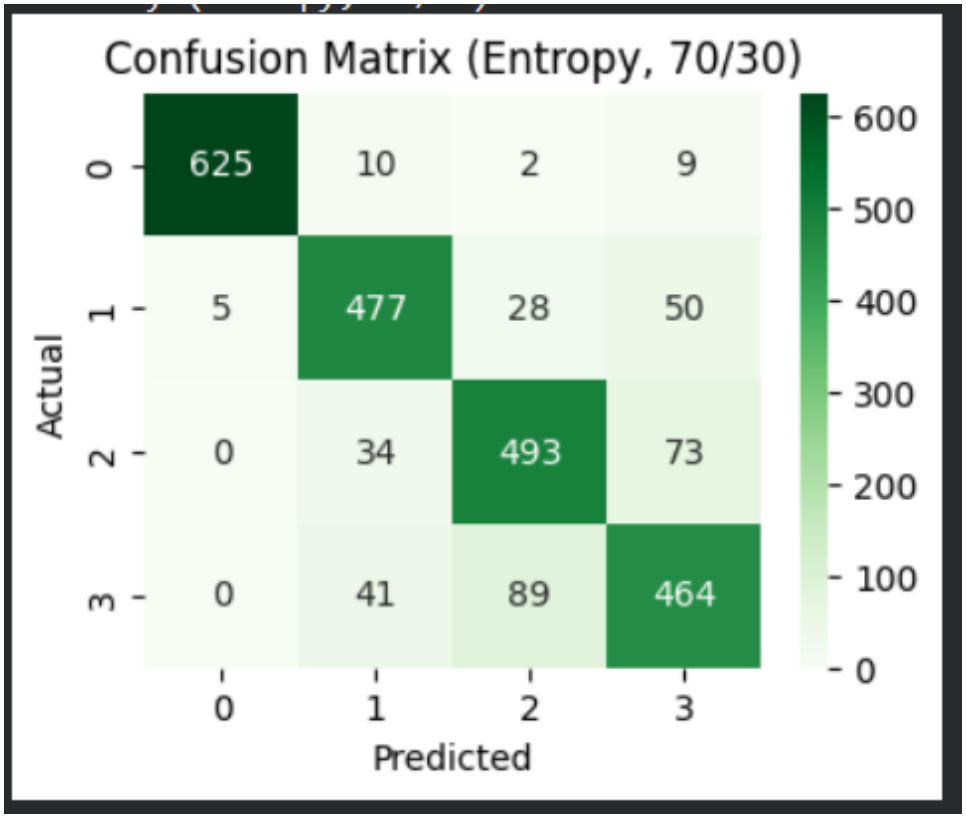

Confusion Matrix (Gini, 90/10 split)
208
7
0
2
3
165
13
14
0
11
147
22
4
7
17
180
Actual
Predicted
0
1
2
3
200
150
100
50
0


Confusion Matrix (Gini, 80/20 split)
411
6
1
7
3
319
28
29
1
25
327
44
4
23
46
326
Actual
Predicted
0
1
2
3
400
300
200
100


Confusion Matrix (Gini, 70/30 split)
Actual
Predicted
0
1
2
3
628
7
4
7
6
476
41
37
0
31
481
88
5
49
87
453
600
500
400
300
200
100
0


Confusion Matrix (Entropy, 90/10)
Actual
Predicted
0
1
2
3
208
4
0
5
1
175
8
11
0
17
144
19
4
12
20
172
200
150
100
50
0


Confusion Matrix (Entropy, 80/20)
Actual
Predicted
0
1
2
3
408
8
3
6
5
320
20
34
0
22
340
35
2
16
39
342
400
300
200
100
0


Confusion Matrix (Entropy, 70/30)
Actual
Predicted
0
1
2
3
625
10
2
9
5
477
28
50
0
34
493
73
0
41
89
464
600
500
400
300
200
100
0

6.1.3 Interpretation of Results:

Both Gini and Entropy Decision Tree models performed strongly on the EEG dataset, with accuracies above 0.84 for all splits. Entropy showed slightly better performance overall, achieving its highest accuracy (0.88125) on the 80/20 split, while Gini performed best on the 90/10 split (0.87500). This indicates that Information Gain was slightly more effective in selecting useful EEG features.

The confusion matrices show that most predictions lie on the diagonal, meaning the models classified many samples correctly—especially for Class 0 and Class 3. Most errors occurred between Class 1 and Class 2, likely due to overlapping EEG patterns between these classes.

Across the train/test splits, the 70/30 split produced the lowest accuracy for both criteria due to reduced training data, while 80/20 was the best for entropy and 90/10 for Gini. Overall, both methods captured the main EEG patterns well, with entropy showing a small advantage in generalization.

---

6.1.4 Feature Importance:

The Decision Tree models identified a few EEG channels as the most important for classification. In both the Gini and Entropy trees, Channel8, Channel11, and Channel7 appeared at the top levels of the tree, meaning they provided the highest reduction in impurity and entropy. These channels contributed most to separating the EEG classes. Other channels only appeared deeper in the tree, indicating lower importance. Overall, the results show that the classification mainly depends on a small set of highly informative EEG channels.

---

**6.2 Clustering Evaluation and Comparison**:

**6.2.1 Evaluation of Different K Value:**

To assess clustering performance, three values of K were tested: K = 3, K= 4, and K = 5.

For each K, the following metrics were computed:

- Average Silhouette Score.
- Total Within-Cluster Sum of Squares (WCSS).
- Cluster Visualization (colored scatter plot).

Testing multiple K values allows identifying the optimal cluster configuration based on both quantitative metrics and visual interpretation.

---

### 6.2.2 Interpretation of Results for Each K:

- When K = 3:

Silhouette Score: 0.6095

WCSS: 304.83

Visual interpretation:

The data is split into three broad diagonal segments. Clusters are moderately separated but still follow the strong linear pattern of the EEG data.

- When K = 4:

Silhouette Score: 0.6144

WCSS: 262.03

Visual interpretation:

Boundary adjustments create slightly tighter groupings. Separation improves marginally, and clusters follow the same linear orientation.

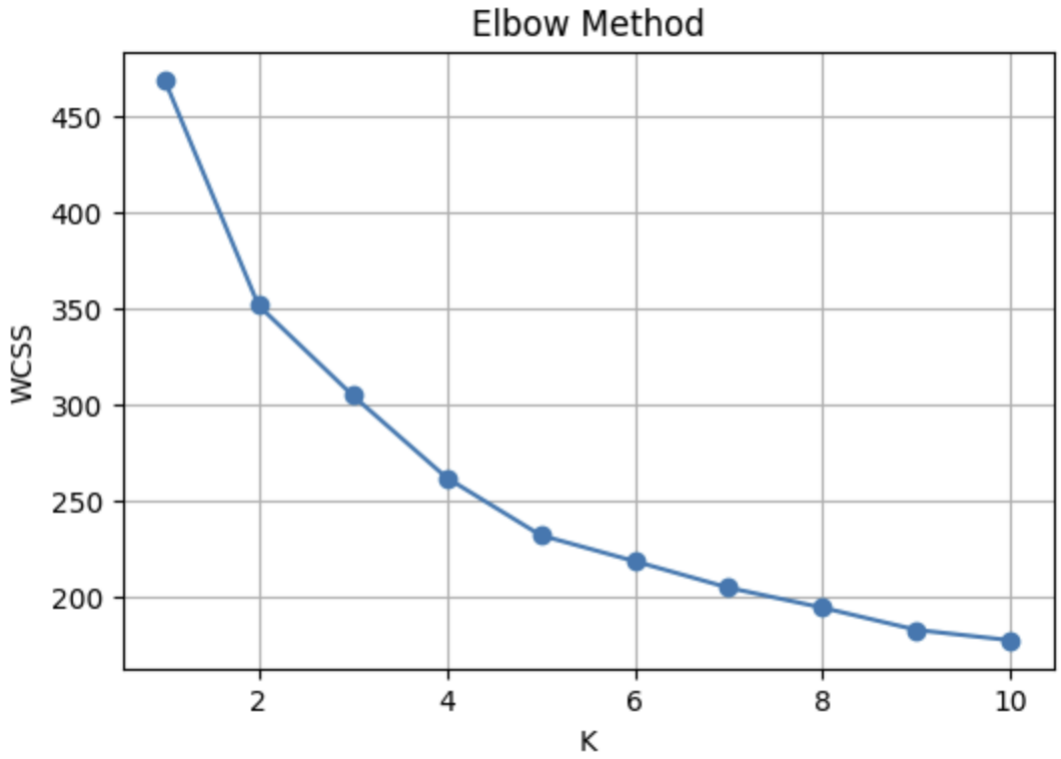
- When K = 5:

Silhouette Score: 0.6200

WCSS: 232.13

Visual interpretation:

Produces the most compact clusters. Group separation is more refined, and the clusters align well with the dataset's elongated distribution.




---

### 6.2.3 Summary Table of Silhouette & WCSS:

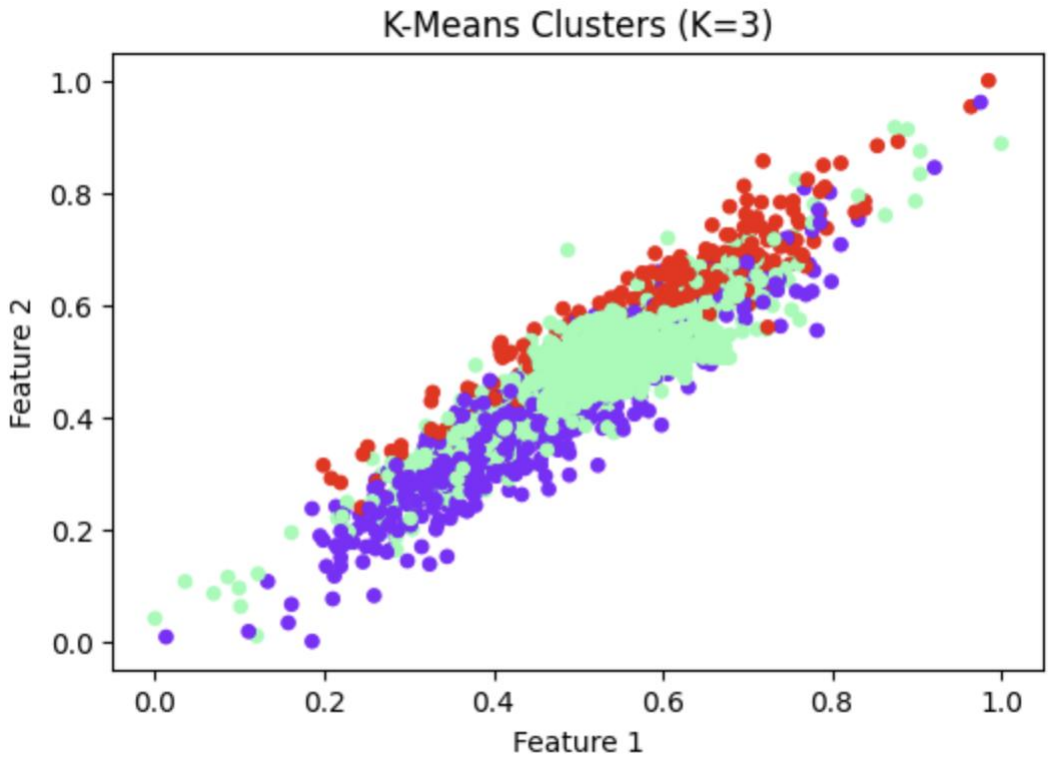
| | K=3 | K=4 | K=5 |
|---|---|---|---|
| Average Silhouette width | 0.60958 | 0.61440 | 0.62000 |
| total within-cluster sum of square | 304.83160 | 262.03107 | 232.13587 |

---

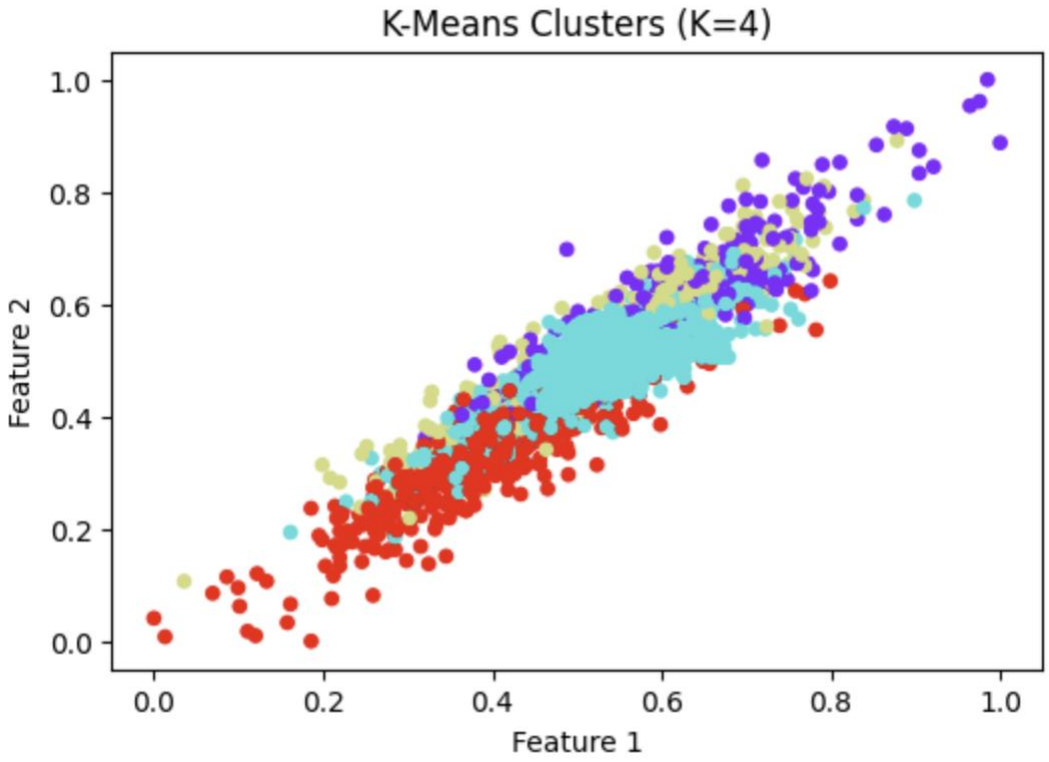
### 6.2.4 Cluster Visualizations:

The following scatter plots were generated to visualize the separation of clusters for each K:

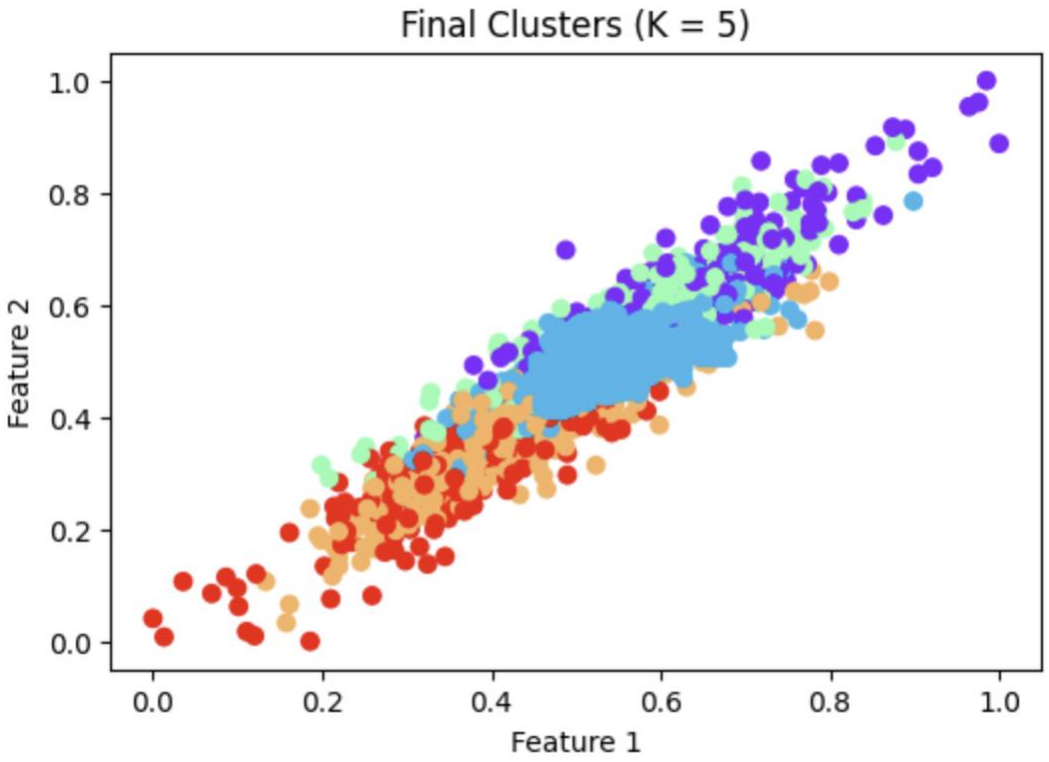
- ***Figure 1:*** K-Means Clusters (K = 3)



- ***Figure 2:*** K-Means Clusters (K = 4)



- ***Figure 3:*** K-Means Clusters (K = 5)



### 7.1.1 Findings and Discussion:

7.1.1 Summary of what we ran:

We ran two families of experiments on the BEED dataset:

- **Supervised classification:**

using Decision Trees with two impurity criteria: Gini and Entropy. For each criterion we trained and evaluated models using three train/test splits: 90/10, 80/20, 70/30. We recorded accuracy and confusion matrices for each configuration and selected the best Entropy model for visualization and interpretation.

**- Unsupervised clustering:**

using K-Means with $K \in \{3, 4, 5\}$. For each K we recorded Silhouette Score and WCSS (Within-Cluster Sum of Squares) and plotted the cluster assignments on a 2D projection (first two selected features) to interpret cluster structure.

All preprocessing steps (IQR-based outlier detection, median-binning smoothing, Min–Max normalization, and variance-threshold feature selection) were applied before modeling. We used the final set of 12 selected EEG channels as model inputs.

### 7.1.2 Classification results:

**- numeric summary and comparison:**

**Accuracy summary  from our analysises:**

**Gini:**

90/10 → 0.8750

80/20 → 0.8644

70/30 → 0.8492

**Entropy:**

90/10 → 0.8738

80/20 → 0.8813 ← best overall

70/30 → 0.8579

---

**Interpretation**

-Both impurity criteria give high and comparable performance (≈84.9%–88.1% accuracy), indicating Decision Trees are well suited to the preprocessed BEED signals.

- Entropy achieved the single highest accuracy (≈**88.13%** at the 80/20 split). However, Gini gave better stability at the largest training set (90/10)

. This behavior is consistent with the fact that small differences between these split criteria often depend on sample size and random variation.

- The roughly 2–4% difference between best and worst configurations suggests the model generalizes reasonably well and that preprocessing produced useful features.

**Classification Findings:**

- Confusion matrices show the model learns meaningful patterns but often confuses **Class 2 ↔ Class 3** and **Class 1 ↔ Class 2**, indicating overlapping EEG characteristics.

- Best supervised model: **Decision Tree (Entropy, 80/20 split)** with **≈88% accuracy**.

- Most informative channels appear repeatedly in the top tree splits and feature importances.

- Classification provides a practical solution for **automatic EEG window labeling**.

---

**clustering Findings:**

- Silhouette scores: 0.61–0.62 for K = 3–5 → indicates well-separated, coherent clusters.

- WCSS decreases predictably; elbow suggests K = 4–5 as reasonable choices.

- Clusters reveal distinct EEG signal regimes, not necessarily matching ground-truth labels.

- Useful for finding subtypes, outliers, and structure in the data.

---

**Comparison of Classification vs. Clustering:**

- classification: supervised, answers “what class is this sample?”; useful for automated decision-making.

- Clustering: unsupervised, reveals hidden structure and patterns; useful for exploratory analysis.

- Together they provide a strong understanding of both label accuracy and natural data grouping.

**Visuals to Include:**

1- Confusion matrices (Gini & Entropy)

2- Accuracy comparison (all splits)

3- Final decision tree figure

3- Feature importance bar chart

4- Cluster scatter plots (K = 3–5)

5-Silhouette & elbow charts
Each figure should include 1–2 sentence interpretations.

---

**Limitations:**

-Trees may overfit

---

**Outcome:**

- **Best model:** Decision Tree (Entropy), ~88% accuracy.
- **Best clustering:** K-Means (K=4–5), silhouette 0.61–0.62.
- **Overall:** We deliver a reliable classifier for automatic labeling and a clustering analysis that reveals deeper EEG patterns for clinical exploration.