Development of a Depth of Anaesthesia (DoA) Index Using Unsupervised Machine Learning

Report 2

JANITH WEERAKKODY

U1161698

Abstract

Determination of a Depth of Anaesthesia (DOA) index is important for enhancing surgical safety by observing patients' Anaesthetic levels in real time. This study is focused on developing a new DOA index by utilizing unsupervised learning techniques, specifically K-means clustering and Hierarchical clustering. The general objective of this work is to design a robust model for DoA classification, which would be applicable for monitoring and maintaining optimum anesthetic administration to reduce side effects during surgical procedures. This study leverages datasets provided by Well Hospital, which are comprised of several feature sets derived from EEG signals. The results of the clustering methods have been compared with known labels. And the two clustering methods have been evaluated using evaluation metrics such as Silhouette Score and Davies-Bouldin Index, according to which k-means clustering is more efficient and scalable than hierarchical clustering when determining the DOA level.

Table of Contents

1 Introduction

The Depth of Anaesthesia (DOA) monitoring is a very significant aspect of the surgical procedures in which accurate control of anaesthetic agents is considered crucial for the success of the operation and safety of the patient. The conventional techniques include the Bispectral (BIS) Index, in which a numerical range is provided for assessment of depth of anaesthesia. However, these indices are limited by the use of preset models, which cannot consider the individual variability in patient responses to anaesthesia. More advanced machine learning, especially unsupervised learning, may provide a useful approach for improved and more personalized DOA assessment, given the complexity and variability within EEG signals.

In this report, unsupervised learning methods such as K-means clustering and Hierarchical clustering have been employed to formulate a new DOA index. In this study, we perform clustering corresponding to different anaesthetic states using raw EEG data and a set of derived features (p1 to p9). This work is part of the effort to enhance patient safety in surgery by proposing a machine learning based approach to DOA monitoring.

2 Literature Review

In the last years, machine learning, especially unsupervised learning approaches to medical data analysis, has increased significantly. Unsupervised learning is frequently used when the labelled data are either scarce or have a high cost. This includes the development of health indices, such as the DOA index, which needs accurate and dynamic classification of patient states.

Clustering techniques in unsupervised learning, such as K-means clustering and Hierarchical clustering, have been used with much success in monitoring different health conditions by analysing EEG data. For example, K-means clustering segmented the patterns in EEGs based on similarities in the processing of the EEG signal. This provides a possible way of distinguishing between different stages of consciousness or anaesthesia (Li et al., 2019). Similarly, Hierarchical clustering often works with time-series data like EEG on the extraction of hidden patterns or groupings, thus proving to be quite promising as an approach in DOA classification according to Zhao et al. 2020. The advantage of unsupervised methods in finding latent structures without preliminary labelling is especially valuable in situations when it's difficult to define clear-cut classes for various stages of anaesthesia.

There is a lot of research available on the use of machine learning techniques to develop the DOA index. For example, unsupervised learning has been adopted for research by Doi et al. (2016) in classifying patients' EEG signals into different states of anesthesia, proving that these methods are more reliable than traditional BIS indices. Similarly, Hassan et al. (2018) successfully classified the EEG data into distinct groups, hence proving that clustering algorithms can be used to predict the awareness of a patient during surgery, improving monitoring systems in anesthesia.

These studies confirm, in fact, that unsupervised learning can be of great value in developing DOA indices, in particular due to raw EEG data, without requirements for labeled samples. The literature illustrates the application of unsupervised clustering algorithms as a promising method for extending the precision and reliability of anesthesia depth assessment, aiming at better meeting the demands related to patient safety in surgery.

3 Data Description

In this study two datasets were used, namely “Project Dataset 2” and “Project Dataset 3”. Project Data Set 2 includes 4965 segments, with each segment representing one patient's EEG data. Each segment consists of nine feature sets, p1 to p9, generated from raw EEG signals. The different features represent various aspects of the signals processed by using different extraction methods. The true labels of these segments, for the above-mentioned, are provided in the Project Data Set 3, highlighting the "Deep anaesthetic" or "Awake" state of the patient. The data will be divided into training and testing sets, for model development and evaluation, respectively.

4 Methodology

For the purpose of formulating a new index for DOA predictions, we are focusing on two approaches using two machine learning methods to determine the index. And then, the two models are evaluated using a few evaluation metrics to determine which index gives the best DOA prediction.

In the scope of this study the two machine learning methods used are K-means Clustering and Hierarchical Clustering. To evaluate the models, statistical metrics such as Silhouette Score and Davies-Bouldin Index have been employed. Also, the classification results are compared with actual labels given in the project dataset 3.

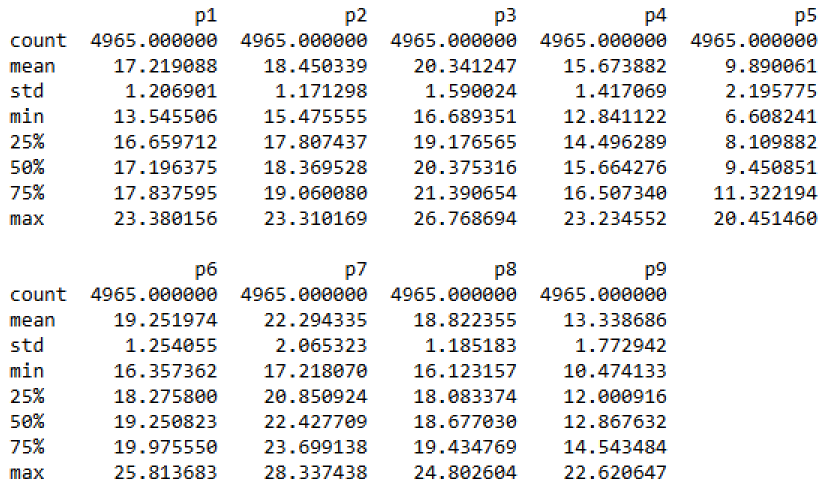
4.1 Data Preprocessing

The preprocessing steps include handling missing values, standardizing the feature sets to ensure all features have the same scale, and splitting the data into training and testing sets. Since there is no explicit split in the data, we treat the dataset as a whole and apply clustering without the need for a training/testing split. Standardization is done to scale the features and ensure no feature dominates the process of clustering. Each feature will be scaled to zero mean and unit variance. This standardization step is particularly important in K-means, since it is sensitive to the magnitude of the features.

4.2 Exploratory Data Analysis

4.2.1 Summary Statistics

The summary statistics of the nine features were observed to have an idea about the dataset and what should be done in the analysis. Since all the variables in the dataset were quantitative variables, the five number summary of the variables along with the metrics such as mean and standard deviation were observed.

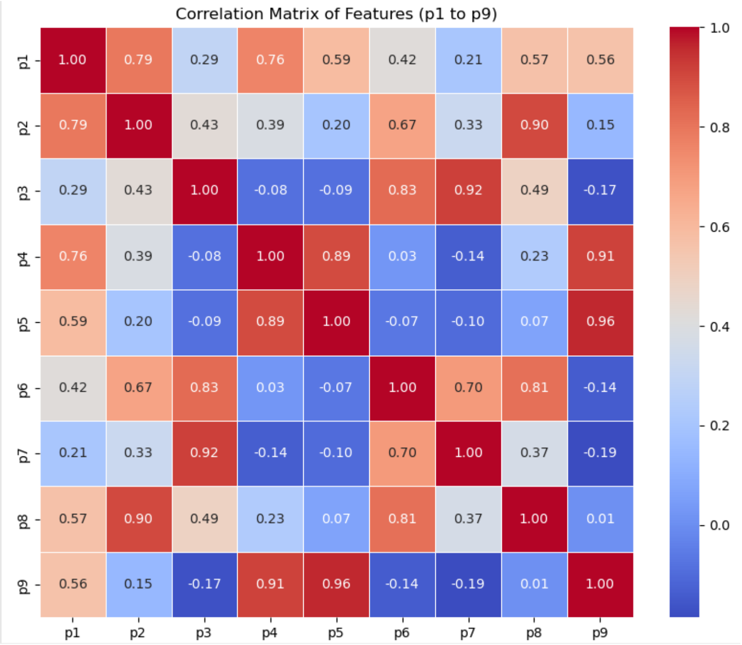


```
                p1           p2           p3           p4           p5
count  4965.000000  4965.000000  4965.000000  4965.000000  4965.000000
mean     17.219088    18.450339    20.341247    15.673882     9.890061
std       1.206901     1.171298     1.590024     1.417069     2.195775
min      13.545506    15.475555    16.689351    12.841122     6.608241
25%      16.659712    17.807437    19.176565    14.496289     8.109882
50%      17.196375    18.369528    20.375316    15.664276     9.450851
75%      17.837595    19.060080    21.390654    16.507340    11.322194
max      23.380156    23.310169    26.768694    23.234552    20.451460

                p6           p7           p8           p9
count  4965.000000  4965.000000  4965.000000  4965.000000
mean     19.251974    22.294335    18.822355    13.338686
std       1.254055     2.065323     1.185183     1.772942
min      16.357362    17.218070    16.123157    10.474133
25%      18.275800    20.850924    18.083374    12.000916
50%      19.250823    22.427709    18.677030    12.867632
75%      19.975550    23.699138    19.434769    14.543484
max      25.813683    28.337438    24.802604    22.620647
```

Figure 1 – Summary Statistics

4.2.2 Correlations Between Variables

A correlation matrix has been created to visualize the relationships between the variables of the dataset.



	p1	p2	p3	p4	p5	p6	p7	p8	p9
p1	1.00	0.79	0.29	0.76	0.59	0.42	0.21	0.57	0.56
p2	0.79	1.00	0.43	0.39	0.20	0.67	0.33	0.90	0.15
p3	0.29	0.43	1.00	-0.08	-0.09	0.83	0.92	0.49	-0.17
p4	0.76	0.39	-0.08	1.00	0.89	0.03	-0.14	0.23	0.91
p5	0.59	0.20	-0.09	0.89	1.00	-0.07	-0.10	0.07	0.96
p6	0.42	0.67	0.83	0.03	-0.07	1.00	0.70	0.81	-0.14
p7	0.21	0.33	0.92	-0.14	-0.10	0.70	1.00	0.37	-0.19
p8	0.57	0.90	0.49	0.23	0.07	0.81	0.37	1.00	0.01
p9	0.56	0.15	-0.17	0.91	0.96	-0.14	-0.19	0.01	1.00

Figure 2 – Correlation matrix

The above correlation matrix depicts a few significant correlations between the features. p2 and p8 are highly positively correlated with a value of 0.90, which gives a good indication that they represent similar information. Similarly, p4 and p5 are highly correlated with a value of 0.89 hence redundant. On the other hand, p7 shows weak correlations with most other features, which makes it a potential distinct variable. The insights derived from the above correlation matrix will be very useful in feature selection step, to determine which features should be retained for the clustering process.

4.3 Advanced Analysis

4.3.1 Model Formulation

K-Means Clustering

K-means Clustering is a typical unsupervised machine learning algorithm that segments data into a specified number of clusters. This algorithm works by iteratively assigning each data point to the closest cluster centroid and then recalculates centroids based on current assignments. The algorithm runs until the cluster assignments no longer change, or a specified number of iterations is reached. K-means is quite useful in problems, such as the Depth of Anaesthesia index development, where the aim will be to classify the EEG data into distinct groups based on underlying patterns of the data, for example, "Awake" and "Deep Anaesthetic". K-means clustering will allow us to group the EEG data into meaningful clusters, which can then be used to develop a DOA index. Another main reason behind the choice of k-means clustering in this study was its effectiveness and efficiency in handling large datasets such as the EEG features in Project Data Set 2.

One of the great benefits of using K-means clustering in this analysis is its computational efficiency. The K-means algorithm runs relatively fast and is suitable for use on large datasets, such as the 4965 EEG segments available in the Project Data Set 2. This technique is quite simple to run and provides a comprehensible outcome by dividing data into distinct clusters. K-means can, therefore, classify these EEG segments into meaningful categories representing different levels of anaesthesia in developing a DOA index. Since K-means is based on variance minimization within each cluster, well-separated states of anaesthesia can also be identified using it, which is critical for correct monitoring of DOA.

Despite these several advantages, there are certain shortcomings with K-means which are essential to consider for the present analysis. One important limitation is that K-means demands pre-identification of the number of clusters. While for this project, two clusters were assumed to be representative of "Awake" and "Deep anesthetic" states, this may not be a valid assumption in all datasets and real scenarios. Insufficient selection of the number of clusters may lead to suboptimal results. Besides, K-means implicitly assumes the clusters to be spherical and of similar sizes, which may not be the actual case in complex data such as EEG signals. This might result in less accurate clustering if the true structure of the data does not meet these assumptions. Second, k-means is sensitive to the initial placement of centroids, which sometimes leads to bad local optima if the centroids are not well-chosen.

Finally, k-means can be sensitive to outliers, since if not treated properly, it might skew the results.

Hierarchical Clustering

Hierarchical Clustering is another unsupervised method. The goal of Hierarchical Clustering is to build a hierarchy of clusters. Unlike in K-means, there is no need to specify the number of clusters in advance; instead, Hierarchical Clustering outputs a tree-like structure describing how each point and cluster is related at every possible level. Hierarchical clustering can be performed by different linkage criteria, such as Ward linkage, which minimizes variance within clusters, and single linkage, which minimizes the distance between the closest points of two clusters.
It is a useful approach in an attempt to develop a Depth of Anaesthesia index, and for pattern identification in EEG data, since it uncovers a hierarchical structure of different anaesthetic states. The resulting dendrogram gives the flexibility of choosing the appropriate number of clusters and hence allows insight into the natural structure of the data.

The main advantage of Hierarchical clustering in this analysis is that it may unveil hidden structure present within the data without knowing the number of clusters a priori. This will be especially helpful with complex data, such as EEG signals, where true number of clusters are not readily apparent. As it is possible to obtain a dendrogram from Hierarchical clustering, this often offers a visual overview of how clusters group at different levels of granularity, thus giving an intuitive idea of the relationships among data points. The other flexibility is that the assumption of spherical clusters does not bind Hierarchical clustering as much as techniques such as K-means. Such flexibility can capture clusters of various shapes and sizes, which is the case with the EEG data representing different levels of anaesthesia.

While Hierarchical clustering has many advantages in this index development task of DoA, there are some challenges and disadvantages associated with it. Its first disadvantage is that it is computationally inefficient. Such a scenario may finally increase time consumption and resources wastage. Secondly, the process is sensitive to noisy data and outliers; such situations distort the clustering outcome and lead to misleading interpretations. Unlike K-means, which is somewhat resistant to outliers, Hierarchical clustering tends to merge outliers into inappropriate clusters, which reduces the accuracy of the classification. Finally, while the dendrogram provides insights into the structure of the data, for very large datasets and when the number of clusters should be decided manually, it may be cumbersome, making the technique less automated compared with K-means.

4.3.2 Model Evaluation

Few evaluation metrics have been used in this project to evaluate the performance of the models formulated.

Silhouette Score

The Silhouette Score is a method developed to assess the quality of the clusters. It gives a value of measurement for every object about how similar it is to its own cluster when compared with other clusters. A score close to 1 means it is well-clustered; close to -1 means bad clustering. A higher Silhouette Score is indicative of well-separated clusters, meaning there would exist clear boundaries between different groups in the data.

Davies-Bouldin Index

The Davies-Bouldin Index is the metric that calculates the quality of clustering by taking as input the average similarity ratio for every cluster with the most similar cluster. In essence, the Davies-Bouldin Index is the measure of the ratio between within-cluster dispersion and between-clusters separation. In general, a low Davies-Bouldin Index corresponds to well-separated and compact clusters, reflecting good clustering. The Davies-Bouldin Index finds its application in verifying the efficiency of certain algorithms on unsupervised learning tasks

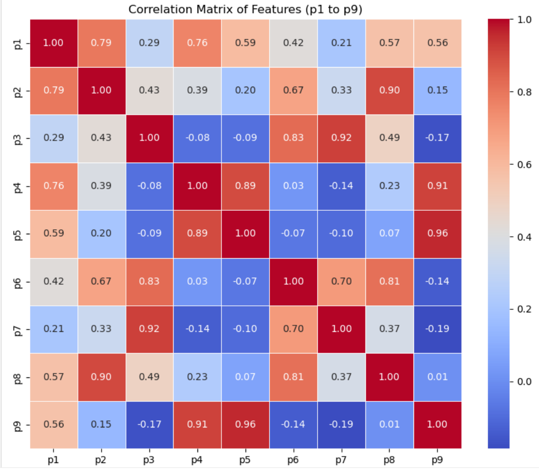
Jupyter Notebook

All the programming codes are written in Python language and for that a platform called Jupyter notebook has been used.

5 Results

5.1 Feature Selection

5.1.1 Random Forest



	p1	p2	p3	p4	p5	p6	p7	p8	p9
p1	1.00	0.79	0.29	0.76	0.59	0.42	0.21	0.57	0.56
p2	0.79	1.00	0.43	0.39	0.20	0.67	0.33	0.90	0.15
p3	0.29	0.43	1.00	-0.08	-0.09	0.83	0.92	0.49	-0.17
p4	0.76	0.39	-0.08	1.00	0.89	0.03	-0.14	0.23	0.91
p5	0.59	0.20	-0.09	0.89	1.00	-0.07	-0.10	0.07	0.96
p6	0.42	0.67	0.83	0.03	-0.07	1.00	0.70	0.81	-0.14
p7	0.21	0.33	0.92	-0.14	-0.10	0.70	1.00	0.37	-0.19
p8	0.57	0.90	0.49	0.23	0.07	0.81	0.37	1.00	0.01
p9	0.56	0.15	-0.17	0.91	0.96	-0.14	-0.19	0.01	1.00

Figure 3 – Correlation matrix

Feature selection is always a very critical step in any kind of clustering to reduce redundancy and improve the efficiency of the model. From the above correlation matrix, some highly correlated features include p2 versus p8 (0.90) and p4 versus p5 (0.89). Removing any one in each case would reduce information redundancy since they all describe roughly the same aspect; hence, the model would be simplified with no significant loss in data. This is because, in either K-means or Hierarchical clustering, such redundant features have been combined or excluded to avoid multicollinearity. This will enable the clustering algorithms to focus on the most informative and independent variables only. Hence, the features p2 and p4 are being removed from the feature set on which the analysis is carried out. Other features like p7, where the relationship from other features is weak, were also retained since they provide diverse information to the clustering. In this way, the clustering model performance improves with enhanced separation of clusters and lower computation complexity.

5.2 Model Performance

5.2.1 K-Means Clustering

Evaluation Metric	Value
Silhouette Score	0.2879
Davies-Bouldin Index	1.3666

Table 1 – Evaluation metrics for K-means clustering

Silhouette Score (0.2879)

The Silhouette Score gives the similarity of each data point to its own cluster in comparison with other clusters. The value 0.2879 does mean that the clusters are somewhat well-separated, but there is some room for improvement. The closer the value to 1, the better the cluster separation; values close to 0 or negative mean that the clusters are not well-separated. Thus, this reflects a moderate clustering, meaning there are some data points that probably are not well-placed in their respective clusters.

Davies-Bouldin Index (1.3666)

The Davies-Bouldin index measures the average similarity of clusters, where a small value depicts a good clustering. A value of 1.3666 would indicate that the clusters are fairly well separated but with some degree of overlap. Ideally, the Davies-Bouldin Index should be as low as possible; values below 1 represent excellent clustering. This would mean further optimization in the K-means cluster to allow minimal overlap between clusters while keeping them separated.

Comparison with the True Labels in Project Data Set 3

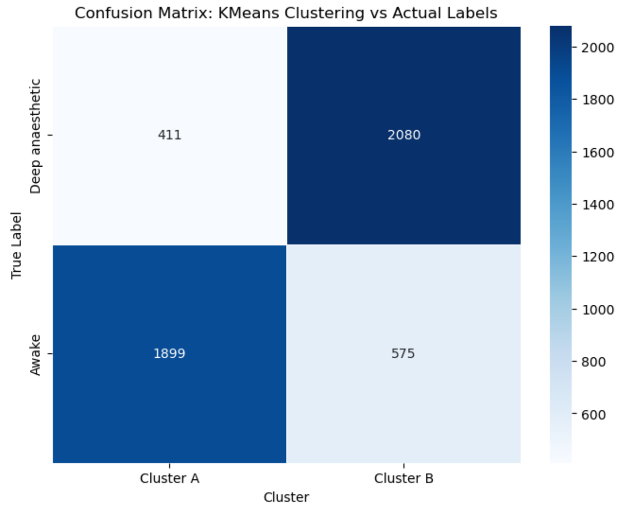


Figure 4 – Confusion Matrix: KMeans Clustering vs Actual Labels

The above confusion matrix shows the clustering results of K-means against the true labels from Project Data Set 3. The rows give the true labels, "Deep anaesthetic" and "Awake", while the columns give the predicted cluster by K-means as "Cluster A" and "Cluster B".

For Cluster A, there are 1899 instances of "Awake" correctly identified (True Positives for "Awake") and 411 instances of "Deep anaesthetic" incorrectly classified as "Awake" (False Negatives for "Deep anaesthetic"). That means some segments of "Deep anaesthetic" were incorrectly placed into the "Awake" cluster.

It correctly classifies 2080 instances of "Deep anaesthetic" as true positives for "Deep anaesthetic", but it classifies 575 instances of "Awake" as "Deep anaesthetic" - so, it is clear that some "Awake" segments are classified into "Deep anaesthetic".

Overall, K-means clustering fairly does the segregation between the two states of anaesthesia, with a larger number of correctly classified instances of "Deep anaesthetic" at 2080 as compared to "Awake" at 1899. In any case, there is quite a decent misclassification, especially with as many as 575 "Awake" instances falling into the "Deep anaesthetic" cluster. This indicates that while the model fares rather well, it can still be optimized to lower the rate of misclassifications.

5.2.2 Hierarchical Clustering

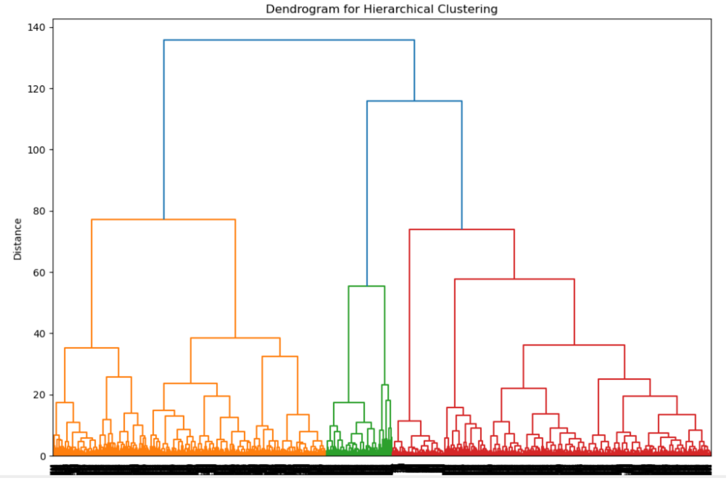


Figure 5 – Dendrogram for Hierarchical clustering

The above dendrogram, developed from Hierarchical clustering, is a visual representation of how the data is grouped based on their similarity. The vertical axis shows how dissimilar the clusters are. Thus, the lowest distances are the most similar. A process of merging the subclusters at increased distance shows three significant clusters from the orange, green, and red branches. It is possible to notice the orange cluster being more compact, hence closely grouped, probably corresponding to Group A. The green and red clusters are spread out, probably representing Group B, with higher variability among those segments.

Evaluation Metric	Value
Silhouette Score	0.2790
Davies-Bouldin Index	1.4260

Table 2 – Evaluation metrics for Hierarchical clustering

Silhouette Score (0.2790)

The Silhouette Score refers to a measure of how well each data point fits into their assigned cluster compared to other clusters. A score of 0.2790 implies that the clusters are somewhat distinct but not highly separated. Though the score is positive, it is generally low. This points to the same conclusion as reached earlier, that the clustering may not be perfect, and therefore, there is a possibility of some overlap between the clusters.

Davies-Bouldin Index (1.4260)

The Davies-Bouldin Index gives the average similarity between every cluster and its most similar one. The lower values indicate better clustering. A value of 1.4260 infers that there is an average overlap among the clusters; hence, the separation is not that good. This shows that though the clusters are somewhat separated, it is open to getting better separability for cluster distinction.

Comparison with the True Labels in Project Data Set 3

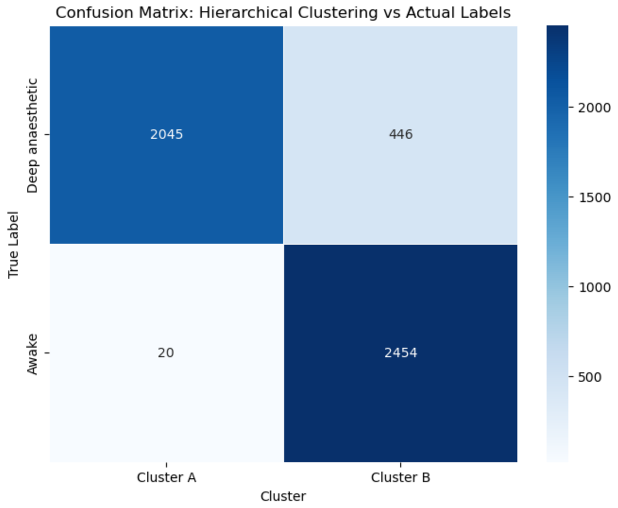


Figure 6 - Confusion Matrix: Hierarchical Clustering vs Actual Labels

Above is a confusion matrix comparing the Hierarchical clustering results with the true labels from Project Data Set 3. Entries for rows represent the true labels ("Deep anaesthetic" and "Awake"), and columns are the predicted clusters from Hierarchical clustering, named "Cluster A" and "Cluster B".

That is, "Deep anaesthetic" is correctly predicted 2045 times-True Positives for "Deep anaesthetic", whereas "Awake" was misrepresented as "Deep anaesthetic" 20 times which means there are 20 False Positives for "Awake". This indicates that most of the segments of "Deep anaesthetic" were rightly caught by the clustering algorithm while misinterpreting very few "Awake" segments.

In Cluster B, the "Awake" class is predicted correctly 2454 times, representing the True Positives, while "Deep anaesthetic" is wrongly classified as "Awake" 446 times, representing the False Negatives. The "Deep anaesthetic" misclassification rate in Cluster B is a bit higher compared to the "Awake" misclassification rate in Cluster A, and that may indicate that the clustering is less capable of discerning the "Deep anaesthetic" state in this cluster.

In general, Hierarchical clustering does a good job, classifying "Deep anaesthetic" cases to 2045 and "Awake" cases to 2454 accordingly. Still, the 446 misclassified "Deep anaesthetic" falling into Cluster B, along with 20 misclassified "Awake" cases into Cluster A, set a dimension for further improvement in the accuracy of clustering. The results of clustering are good but can be fine-tuned by feature selection or by tuning of clustering parameters in order to keep the rate of misclassification as low as possible.

6 Conclusion

We have seen from the above analysis that K-means and Hierarchical clustering successfully classified EEG data for Project Data Set 2 into two "Awake" and "Deep anaesthetic" states.

The two methods could potentially produce fairly decent separation between the two states, although K-means clustering came out to have a better performance on this aspect, given with some misclassifications.

K-means has provided better overall performance, with 2080 "Deep anaesthetic" and 1899 "Awake" instances classified correctly. However, it misclassified 575 "Awake" instances as "Deep anaesthetic," which means there was partial overlap. The Silhouette Score of 0.2879 maintains that there is a moderate separation between the clusters. The Davies-Bouldin Index is 1.3666, indicating that the clusters are somewhat separated but have a partial overlap in them. There is indeed further room for optimization, but based on overall cluster separation and efficiency, K-means is superior compared to Hierarchical clustering.

As for Hierarchical clustering, the result is quite similar, but the performance is relatively low. A Silhouette Score of 0.2790 and a Davies-Bouldin Index of 1.4260 indicate that cluster separation is somewhat weaker. Although it had 2045 "Deep anaesthetic" and 2454 "Awake" correct classifications, it classified 446 "Deep anaesthetic" as "Awake" and 20 "Awake" as "Deep anaesthetic." Based on the misclassifications along with the fact that it wrongly put the "Deep anaesthetic" instances in the "Awake" cluster much more than it did vice versa, it establishes that the Hierarchical clustering finds it difficult to separate the two states compared to K-means.

Therefore, while Hierarchical Clustering gives a broad overview of the data structure, K-means clustering is the more practical and valid tool in this study for developing the DOA index. It offers a good balance among cluster separation, low misclassification rate, and computational efficiency. Nevertheless, both methods retain some degree of misclassification that requires further refinement in feature selection and the selection of clustering parameters.

7 References

- Li, X., et al. (2019). Unsupervised Clustering of EEG Signals for Anesthetic Depth Estimation. *Journal of Clinical Monitoring*.

- Zhao, H., et al. (2020). Machine Learning and EEG: Applications in Psychiatry. *Healthcare Informatics Research*.

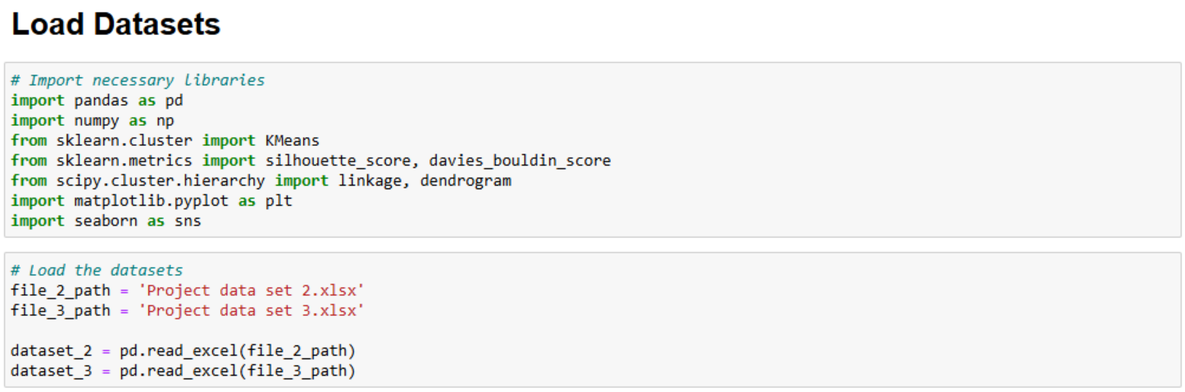
- Doi, M., et al. (2016). EEG-Based Depth of Anesthesia Monitoring Using Unsupervised Learning. *Anesthesia & Analgesia*.

- Hassan, A., et al. (2018). EEG Data Clustering for Anesthesia Monitoring: A Machine Learning Approach. *Journal of Medical Systems*.

8 Appendices

8.1 Source Code

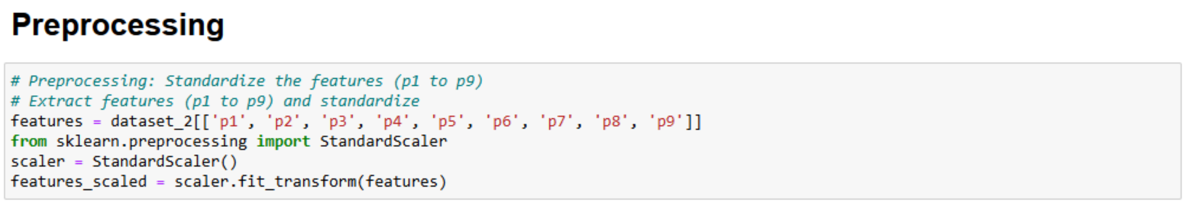
8.1.1 Data Loading and Preprocessing



Load Datasets

```python
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
from scipy.cluster.hierarchy import linkage, dendrogram
import matplotlib.pyplot as plt
import seaborn as sns
```

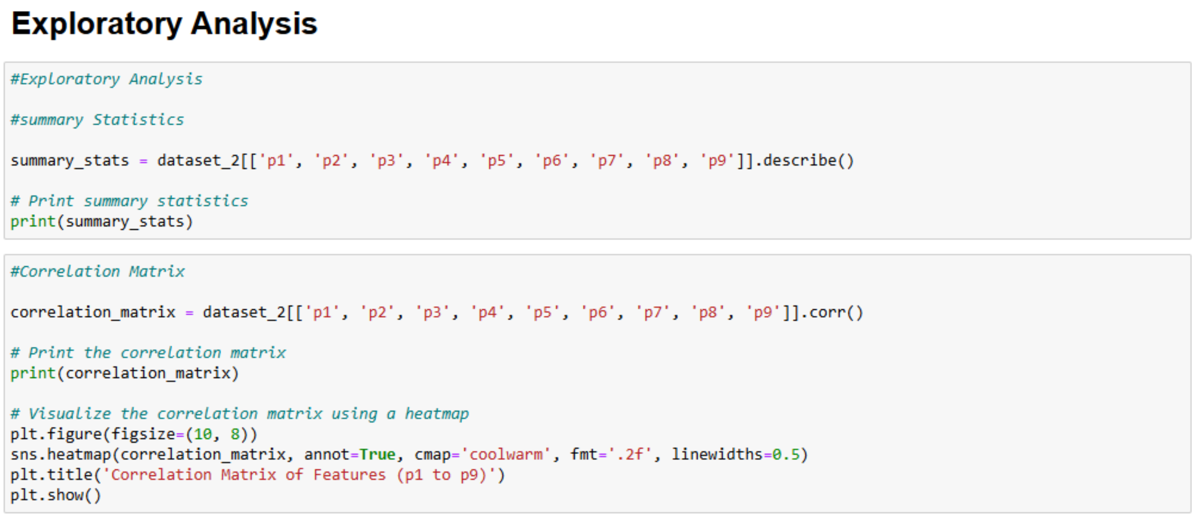
```python
# Load the datasets
file_2_path = 'Project data set 2.xlsx'
file_3_path = 'Project data set 3.xlsx'

dataset_2 = pd.read_excel(file_2_path)
dataset_3 = pd.read_excel(file_3_path)
```



Preprocessing

```python
# Preprocessing: Standardize the features (p1 to p9)
# Extract features (p1 to p9) and standardize
features = dataset_2[['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7', 'p8', 'p9']]
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

8.1.2 Exploratory Analysis

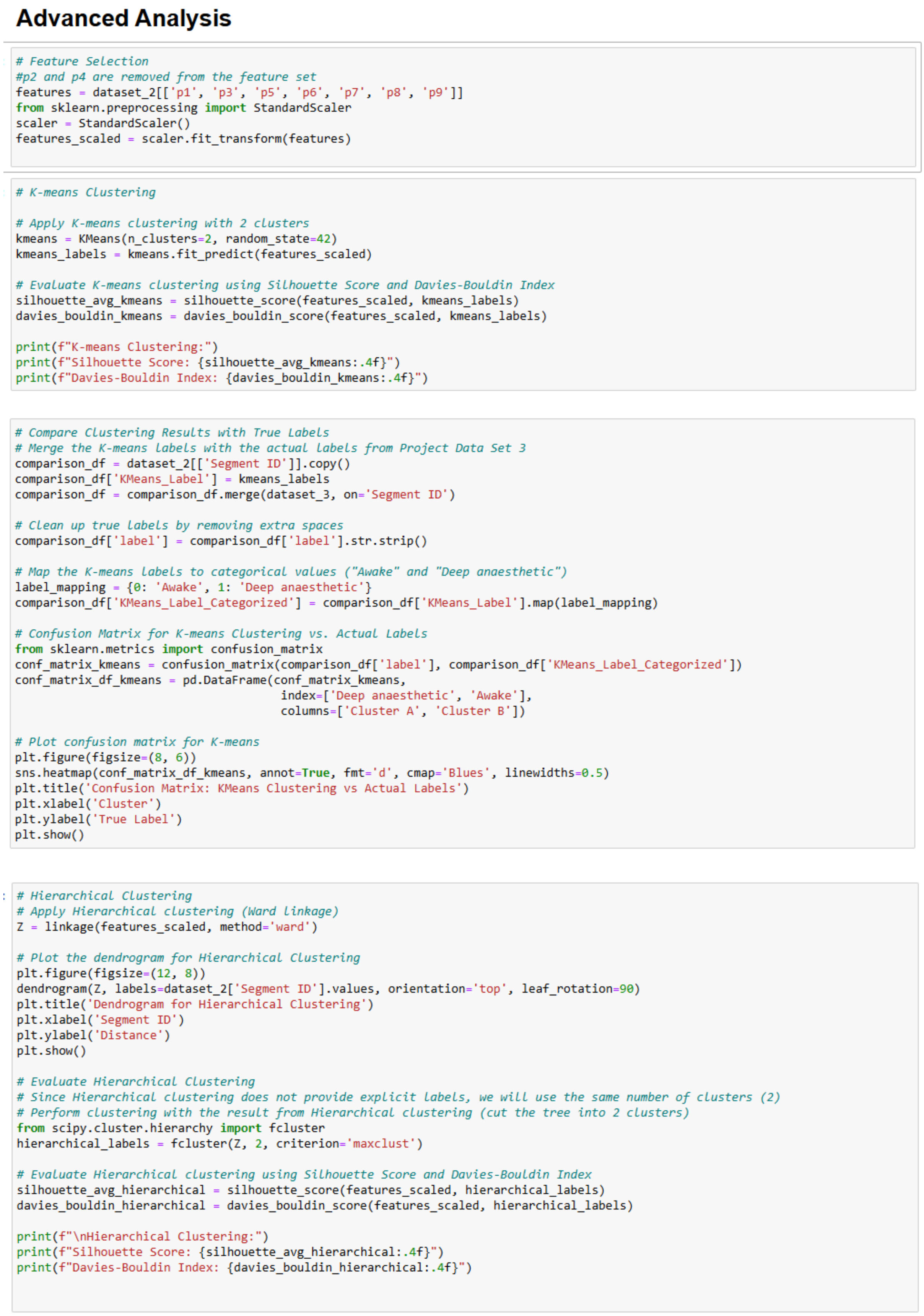


Exploratory Analysis

```python
#Exploratory Analysis

#summary Statistics

summary_stats = dataset_2[['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7', 'p8', 'p9']].describe()

# Print summary statistics
print(summary_stats)
```

```python
#Correlation Matrix

correlation_matrix = dataset_2[['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7', 'p8', 'p9']].corr()

# Print the correlation matrix
print(correlation_matrix)

# Visualize the correlation matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix of Features (p1 to p9)')
plt.show()
```

8.1.3 Advanced Analysis



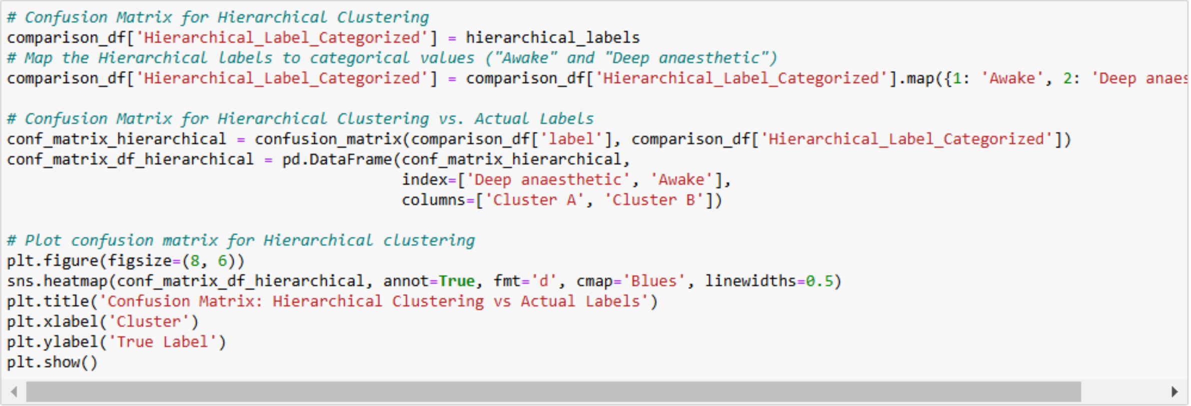
Advanced Analysis

```
# Feature Selection
#p2 and p4 are removed from the feature set
features = dataset_2[['p1', 'p3', 'p5', 'p6', 'p7', 'p8', 'p9']]
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)
```

```
# K-means Clustering

# Apply K-means clustering with 2 clusters
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans_labels = kmeans.fit_predict(features_scaled)

# Evaluate K-means clustering using Silhouette Score and Davies-Bouldin Index
silhouette_avg_kmeans = silhouette_score(features_scaled, kmeans_labels)
davies_bouldin_kmeans = davies_bouldin_score(features_scaled, kmeans_labels)

print(f"K-means Clustering:")
print(f"Silhouette Score: {silhouette_avg_kmeans:.4f}")
print(f"Davies-Bouldin Index: {davies_bouldin_kmeans:.4f}")
```

```
# Compare Clustering Results with True Labels
# Merge the K-means labels with the actual labels from Project Data Set 3
comparison_df = dataset_2[['Segment ID']].copy()
comparison_df['KMeans_Label'] = kmeans_labels
comparison_df = comparison_df.merge(dataset_3, on='Segment ID')

# Clean up true labels by removing extra spaces
comparison_df['label'] = comparison_df['label'].str.strip()

# Map the K-means labels to categorical values ("Awake" and "Deep anaesthetic")
label_mapping = {0: 'Awake', 1: 'Deep anaesthetic'}
comparison_df['KMeans_Label_Categorized'] = comparison_df['KMeans_Label'].map(label_mapping)

# Confusion Matrix for K-means Clustering vs. Actual Labels
from sklearn.metrics import confusion_matrix
conf_matrix_kmeans = confusion_matrix(comparison_df['label'], comparison_df['KMeans_Label_Categorized'])
conf_matrix_df_kmeans = pd.DataFrame(conf_matrix_kmeans,
                                     index=['Deep anaesthetic', 'Awake'],
                                     columns=['Cluster A', 'Cluster B'])

# Plot confusion matrix for K-means
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_df_kmeans, annot=True, fmt='d', cmap='Blues', linewidths=0.5)
plt.title('Confusion Matrix: KMeans Clustering vs Actual Labels')
plt.xlabel('Cluster')
plt.ylabel('True Label')
plt.show()
```

```
# Hierarchical Clustering
# Apply Hierarchical clustering (Ward linkage)
Z = linkage(features_scaled, method='ward')

# Plot the dendrogram for Hierarchical Clustering
plt.figure(figsize=(12, 8))
dendrogram(Z, labels=dataset_2['Segment ID'].values, orientation='top', leaf_rotation=90)
plt.title('Dendrogram for Hierarchical Clustering')
plt.xlabel('Segment ID')
plt.ylabel('Distance')
plt.show()

# Evaluate Hierarchical Clustering
# Since Hierarchical clustering does not provide explicit labels, we will use the same number of clusters (2)
# Perform clustering with the result from Hierarchical clustering (cut the tree into 2 clusters)
from scipy.cluster.hierarchy import fcluster
hierarchical_labels = fcluster(Z, 2, criterion='maxclust')

# Evaluate Hierarchical clustering using Silhouette Score and Davies-Bouldin Index
silhouette_avg_hierarchical = silhouette_score(features_scaled, hierarchical_labels)
davies_bouldin_hierarchical = davies_bouldin_score(features_scaled, hierarchical_labels)

print(f"\nHierarchical Clustering:")
print(f"Silhouette Score: {silhouette_avg_hierarchical:.4f}")
print(f"Davies-Bouldin Index: {davies_bouldin_hierarchical:.4f}")
```



```python
# Confusion Matrix for Hierarchical Clustering
comparison_df['Hierarchical_Label_Categorized'] = hierarchical_labels
# Map the Hierarchical labels to categorical values ("Awake" and "Deep anaesthetic")
comparison_df['Hierarchical_Label_Categorized'] = comparison_df['Hierarchical_Label_Categorized'].map({1: 'Awake', 2: 'Deep anaes

# Confusion Matrix for Hierarchical Clustering vs. Actual Labels
conf_matrix_hierarchical = confusion_matrix(comparison_df['label'], comparison_df['Hierarchical_Label_Categorized'])
conf_matrix_df_hierarchical = pd.DataFrame(conf_matrix_hierarchical,
                                           index=['Deep anaesthetic', 'Awake'],
                                           columns=['Cluster A', 'Cluster B'])

# Plot confusion matrix for Hierarchical clustering
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_df_hierarchical, annot=True, fmt='d', cmap='Blues', linewidths=0.5)
plt.title('Confusion Matrix: Hierarchical Clustering vs Actual Labels')
plt.xlabel('Cluster')
plt.ylabel('True Label')
plt.show()
```