D212 Data Mining II

OFM4 Clustering Techniques (Task 1)

Performance Assessment

Hillary Osei (Student ID #011039266)

Western Governors University, College of Information Technology

Program Mentor: Dan Estes

August 23, 2025

Table of Contents

Part I: Research Question

Section A1. Proposal of Question

The research question addressed is “Using KMeans clustering, can we identify patient groups by features heavily correlated to the length of their initial hospital visits?”

Section A2. Defined Goal

The goal of the analysis is to identify and describe clusters of patients based on continuous variables such as latitude, longitude, income, vitamin D levels, initial days, total charges, and additional charges. The clusters are evaluated to determine their relationships with hospital readmission rates to better inform patient care strategies.

Part II: Method Justification

Section B1. Explanation of Clustering Technique

K-means clustering is an iterative clustering algorithm that aims to reduce the sum of distances between data points and centroids. First, it initializes k centroids, which correspond to the number of clusters selected. The nearest centroids are assigned based on each data point’s distance to it based on the expectation maximization learning algorithm. The mean of all the points in each cluster is calculated and a centroid is assigned. This repeats over and over again until centroids no longer move significantly or until the maximum number of iterations is reached. (Kavlakoglu & Winland, 2024) The expected outcome is to reveal patient clusters with shared attributes that may influence readmission risk.

Section B2. Summary of the Technique Assumption

According to GeeksforGeeks, one key assumption of the K-means algorithm is that all clusters have approximately the same variance. This means that data points within a cluster are assumed to be distributed evenly around a centroid. This means that in each cluster, the distribution of data points around the center is about the same (Jain, 2025)

Section B3. Packages or Libraries List

The following Python libraries are used to support clustering analysis:

1. matplotlib.pyplot: Used to visualize cluster results, including PCA-reduced scatter plots and and elbow method plots to find the optimal number of clusters

2. sklearn.cluster.KMeans: Implements the K-means clustering algorithm
3. sklearn.preprocessing.StandardScaler: Normalizes continuous variables to ensure all features contribute equally to distance calculations in K-means
4. sklearn.decomposition.PCA: Reduces dimensionality of data for visualization purposes, allows 2D plotting of clusters
5. sklearn.metric.silhouette_score, davies_bouldin_score: Evaluates quality of clustering

Part III: Data Preparation

Section C1. Data Preprocessing

The goal of preprocessing is to normalize the continuous variables before applying the K-means clustering algorithm. Standardization ensures that each feature contributes equally to distance calculations, which is essential for accurate clustering results.

Section C2. Data Set Variables

The continuous variables used to address the research question are:

- Lat
- Lng
- Income
- VitD_levels
- Initial_days
- TotalCharge
- Additional_charges

Section C3. Steps for Analysis

To begin the data preparation process, the dataset was imported using the read_csv() function from the pandas library. An initial overview of the data was obtained using df.info(), which gives insight into the data type, names, and non-null counts for each variable. Next, duplicated records were checked using df.duplicated(). No duplicates were found. Next, the dataset was checked for missing values using df.isnull.sum() to make sure the data was complete. No missing values were found.

Then, all non-continuous variables were removed to retain only ones that are suitable for K-means clustering.

```
# Remove irrelevant variables
df = df.drop(columns=[
```

```
    'CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'County',
    'Zip', 'Population', 'Area', 'TimeZone', 'Job', 'Children', 'Age', 'Marital', 'Gender',
'ReAdmis',
    'Doc_visits', 'Full_meals_eaten', 'vitD_supp',  'Soft_drink', 'Initial_admin','HighBlood',
    'Stroke', 'Complication_risk', 'Overweight', 'Arthritis','Diabetes', 'Hyperlipidemia',
    'BackPain', 'Anxiety',  'Allergic_rhinitis','Reflux_esophagitis', 'Asthma', 'Services',
    'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'
])
```

The selected continuous variables include “Lat,” “Lng,” “Income,” “TotalCharge,” etc. The dataset was re-examined using df.info() to confirm that only the continuous variables remained.

Finally, the data was standardized using StandardScaler from the sklearn.preprocessing module. Standardization transforms features to have a mean of zero and a standard deviation of one (Brownlee, 2020) This is important for accurate clustering.

```
# Use Standard Scaler to standardize data
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
sc.fit(df)
scaled_data_array = sc.transform(df)
scaled_data = pd.DataFrame(scaled_data_array, columns = df.columns)
scaled_data.head()
```

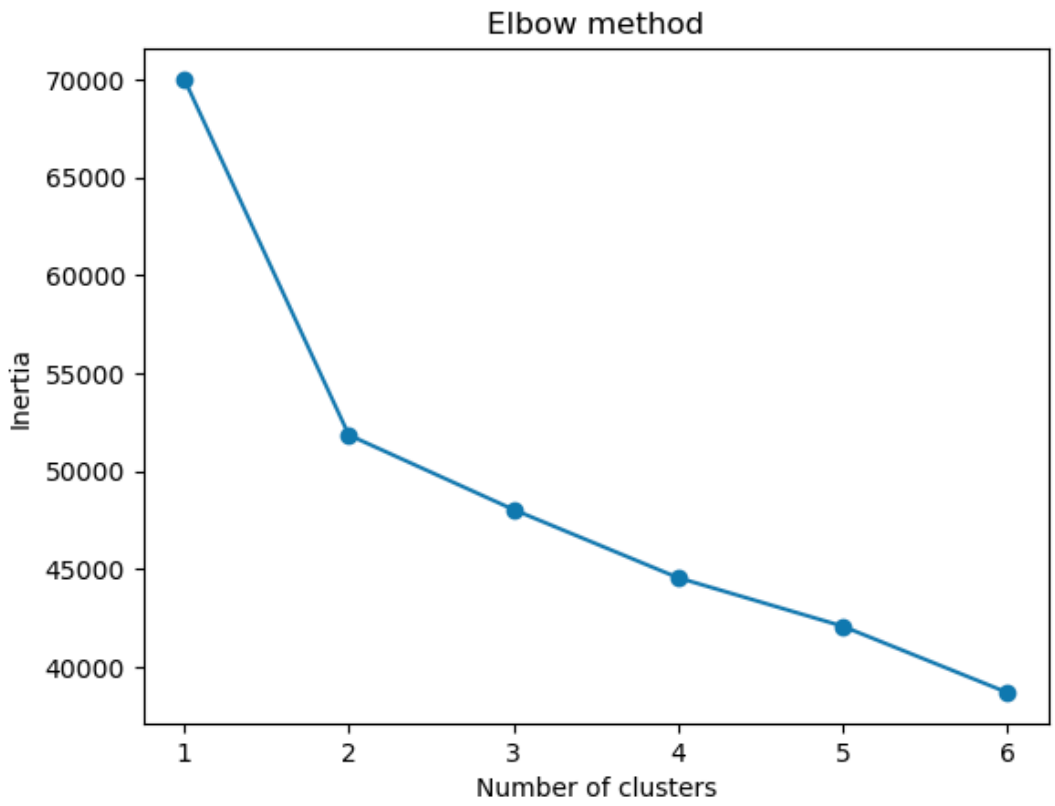
Section C4. Cleaned Data Set

The cleaned dataset is attached, titled “D212_scaled_task1.csv"

Part IV: Analysis

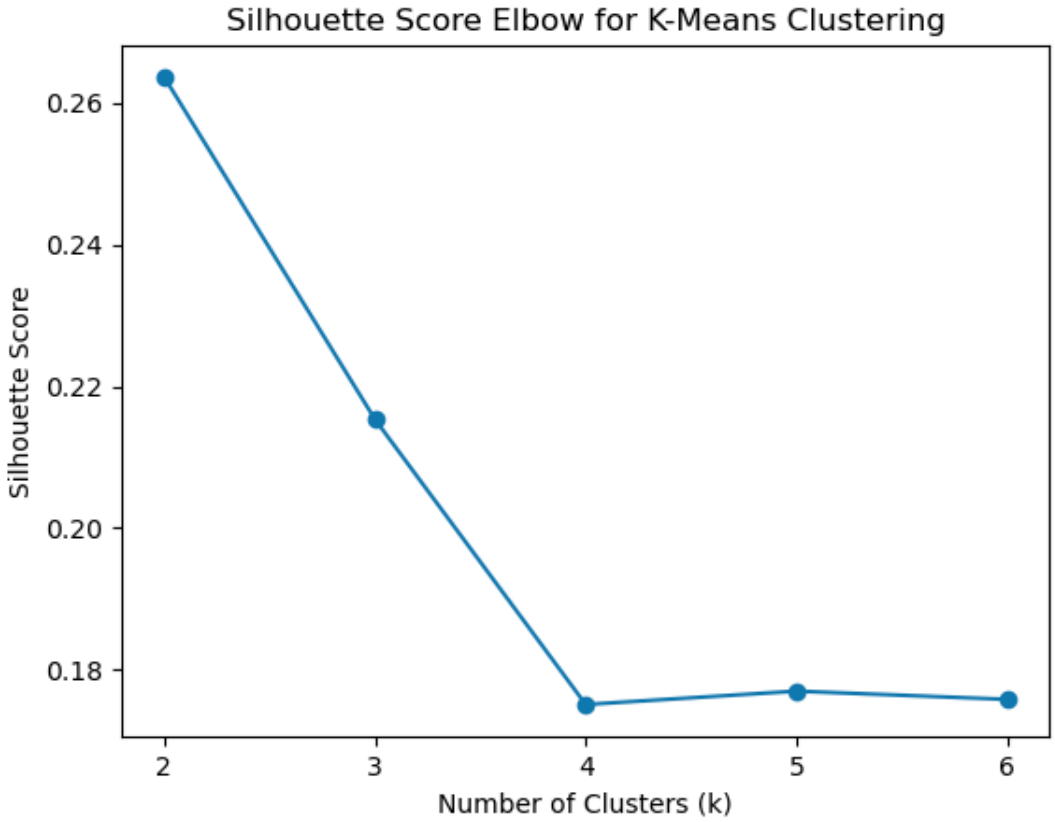
Section D1. Output and Intermediate Calculations

To determine the optimal number of clusters for the K-means algorithm, two methods were used: the Elbow Method and the Silhouette Score Method. The Elbow Method involves calculating the inertia, or within-cluster sum of squares (WCSS), for a range of cluster values from k=1 to k=6 (Jain, 2025). Inertia measures how compact clusters are, calculating the total variance within clusters (Leung, 2023) These values were plotted, and the elbow curve showed a bend at k=2, suggesting that 2 is the optimal number of clusters (Jain, 2025).



To validate this result, silhouette scores were computed for k values from 2 to 7. According to GeeksforGeeks, silhouette scores "[evaluate] how well each data point fits within its assigned cluster, and how distinctly separated it is from other clusters." (GeeksforGeeks, 2025) A silhouette score above 0.5 is considered good clustering while values close to 1.0 is considered strong separation (GeeksforGeeks, 2025). In the plot for silhouette scores, the highest silhouette occurs at k=2, with 0.26. As k increased beyond 2, the silhouette score declined, meaning adding more clusters might reduce cluster quality.

```
For n_clusters = 2, the silhouette score is 0.26365098313790236
For n_clusters = 3, the silhouette score is 0.21534728372534528
For n_clusters = 4, the silhouette score is 0.1751010846453575
For n_clusters = 5, the silhouette score is 0.1769898704393218
For n_clusters = 6, the silhouette score is 0.17582456597405954
```



Section D2. Code Execution

```
# Adapted from: W3Schools. (n.d.). Python Machine Learning - K-means.
# https://www.w3schools.com/python/python_ml_k-means.asp

inertias = []

for i in range(1,7):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_data)
    inertias.append(kmeans.inertia_)

plt.plot(range(1,7), inertias, marker='o')
plt.title('Elbow method')
```

```
plt.xlabel('Number of clusters')
plt.ylabel('Inertia')
plt.xticks(range(1,7))
plt.show()
```

```
# Find silhouette scores

ks = range(2, 7)
sil_scores = []

for k in ks:
    kmeans = KMeans(n_clusters=k)
    kmeans.fit(scaled_data)
    labels = kmeans.labels_
    sil_score = silhouette_score(scaled_data, labels)
    sil_scores.append(sil_score)
    print(f"For n_clusters = {k}, the silhouette score is {sil_score}")

# Plot the Silhouette Score Elbow
plt.plot(ks, sil_scores, '-o')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Silhouette Score')
plt.title('Silhouette Score Elbow for K-Means Clustering')
plt.xticks(ks)
plt.show()
```

```
# Adapted from: W3Schools. (n.d.). Python Machine Learning - K-means.
# https://www.w3schools.com/python/python_ml_k-means.asp

kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(scaled_data)
labels = kmeans.labels_
print(labels)
```

```
# Get cluster centers
kmeans.cluster_centers_
```

```
# Run PCA to reduce to 2 dimensions
pca = PCA(n_components=2)
X_pca = pca.fit_transform(scaled_data)
```

```
# Source: GeeksforGeeks. (2021, November 15). KMeans Clustering and PCA on Wine Dataset.
# https://www.geeksforgeeks.org/machine-learning/kmeans-clustering-and-pca-on-wine-dataset/

# Reduce clusters using PCA
centers = pca.transform(kmeans.cluster_centers_)

# reduced centers
centers
```

```
# Shows how each original feature contributes to PC1 and PC2
pd.DataFrame(pca.components_, columns=scaled_data.columns, index=['PC1', 'PC2'])
```

```
# Source: GeeksforGeeks. (2021, November 15). KMeans Clustering and PCA on Wine Dataset.
# https://www.geeksforgeeks.org/machine-learning/kmeans-clustering-and-pca-on-wine-dataset/

plt.figure(figsize=(8, 4))

# Scatter plot
plt.scatter(X_pca[:, 0], X_pca[:, 1],c=kmeans.labels_,  cmap='viridis')
plt.scatter(centers[:,0],centers[:,1],marker='x',s=100,c='red')
plt.xlabel('PCA1')
plt.ylabel('PCA2')
plt.title('Clusters')
plt.grid(True)
plt.tight_layout()
```

```
# Source: Cervantes, A. (n.d.). Interpreting and validating clustering results with K-means.
#
https://medium.com/@a.cervantes2012/interpreting-and-validating-clustering-results-with-k-means-e98227183a4d


# Validate clustering results

# Silhouette score
sil_score = silhouette_score(scaled_data, kmeans.labels_)
print(f"Silhouette Score: {sil_score}")

# Davies-Bouldin score
db_score = davies_bouldin_score(scaled_data, kmeans.labels_)
print(f"Davies-Bouldin Index: {db_score}")

# WCSS
wcss = kmeans.inertia_
print(f"Within-Cluster Sum of Squares (WCSS): {wcss}")
```

Part V: Data Summary and Implications

Section E1. Quality of the Clustering Technique

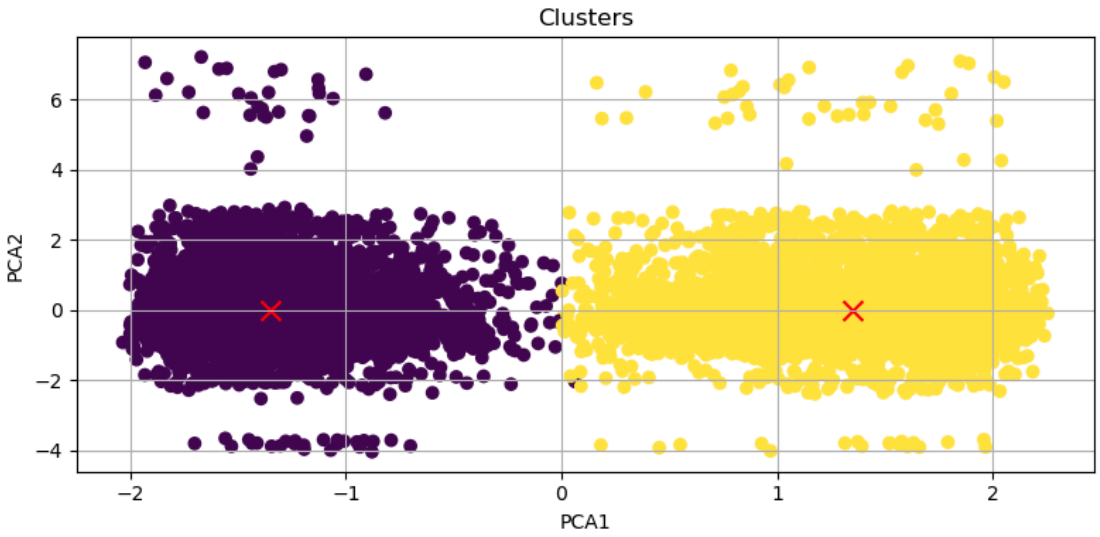
The quality of the clusters was evaluated using three metrics: Silhouette Score, Davies-Bouldin Index, and Within-Cluster Sum of Squares (WCSS)). The silhouette score for the cluster at k = 2 is at about 0.26, meaning that there is poor clustering (Zulfikar et al., n.d.). The Davies-Bouldin Index is at 1.57, confirming that the clusters are of poor quality (Barragan, 2024). WCSS is at 51833.78. This score is considered high, meaning that the data points in the cluster are not well-formed because they are spread farther away from the centroid (Barragan, 2024).

```
Silhouette Score: 0.26365098313790236
Davies-Bouldin Index: 1.5792154347299237
Within-Cluster Sum of Squares (WCSS): 51833.782601330204
```

Section E2. Results and Implications

After applying K-means clustering on the scaled dataset with the continuous variables, two clusters were identified as optimal based on the elbow method and silhouette score. To make

results easier to visualize and interpret, PCA was used to reduce the dimensionality of the data from 7 to 2 principal components, with the first principal component capturing the most variance in the data and the second one capturing the next highest. The two components are plotted in the graph, where each point represents a patient and is colored based on their cluster assignment (either purple or yellow). There is a visible but not strong separation between the clusters, which aligns with the relatively low silhouette score of 0.26. The Davies-Bouldin Index of 1.5792 and WCSS value of 51833.78 also further confirms that the clusters are not tightly compact or strongly separated.



The centroids of the PCA-transformed clusters are located at [-1.34706439, -0.00711865] for Cluster 0 (purple) and [1.34760333, 0.0071215]for Cluster 1 (yellow). Since PC1 is strongly influenced by “Initial_days” and “TotalCharge” with both having loadings of ~0.71, this suggests that Cluster 1 represents patients with longer hospital stays and higher total charges, while Cluster 0 contains patients with shorter stays and lower costs. The second component, PC2, is primarily influenced by “Lat” and “Lng.” However, the cluster centroids for PC1 and PC2 are small suggesting that location doesn’t have a major impact on how patients are grouped.

[31]:
```
# Shows how each original feature contributes to PC1 and PC2
import pandas as pd
pd.DataFrame(pca.components_, columns=scaled_data.columns, index=['PC1', 'PC2'])
```

[31]:

	Lat	Lng	Income	VitD_levels	Initial_days	TotalCharge	Additional_charges
PC1	-0.012385	-0.011393	-0.019039	-0.003094	0.706503	0.706831	0.024252
PC2	0.707635	-0.698869	-0.086107	0.058388	0.000265	-0.001082	-0.003272

Section E3. Limitation

One limitation is that K-means assumes spherical clusters and equal variance (Jain, 2025) which may not reflect real-world patient groupings. For example, some groups might have wide variations in age but similar income levels while other groups can vary. Additionally, some patient groups might be tightly clustered while others are more spread out and diverse. Another limitation is that K-means assigns a patient to only one group when it is possible that a patient can show characteristics of multiple groups, such as both being elderly and having multiple chronic medical conditions.

Section E4. Course of Action

Based on the results of the K-means clustering and PCA analysis, two main groups of patients were identified. One group (Cluster 0) includes patients with shorter hospital stays and lower total charges. These patients seem to require fewer resources and might represent less severe cases or more efficient care. For this group, hospitals could study what is working well, such as fast treatment planning or smooth discharges, and consider using those methods for other patients too. The other group (Cluster 1) has patients with longer stays and higher total costs. This may point to more complex medical needs, delays in care, or inefficient resource use. For this group, it might help to provide more follow-up support, better care coordination, or programs that manage chronic conditions early. Since features like income, vitamin D levels, and location (latitude and longitude) didn't strongly affect the clustering, the focus should stay on improving hospital care processes rather than targeting specific regions or patient backgrounds.

Part VI: Demonstration

Section F. Panopto

Link to Panopto here:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=c516be09-bc8e-4d5a-9a51-b343000a013a

Section G. Sources for Third-Party Code

Barragan, A. C. (2024, July 15). *Interpreting and Validating Clustering Results with K-Means*.

Medium. Retrieved July 31, 2025, from

https://medium.com/@a.cervantes2012/interpreting-and-validating-clustering-results-with-k-means-e98227183a4d

GeeksforGeeks. (2021, November 15). KMeans clustering and PCA on wine dataset. https://www.geeksforgeeks.org/machine-learning/kmeans-clustering-and-pca-on-wine-dataset/

W3Schools. (n.d.). Python machine learning – K-means clustering. https://www.w3schools.com/python/python_ml_k-means.asp

Section H. Sources

Barragan, A. C. (2024, July 15). *Interpreting and Validating Clustering Results with K-Means*. Medium. Retrieved July 31, 2025, from https://medium.com/@a.cervantes2012/interpreting-and-validating-clustering-results-with-k-means-e98227183a4d

Brownlee, J. (2020, August 28). *How to Use StandardScaler and MinMaxScaler Transforms in Python - MachineLearningMastery.com*. Machine Learning Mastery. Retrieved July 31, 2025, from https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/

GeeksforGeeks. (2025, June 23). *What is Silhouette Score?* GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/what-is-silhouette-score/

Jain, S. (2025, July 11). *Elbow Method for optimal value of k in KMeans*. GeeksforGeeks. Retrieved July 31, 2025, from

https://www.geeksforgeeks.org/machine-learning/elbow-method-for-optimal-value-of-k-in-kmeans/

Jain, S. (2025, July 23). *Demonstration of K-Means Assumptions*. GeeksforGeeks. Retrieved July 31, 2025, from https://www.geeksforgeeks.org/machine-learning/demonstration-of-k-means-assumptions/

Kavlakoglu, E., & Winland, V. (2024, June 26). *What is k-means clustering?* IBM. Retrieved August 21, 2025, from https://www.ibm.com/think/topics/k-means-clustering

Leung, V. (2023, December 16). *Understanding Inertia and Silhouette Coefficient - Key Metrics in Clustering Analysis*. Victor Leung TW. https://victorleungtw.com/2023/12/16/inertia/

Zulfikar, N., Karimov, E., & Sanabria, A. (n.d.). *How can you calculate the silhouette score for a clustering algorithm?* LinkedIn. https://www.linkedin.com/advice/0/how-can-you-calculate-silhouette-score-clustering-algorithm-w9bcc