**Performance Assessment**

OFM4 — OFM4 Task 1: Clustering Techniques

Bader Ale

Department of Information Technology, Western Governors University

D212 Data Mining II

2024

Table of Contents

[Part I: Research Question 3](#_Toc179451279)

[Part II: Technique Justification 3](#_Toc179451280)

[Part III: Data Preparation 5](#_Toc179451281)

[Part IV: Analysis 7](#_Toc179451282)

[Part V: Data Summary and Implications 7](#_Toc179451283)

# Part I: Research Question

The purpose of the analysis using k-means clustering is to answer the following “*In the context of a major hospital chain, "Can we identify distinct patient clusters based on demographic and clinical factors that are more likely to experience hospital readmission within 30 days?".*  For this task, k-means clustering was used on the given medical data set. This unsupervised learning algorithm was used to group patients according to the following common features: Age, Vitamin D levels during hospital stay and Readmission status. These clusters can help medical administrators understand patterns in these clusters that show them a high risk for readmission, which would form a proper basis on which effective targeted management interventions could be based.

# Part II: Technique Justification

The K-means algorithm is a kind of partitioning method where clusters are formed by the similarities of data points, in this case, patients, based on some predefined features such as age, comorbidities, length of stay, and location of discharge. First, the algorithm randomly picks some centroids and then assigns patients to the nearest centroid according to Euclidean distance. The centroids are recalculated when all the patients have been assigned to their respective clusters. The process repeats until the centroids no longer move significantly-when, in other words, the clusters are stable (scikit-learn developers, n.d.).

For this this k-means model, three variables were chosen for the clusters: Age, Vitamin D levels, and the target variable ReAdmission. The purpose of this analysis was to group patients into well-defined clusters that possess similar characteristics, hence allowing an analysis of interactions and influences these factors have on readmission.  
The k-means algorithm represents each patient as a data point in three-dimensional space, whose dimensions are Age, Vitamin D levels, and ReAdmission status. In an iterative process, it assigns each patient to a cluster by computing a distance measure between the data point representing a patient and the centroid of each cluster. It starts with the centroids chosen arbitrarily, but in every subsequent step, the centroid is updated to actually represent an 'mean' value of patients for a cluster. This process is repeated until convergence occurs; the centroids are not changing significantly anymore. We would expect the result of this to be the clear formation of patient clusters that reflect distinct relationships between Age, Vitamin D levels, and ReAdmission. For instance, one cluster may consist of a higher group of aged patients with low Vitamin D levels who are more predisposed to readmission, while another cluster can represent the younger patients who have higher Vitamin D levels and thus will present less likelihood of readmission. The information shall allow the administrators to understand how such factors as age and Vitamin D level result in readmissions to elaborate more focused preventive measures for high-risk groups.

One of the important assumptions of the k-means clustering algorithm is that clusters are spherical and equally sized. What this means is that the algorithm takes on the assumption that data points inside each cluster are evenly distributed around a central point, called a centroid, and the distance between points and their centroid is relatively uniform across all clusters. This works well when the clusters of data are near symmetric. The algorithm can sometimes fail to capture the exact underlying structure of the data set for oddly shaped data or of varying cluster size (scikit-learn developers, n.d.).

The packages used for this assessment include pandas for handling the data structures, seaborn to visualize the data and SKLearn which was used as the main package for the K-means learning algorithm. SKLearn was also used to scale the data before running K-means as well as using the Label Encoder function to encode the ‘ReAdmis’ variable from Yes/No to numerical values.

# Part III: Data Preparation

One important goal in preparing the data for k-means clustering is to ensure that the variables are on a similar scale. Since k-means calculates distances between data points, variables with larger ranges can disproportionately influence the clustering results. For example, if "Age" values range from 20 to 90, and "Vitamin D levels" vary between 10 and 50, the algorithm may give more weight to Age simply because its numerical range is larger. To address this, we need to standardize the continuous variables, so they all contribute equally to the analysis. This step ensures that the clustering results are not skewed by differences in measurement scales.

For this assessment, three variables were used: TotalCharge, Initial\_days and ReAdmission. TotalCharge is a continuous variable representing the total cost associated with a patient's initial hospital stay, offering insight into the financial aspect of care. Initial Days, another continuous variable, captures the length of the initial hospital stay, providing valuable information about the duration of treatment. The third variable, Readmission, is categorical and indicates whether a patient was readmitted after discharge. To ensure that the k-means clustering algorithm can process this variable effectively, LabelEncoder was used to convert Readmission from categorical (Yes/No) into a continuous numerical format.

To prepare the data for k-means clustering, the first step involved importing the dataset using pandas. This allowed for an initial inspection of the data and identification of relevant features for the analysis. The dataset was loaded with the following code:

1. df = pd.read\_csv('D:\GitHub Repos\WGU\_MSDA\D212\_Data Mining\medical\_clean.csv')

2. df.head()

Once the data was loaded, the next step was data cleaning. This involved removing unnecessary columns, such as customer identifiers and geographic details, to focus on the key variables relevant to clustering. These irrelevant columns were dropped using the following code:

1. df\_kmeans = df.copy()

2. df\_kmeans.drop(['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'City', 'Zip', 'Lat', 'Lng',

3. 'Population', 'Area', 'TimeZone', 'State', 'County', 'Job', 'Marital',

4. 'Gender', 'Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke', 'Complication\_risk',

5. 'Overweight', 'Arthritis', 'Diabetes', 'Hyperlipidemia', 'BackPain', 'Anxiety',

6. 'Allergic\_rhinitis', 'Reflux\_esophagitis', 'Asthma', 'Services', 'Item1', 'Item2'], axis=1, inplace=True)

7. df\_kmeans.head()

8.

Next, the categorical variable ReAdmis (representing whether a patient was readmitted) was converted into a numerical format using LabelEncoder. This step was necessary to include categorical data in the k-means clustering algorithm, which only works with numerical inputs. The encoding was done as follows:

1. from sklearn.preprocessing import LabelEncoder

2.

3. enc = LabelEncoder()

4. df\_kmeans['ReAdmis'] = enc.fit\_transform(df\_kmeans['ReAdmis'])

Following this, the continuous variables like TotalCharge and Initial Days were standardized using StandardScaler. Standardization ensures that each feature contributes equally to the distance calculations used in the k-means algorithm, which is sensitive to scale. The code for this step was:

1. from sklearn.preprocessing import StandardScaler

2.

3. scaler = StandardScaler()

4. df\_means\_columns = df\_kmeans.columns

5. df\_kmeans\_scaled = pd.DataFrame(scaler.fit\_transform(df\_kmeans), columns=df\_means\_columns)

6. df\_kmeans\_scaled.head()

These preprocessing steps—data cleaning, encoding categorical variables, and standardizing continuous variables—ensured the dataset was prepared for effective clustering using the k-means algorithm. A copy of the cleaned dataset is provided for review.

# Part IV: Analysis

The Elbow Method was chosen to determine the most appropriate number of clusters (3) in the dataset. This technique works by running the k-means algorithm over a range of cluster values and calculating the within-cluster sum of squares (WCSS) for each option. The WCSS reflects how well data points fit within each cluster, with lower values signifying tighter groupings. As the number of clusters increases, the WCSS naturally decreases because more clusters provide more flexibility in grouping data points. However, after a certain point, the reduction in WCSS slows down, forming an "elbow" in the graph. This elbow point suggests the ideal number of clusters, as adding more clusters beyond this point yields minimal improvement in the tightness of the groupings. This method provides a visual way to assess the optimal number of clusters and is commonly used due to its simplicity and clarity. While the exact point of the elbow can sometimes be subjective, it offers a useful starting point for determining cluster numbers, especially when used in combination with other metrics like the Silhouette Score to validate the results (Basil, 2024).

A graph with a line

Description automatically generated

1. from sklearn.cluster import KMeans

2.

3. # Create an empty list to store inertia values

4. inertia = []

5.

6. # Fit the k-means model for cluster values ranging from 1 to 10

7. for k in range(1, 11):

8.     kmeans = KMeans(n\_clusters=k, random\_state=42)

9.     kmeans.fit(df\_kmeans\_scaled)

10.     inertia.append(kmeans.inertia\_)

11.

12. # Plot inertia vs. number of clusters (k)

13. plt.figure(figsize=(8,5))

14. plt.plot(range(1, 11), inertia, marker='o')

15. plt.title('Elbow Method for Optimal k')

16. plt.xlabel('Number of Clusters (k)')

17. plt.ylabel('Inertia')

18. plt.show();

19.

This code is designed to help determine the optimal number of clusters for a k-means clustering model using the Elbow Method. It starts by creating an empty list to store the inertia values, which measure how tightly data points are grouped within clusters. Lower inertia values indicate better, more compact clustering. The code then fits the k-means algorithm for a range of cluster numbers, from 1 to 10, by looping through different values of `k` (the number of clusters). For each value of `k`, the model calculates the inertia and stores it in the list. Finally, the inertia values are plotted against the number of clusters, with the resulting graph displaying the "elbow" point, which helps identify the optimal number of clusters. The elbow is the point where the reduction in inertia slows down, indicating that adding more clusters beyond this point provides little improvement in clustering performance.

# Part V: Data Summary and Implications

## Quality of Data Clusters

This cluster produced in the analysis is a solid grouping of the data points based on such variables as TotalCharge, Initial Days, and Readmission. The model has found groupings using the Elbow Method in determining the optimal number of clusters that showed internal cohesion-well and how the data points were fitted within their cluster-and separation from other clusters. It is seen that the elbow point's reduction in inertia denotes that these clusters provide meaningful distinctions between sets of patients, indicating a good quality cluster with generally very useful insights into the pattern of patient readmission.

## Results and Implications

The clustering results indicate clear patterns among patients based on their hospitalization costs, length of stay, and likelihood of readmission. For instance, one group may consist of patients who had high total hospital charges and long initial stays, which correlates with a higher probability of being readmitted. Another group may consist of patients with shorter stays and lower costs, who are less likely to return. These findings can guide the hospital chain in focusing its resources on patients who fall into high-risk clusters. By identifying these groups, the hospital can implement more effective post-discharge care plans and other intervention strategies, such as remote monitoring or follow-up visits, which could reduce readmission rates and improve patient outcomes.

## Limitations

A notable limitation of the analysis is the absence of data on the original reason for hospitalization. Without this key piece of information, the analysis is based solely on general characteristics like cost and length of stay, which may not fully capture the factors that influence readmission rates. The lack of medical condition-specific data means the clusters might overlook important nuances in patient care and outcomes. While the current clustering model offers insights into readmissions based on available variables, including diagnosis or treatment details in future analyses would likely improve the accuracy and relevance of the results.

## Recommendations

Based on the results, it is recommended that the hospital chain focus its efforts on the patient groups identified as high-risk for readmissions, particularly those with high hospital costs and longer initial stays. Implementing targeted interventions, such as personalized post-discharge care, regular follow-ups, or leveraging technology for remote patient monitoring, could reduce readmission rates. Additionally, incorporating more detailed medical data, such as the reason for the initial hospitalization, would allow for even more refined strategies to be developed. By focusing on these high-risk groups and enhancing the use of patient data, the hospital can reduce penalties from readmissions and improve overall patient care outcomes.

Works Cited

Basil. (2024, September 27). *Elbow Method for Optimal Cluster Number in K-Means*. Retrieved October 2024, from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/

scikit-learn developers. (n.d.). *2.3. Clustering*. Retrieved October 2024, from Unsupervised Learning: https://scikit-learn.org/stable/modules/clustering.html#k-means