Patient Segmentation Using K-Means Clustering

Ashley Traore

Business Question

Can K-Means clustering identify actionable patient clusters using Age and Initial Days?

Goal of Analysis

The goal of this analytics is to uncover meaningful and actionable patient segments using Age and Initial Days through K-Means clustering, to help the hospital make data-informed decisions to improve operational efficiency, personalize care strategies, and optimize resource allocation.

Cluster Technique and Outcomes

This analysis leverages K-Means clustering to segment patients based on two key features, age and initial length of hospital stay. K-Means is an unsupervised machine learning algorithm that partitions data into distinct groups by minimizing the distance between each data point and its assigned cluster center. In this analysis, the algorithm evaluates similarities in age and hospital stay duration to identify naturally occurring patient groupings, without relying on pre-labeled outcomes or diagnoses.

The clustering process begins by selecting a number of cluster centers (K), then iteratively grouping patients based on their proximity to these centers in the two-dimensional space formed by the selected variables. As the algorithm refines the cluster assignments, it seeks to minimize the within-cluster variation, thereby ensuring that each group is as internally cohesive as possible.

The expected outcome is the emergence of clearly defined clusters representing different patient profiles, such as younger patients with short stays or older patients with extended care needs. These segments could provide meaningful insight into patterns of care utilization, allowing the hospital to develop more targeted interventions, streamline resource allocation, and ultimately improve patient outcomes. By visualizing and interpreting these clusters, the analysis can guide strategic planning without requiring prior assumptions about patient types.

One Assumption of K-Means

A key assumption of K-Means is that the data forms clusters of similar size (Editor, 2023). K-Means performs optimally when the clusters contain a roughly equal number of observations and exhibit similar spread in the feature space. When clusters vary widely, for example, one being large and diffuse while another is small and dense, the algorithm may incorrectly assign points or pull centroids toward disproportionately sized groups. To reduce the impact of this limitation, data should be standardized so that all features contribute equally to distance calculations. Additionally, dimensionality reduction techniques like Principal Component Analysis (PCA) can help reveal more compact and balanced cluster structures by simplifying the feature space.

B3. Libraries Used

Libraries	**Usage**
pandas	Used to handle and prepare the data.
numpy	For performing numerical operations.
sklearn.preprocessing import StandardScaler	Used to scale the data.
from scipy.stats import zscore	Used to calculate z-score.
sklearn.cluster import KMeans	To perfom K-Means clustering
from sklearn.metrics import silhouette_score	Used to calculate the silhouette score for the model
Matplotlib.pyplot and seaborn	Used for data visualization.

One Data Preprocessing Goal

Before applying K-Means clustering, several key preprocessing steps are essential to ensure reliable and meaningful results. These include scaling numerical features, addressing missing values and duplicates, and cleaning outliers that could skew the model. Among these, feature scaling is especially critical, as K-Means relies on distance calculations and assumes that all features contribute equally. Proper scaling helps minimize the impact of K-Means' inherent assumption that clusters are of similar size and density, reducing the risk of biased or misleading group assignments.

Variables Used in Analysis

Variables	**Categorical/Continuous**
Age	Continuous
Initial_days	Continuous

Data Preparation

1. Import the required libraries.

   ```
   # Import Libraries
   import pandas as pd
   import numpy as np
   from sklearn.preprocessing import StandardScaler
   from sklearn.cluster import KMeans
   import seaborn as sns
   import matplotlib.pyplot as plt
   from sklearn.metrics import silhouette_score
   from scipy.stats import zscore

   # Change setting in pandas to display all columns
   pd.options.display.max_columns = None
   ```

2. Load the data into a pandas dataframe and view the data.

   ```
   # Import medical dataset
   medicalDF =pd.read_csv(r"C:\Users\ashle\Desktop\MSDA WGU\Data Mining 2
   -D212\task 1\dataset\medical_clean.csv")

   medicalDF
   ```

3. Explore the dataset through medicalDF.info() to examine the data types of each column. Then, use the describe() method on the Initial_days and Age columns to generate summary statistics and gain insight into their distributions.

   ```
   medicalDF.info()
   medicalDF[['Initial_days', 'Age']].describe()
   ```

4. Create a copy of the data frame for data cleaning, then identify any duplicates. No duplicates were identified.

   ```
   # Create a copy of the data frame for cleaning the data.
   medicalClean = medicalDF.copy()

   # Identify duplicates
   duplicates = medicalClean[medicalClean.duplicated()]
   print(duplicates)
   ```

5. Identify any null values. No missing data was identified.

   ```
   # Identify null values in the dataframe.
   missingValues = medicalClean.isnull().sum().sum()
   print(missingValues)
   ```

6. Identify and remove outliers using Z-score capping with a threshold set at 3 standard deviations from the mean.

   ```
   # Identify and remove outliers using the z-score with a threshold set at 3
   standard deviations from the mean
   # Select numeric columns to check for outliers
   numeric_cols = medicalClean.select_dtypes(include=[np.number]).columns

   # Calculate Z-scores
   z_scores = zscore(medicalClean[numeric_cols])

   # Set a Z-score threshold, 3 standard deviations from the mean
   threshold = 3

   # Identify rows where any column exceeds the threshold
   outliers = z_scores.abs() > threshold
   ```

```
#Print a count of rows with outliers
rows_with_outliers = outliers.any(axis=1).sum()
print(f"Number of rows with outliers: {rows_with_outliers}")

# Remove the outliers
medicalClean_non_outliers = medicalClean[~outliers.any(axis=1)]

# Print the count of outliers removed for each column.
print(f"Removed {outliers.sum()} outliers.")

# Print the row count of the dataframe after removing outliers
print(medicalClean_non_outliers.shape)
```

7. Scaling the features for analysis.

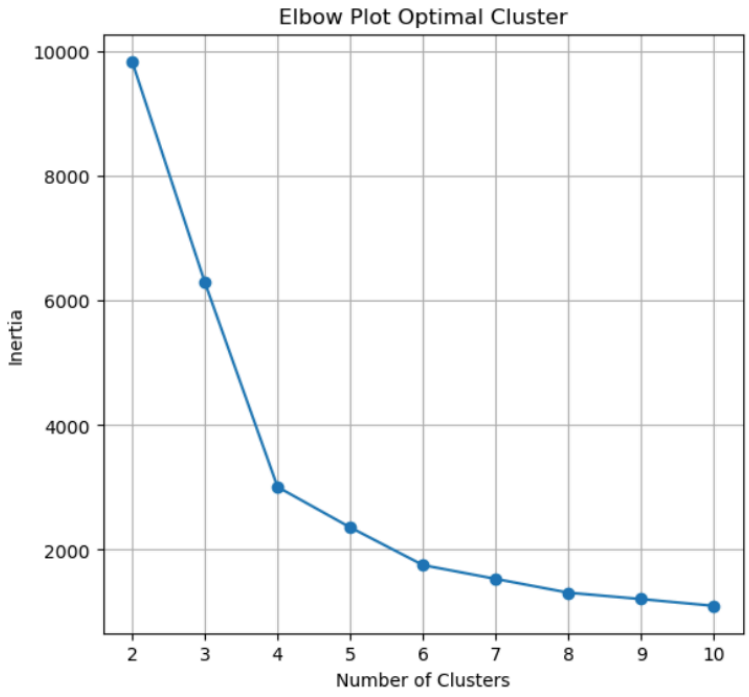
```
# Assign a features variable with the features used in the clustering
features = medicalClean_non_outliers[['Age', 'Initial_days']]
# Scale the features
scaler = StandardScaler()
scaled_data = scaler.fit_transform(features)
# Setting the scaled data to a pandas dataframe
scaled_data = pd.DataFrame(scaled_data, columns=features.columns)
scaled_data
```

8. Exporting the cleaned data.

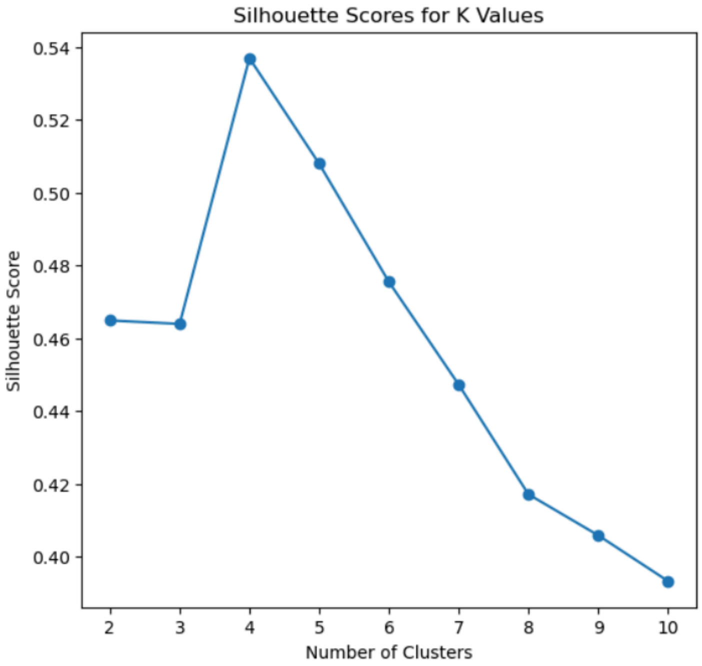
```
# Export cleaned data
scaled_data.to_csv(r'C:\Users\ashle\Desktop\MSDA WGU\Data Mining 2
-D212\task 1\cleaned data\scaled_data.csv')
```

Optimal Number of Clusters

The optimal number of clusters for this analysis was determined to be four. This was identified using an elbow plot, which visualized the within-cluster sum of squares (WCSS) for values of k ranging from 2 to 10. The plot reveals a clear elbow at k = 4 where the rate of improvement in model fit begins tapering off, indicating that adding more clusters beyond this point yields diminishing returns.



Alongside the elbow plot, a silhouette score plot was generated to help determine the optimal number of clusters. The visualization reveals a distinct peak at k = 4, indicating that this configuration yields the highest clustering quality based on silhouette values.



In conclusion, both the elbow plot and silhouette analysis pointed to an optimal cluster count of four. This consensus provides strong support for selecting k = 4 as the most suitable configuration, striking a balance between compactness within clusters and separation between them.

Quality of Clusters

To assess the clustering quality, the silhouette score was used, a widely used metric that evaluates how well data points fit within their assigned clusters. The score ranges from -1 to +1, where values closer to 1 indicate well-defined, separated clusters, and values near or below 0 suggest overlapping or poorly formed groupings. According to Islam, a silhouette score above 0.5 reflects strong clustering performance. The model achieved a score of 0.54, indicating a reasonably good structure with moderate cohesion and separation among clusters.

Results and Implications

The centroid values from the K-Means clustering analysis show four clear groups of patients based on their age and how long they stayed in the hospital. Cluster 0 includes younger patients who had longer stays, which may point to unexpected health issues or special care needs. These patients could benefit from closer monitoring and customized follow-up care. Cluster 1 also includes younger patients, but with shorter hospital stays. This group likely reflects routine or low-risk cases and offers chances to make care more efficient or shift some services outside the hospital.

Cluster 2 is made up of older patients who stayed in the hospital longer. This may be due to chronic conditions or more complex health problems. These patients could benefit from special senior-focused care plans, early discharge planning, and better coordination between teams. Cluster 3 consists of older patients with short hospital stays, likely for well-planned procedures or because their care was managed effectively. This group could be a model for delivering quality care to older adults. Overall, these clusters provide useful insights that can help the hospital improve planning, manage resources, and deliver more personalized care.

One Limitation of the Analysis

A key limitation of this analysis is the challenge of validating results in unsupervised learning. Since K-Means clusters data without using predefined labels, there’s no direct way to confirm whether the resulting groups reflect “correct” or clinically meaningful categories. Unlike supervised methods, where outcomes guide evaluation, clustering relies entirely on patterns within the input data.

Additionally, the analysis relies on only two variables: age and initial length of stay. While these features are meaningful, they may not capture the full complexity of what influences patient profiles or outcomes. As a result, the clusters might oversimplify patient needs or miss hidden patterns that more features could reveal.

Recommendations by Cluster:

- Cluster 0: Younger patients with longer hospital stays
    - This group likely includes individuals experiencing unexpected complications or requiring additional care. To better support these patients, the hospital should implement closer monitoring protocols and conduct case reviews to identify potential delays or gaps in treatment. Enhancing care coordination and assigning dedicated support teams could help reduce unnecessary length of stay and improve overall outcomes.
- Cluster 1: Younger patients with shorter hospital stays
    - These patients appear to represent low-acuity or routine cases that are efficiently managed. The hospital should continue to support this efficiency

by optimizing workflows and exploring additional opportunities for streamlining care. This cluster could also be used as a reference point for developing best practices across other patient segments.

- Cluster 2: Older patients with extended hospital stays
  - This segment likely reflects patients with chronic conditions or complex medical needs. To address this, the hospital should implement targeted interventions such as targeted care planning or effective discharge strategies. These efforts can improve patient outcomes while optimizing the use of hospital resources.
- Cluster 3: Older patients with shorter hospital stays
  - These patients may be benefiting from well-coordinated care plans or undergoing planned procedures. The hospital should analyze the care strategies used for this group and consider replicating successful elements across clusters with longer stays. This cluster can serve as a valuable benchmark for effective and efficient care in older populations.

Sources:

Kmeans. scikit. (n.d.-c). https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html