Western Governors University

# D212 – Data Mining II – Task 1

By Krista Moik

# Table of Contents

**Part I: Research Question**

**A1. Proposal of Question**

From the medical_clean dataset, **can K-Means Clustering provide useful insights or groupings of patients?** By understanding the patients that are being treated at a hospital, the hospital can obtain better insights into the needs and trends within the patient data to make data-driven decisions to reach business goals.

**A2. Defined Goal**

My goal is to use K-Means Clustering to create clusters grouping similar datapoints together to discover any patterns or trends that may not have been initially visible (Towards Data Science, 2018). By doing this, I will be able to identify the patterns and trends that emerge to yield more information about the patients the hospital is serving that could help make data-driven decisions to improve efficiency and show trends in the data. I will focus on Age and Income, amongst other variables in the dataset. I want to see if there is a relationship between age and income (one may assume the older you are before retirement age, the more money you make, but that is not always true), but I wonder if older patients with lower income require more admission days or have higher rates of admission as they may not have anyone to care for them after discharge or cannot afford home care. The clusters I create in this task may uncover trends that could help answer these questions.

**Part II: Technique Justification**

**B1. Explanation of the Clustering Technique**

K-Means Clustering will randomly create k-number of clusters where the center is called the centroid, which is the mean of the data points in the cluster, and then group datapoints to the nearest cluster using its sum of squared distance from the centroid. This form of unsupervised machine learning will repeat this process until the centroids are stabilized or the defined number of iterations is reached (Towards Data Science, 2018).

The expected outcome of this technique is to find patterns and trends in the dataset based on the clusters.

**B2. Summary of the Technique Assumption**

One assumption of using K-Means Clustering is that the clusters are even and spherical in shape. This assumption can reduce accuracy (Towards Data Science, 2018).

**B3. Packages of Libraries List**

I will use pandas for its data frame and data manipulation capabilities. I will then import matplotlib and seaborn for their visualizations. From sklearn, I will use KMeans, StandardScaler, and Silhouette_score. These 3 packages will allow me to create a KMeans model, scale the data in the model, and measure the fit of the technique on the data. I will also import warnings to ignore the filter warnings in my code.

**Part III: Data Preparation**

**C1. Data Preprocessing**

Probably the most important preprocessing goal for this task is to use the StandardScaler to normalize and scale the data using z-scores. Since K-means uses distance to group data points it is important that the data points have been normalized to reduce errors.

**C2. Data Set Variables**

I chose the numeric variables that I thought would provide the most useful information. My data set variables are:

| Variable | Data Type |
|---|---|
| Children | Discrete |
| Age | Continuous |
| Income | Continuous |
| Initial_days | Continuous |
| TotalCharge | Continuous |

**C3. Steps of Analysis**

First, I imported the packages I would be using:

```
#import packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
import warnings
warnings.filterwarnings('ignore')
```

Then I loaded the data:

```
#Load medical_clean CSV
df=pd.read_csv('C:/Users/Kmoik WGU/Desktop/D212/medical_clean.csv')
```

Next, I viewed the data:

```
df.head()
```

 ●●●

```
df.info()
```

I searched for duplicates and nulls:

```
#search for duplicates
print(df.duplicated().value_counts())
```

False 10000 ●●●

```
#check for null values - even though view of data indicates no nulls
df.isnull().sum()
```

I created a reduced dataset with just the variables I would use in my analysis:

```
red_df=df[['Children', 'Age', 'Income', 'Initial_days', 'TotalCharge']]
```

I visualized and obtained statistical information on my variables: (Western Governors University, n.d.):

```
clusterdata=red_df.describe().round(2)
clusterdata
```

I scaled my data (Western Governors University, n.d.):

```
#normalize data using z-score
scaler=StandardScaler()
```

```
#scaled df
scaled_df=scaler.fit_transform(red_df)
```

```
scaled_df=pd.DataFrame(scaled_df)
scaled_df
```

I created the K-Means model (Western Governors University, n.d.):

```
#create k means object
k_model=KMeans(n_clusters=3, n_init=25, random_state=300)
k_model.fit(scaled_df)
```

I saved my reduced and scaled data set as a new CSV:

```
#save to CSV
scaled_df.to_csv('C:/Users/Kmoik WGU/Desktop/KMoikD212_WGUmedical.csv')
```

Further steps are included in Part IV of this report. A copy of my full code used can be found in the attached ipynb and pdf documents titled: KMoikD212Code1

**C4. Cleaned Data Set**

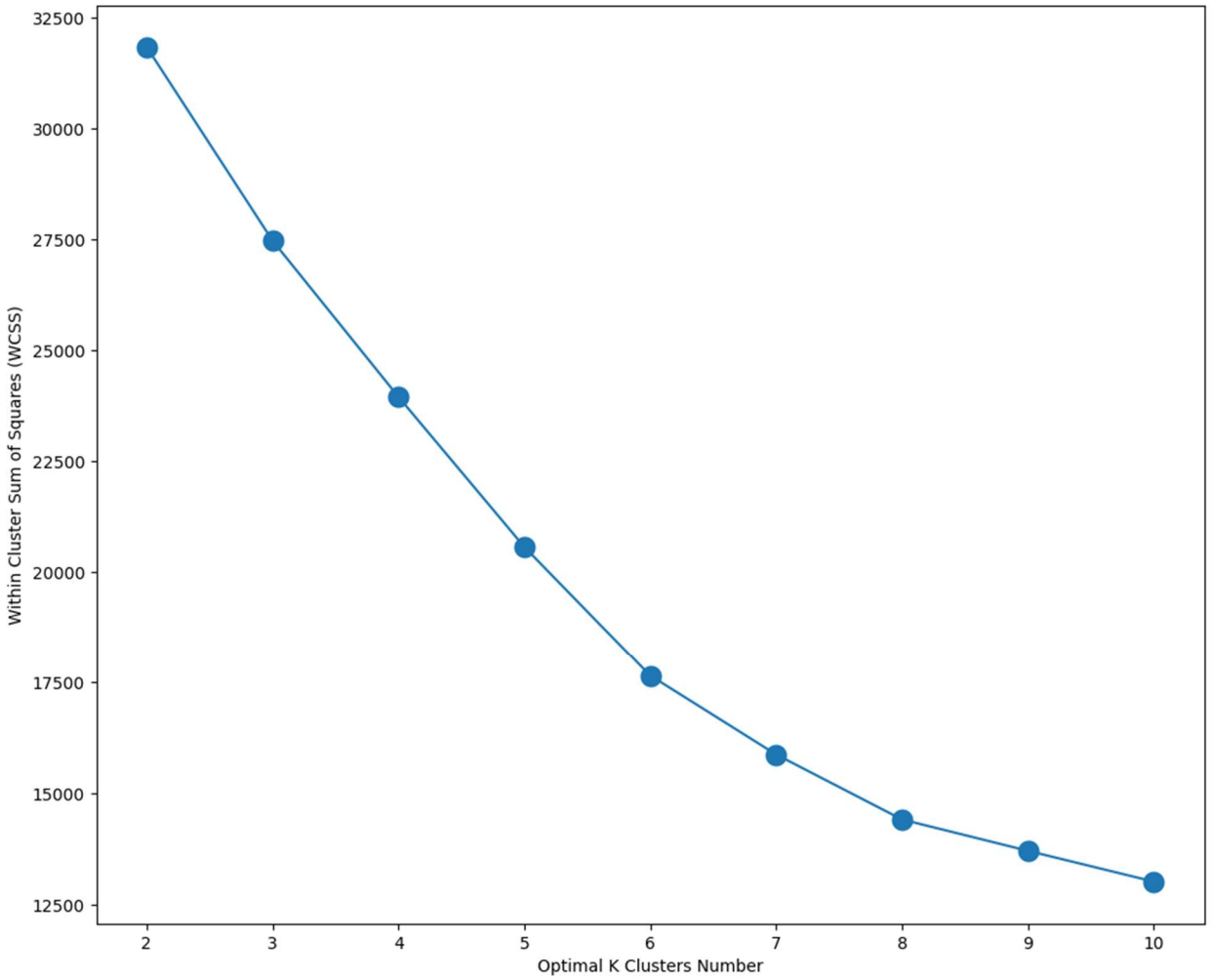
Please see the attached CSV titled: KMoikD212_WGUmedical.

**Part IV: Analysis**

**D1. Output and Intermediate Calculations**

Using the elbow method, I was able to determine that the optimal number of clusters in the data set is 3:

```
#find optimal k value
wcss=[]
for k in range (2, 11):
    model=KMeans(n_clusters=k, n_init=50, random_state=300)
    model.fit(scaled_df)
    wcss.append(model.inertia_)
wcss_s=pd.Series(wcss, index=range(2, 11))

plt.figure(figsize=(12, 10))
ax=sns.lineplot(y=wcss_s, x=wcss_s.index)
ax=sns.scatterplot(y=wcss_s, x=wcss_s.index, s=200)
ax=ax.set(xlabel="Optimal K Clusters Number",
          ylabel="Within Cluster Sum of Squares (WCSS)")
```



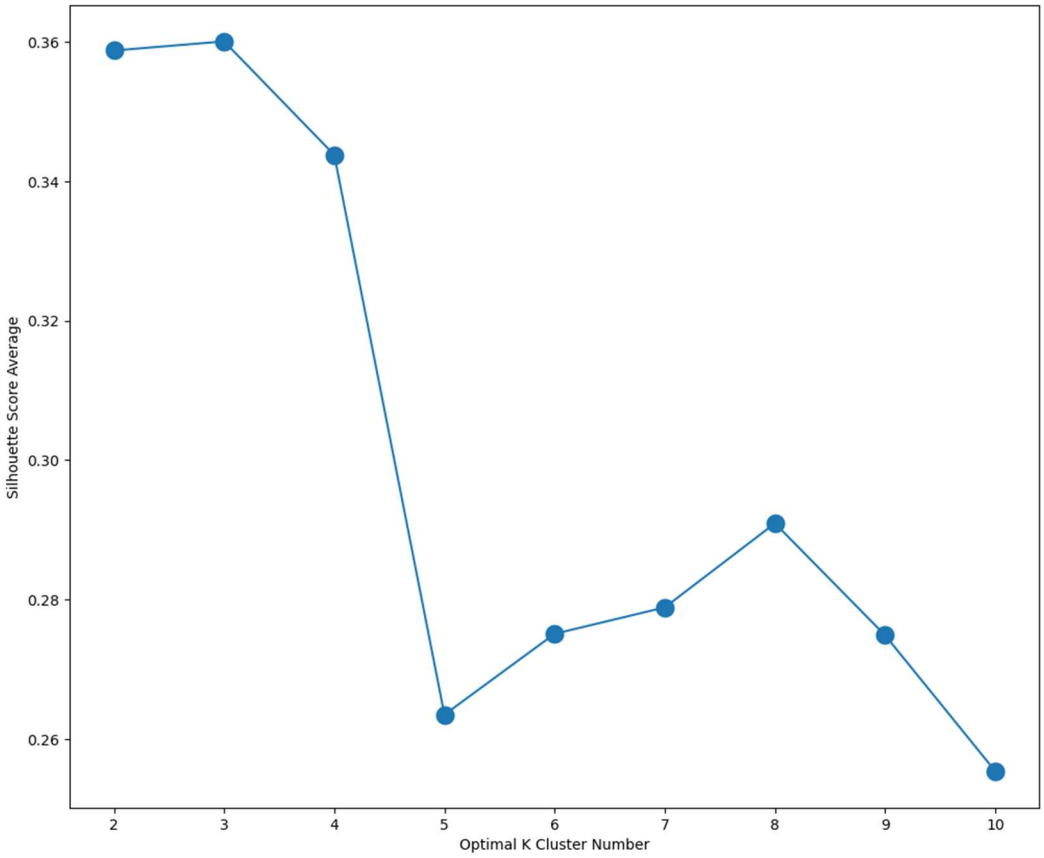

The downward trend becomes more gradual after k=3. I confirmed this when I obtained the silhouette score which found that the model fit the data best when using k=3:

```
silhouette = []

for k in range(2, 11):
    model = KMeans(n_clusters=k, n_init=25, random_state=300)
    model.fit(scaled_df)
    silhouette.append(silhouette_score(scaled_df, model.labels_))

silhouette_s=pd.Series(silhouette, index=range(2, 11))
```

```
silhouette_score=silhouette_score(scaled_df, k_model.labels_)
silhouette_score
```

```
0.36006601461710924
```

```
plt.figure(figsize=(12, 10))
ax=sns.lineplot(y=silhouette_s, x=silhouette_s.index)
ax=sns.scatterplot(y=silhouette_s, x=silhouette_s.index, s=200)
ax=ax.set(xlabel="Optimal K Cluster Number",
          ylabel="Silhouette Score Average")
```




**D2. Code Execution**

For my analysis, after the steps shown in section C3, which included creating the K-Means model (Western Governors University, n.d.):

```
#create k means object
k_model=KMeans(n_clusters=3, n_init=25, random_state=300)
k_model.fit(scaled_df)
```

```
KMeans(n_clusters=3, n_init=25, random_state=300)
```

I then counted the data points in each cluster (Western Governors University, n.d.):

```
evaluate=pd.Series(k_model.labels_).value_counts()
evaluate
```

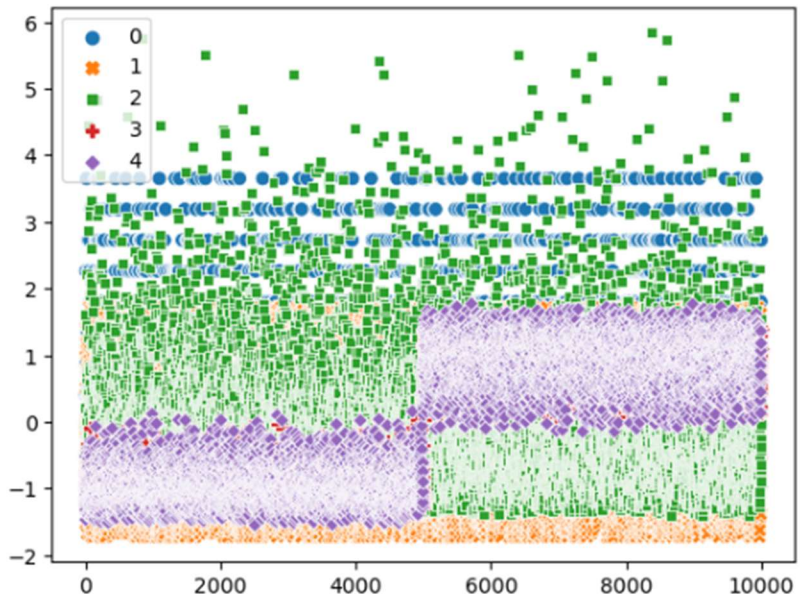
```
1    4605
2    4530
0     865
Name: count, dtype: int64
```

I visualized the 3 clusters that were created (Western Governors University, n.d.):

```
centeroid=pd.DataFrame(fin_model.cluster_centers_)
centeroid
```

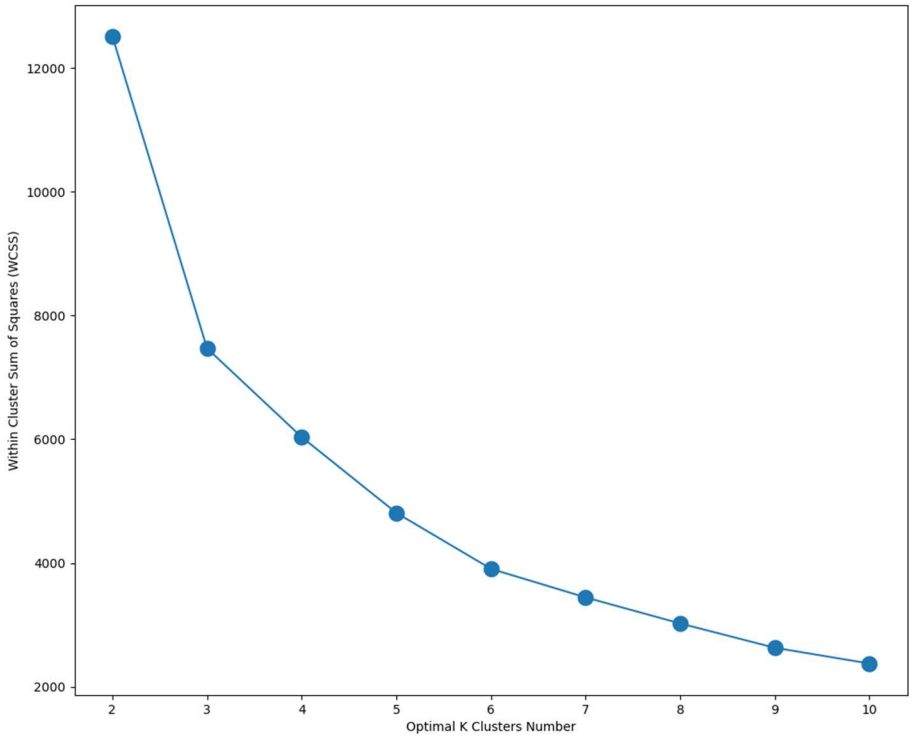
| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | 2.482247 | 0.041902 | 0.019120 | 0.085010 | 0.092037 |
| **1** | -0.238582 | -0.022679 | 0.007154 | -0.961621 | -0.950396 |
| **2** | -0.232052 | 0.015039 | -0.010926 | 0.961079 | 0.948326 |

```
#visualize
ax=sns.scatterplot(data=scaled_df, s=50)
```




I found the optimal value for k (Western Governors University, n.d.):

```
#find optimal k value
wcss=[]
for k in range (2, 11):
    model=KMeans(n_clusters=k, n_init=50, random_state=300)
    model.fit(scaled_df)
    wcss.append(model.inertia_)
wcss_s=pd.Series(wcss, index=range(2, 11))

plt.figure(figsize=(12, 10))
ax=sns.lineplot(y=wcss_s, x=wcss_s.index)
ax=sns.scatterplot(y=wcss_s, x=wcss_s.index, s=200)
ax=ax.set(xlabel="Optimal K Clusters Number",
          ylabel="Within Cluster Sum of Squares (WCSS)")
```




Obtained the silhouette score (Western Governors University, n.d.):

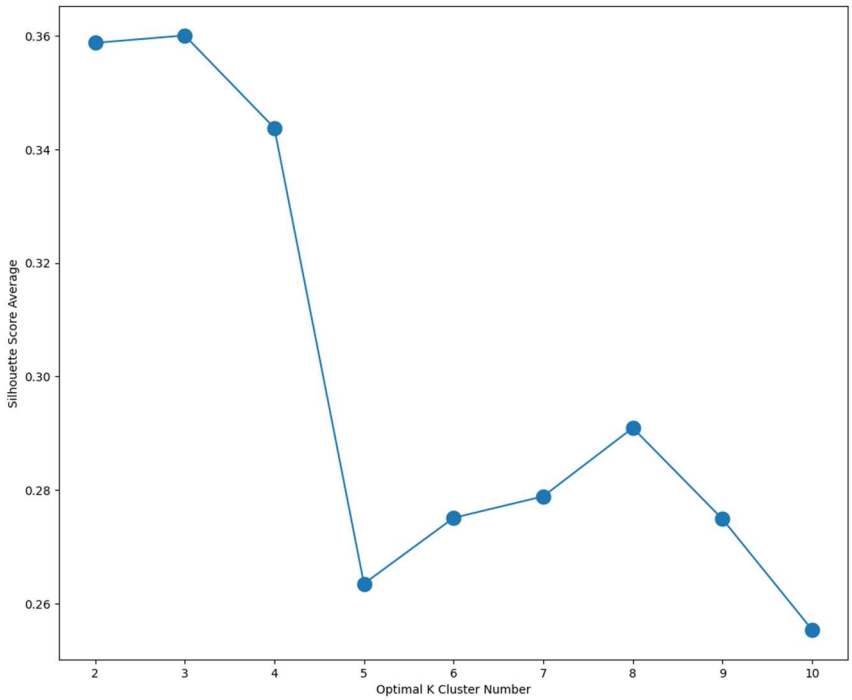
```
silhouette = []

for k in range(2, 11):
    model = KMeans(n_clusters=k, n_init=25, random_state=300)
    model.fit(scaled_df)
    silhouette.append(silhouette_score(scaled_df, model.labels_))

silhouette_s=pd.Series(silhouette, index=range(2, 11))
```

```
silhouette_score=silhouette_score(scaled_df, k_model.labels_)
silhouette_score
```

```
0.36006601461710924
```

Plotted the silhouette score with optimal K cluster number (Western Governors University, n.d.):

```
plt.figure(figsize=(12, 10))
ax=sns.lineplot(y=silhouette_s, x=silhouette_s.index)
ax=sns.scatterplot(y=silhouette_s, x=silhouette_s.index, s=200)
ax=ax.set(xlabel="Optimal K Cluster Number",
          ylabel="Silhouette Score Average")
```




Created my final model (Western Governors University, n.d.):

```
#final model
fin_model=KMeans(n_clusters=3, n_init=25, random_state=300)
fin_model.fit(scaled_df)
```

```
KMeans(n_clusters=3, n_init=25, random_state=300)
```

Visualized my final centroids (Western Governors University, n.d.):

```
centeroid=pd.DataFrame(fin_model.cluster_centers_)
centeroid
```

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | 2.482247 | 0.041902 | 0.019120 | 0.085010 | 0.092037 |
| **1** | -0.238582 | -0.022679 | 0.007154 | -0.961621 | -0.950396 |
| **2** | -0.232052 | 0.015039 | -0.010926 | 0.961079 | 0.948326 |

Labeled the clusters (Western Governors University, n.d.):

```
#label clusters
red_df["Cluster"]=fin_model.labels_.tolist()
red_df.head(12)
```

| | Children | Age | Income | Initial_days | TotalCharge | Cluster |
|---|---|---|---|---|---|---|
| **0** | 1 | 53 | 86575.93 | 10.585770 | 3726.702860 | 1 |
| **1** | 3 | 51 | 46805.99 | 15.129562 | 4193.190458 | 1 |
| **2** | 3 | 53 | 14370.14 | 4.772177 | 2434.234222 | 1 |
| **3** | 0 | 78 | 39741.49 | 1.714879 | 2127.830423 | 1 |
| **4** | 1 | 22 | 1209.56 | 1.254807 | 2113.073274 | 1 |
| **5** | 3 | 76 | 81999.88 | 5.957250 | 2636.691180 | 1 |
| **6** | 0 | 50 | 10456.05 | 9.058210 | 3694.627161 | 1 |
| **7** | 7 | 40 | 38319.29 | 14.228019 | 3021.499039 | 0 |
| **8** | 0 | 48 | 55586.48 | 6.180339 | 2968.402860 | 1 |
| **9** | 2 | 78 | 38965.22 | 1.632554 | 3147.855813 | 1 |
| **10** | 4 | 55 | 38503.82 | 2.595912 | 2837.861788 | 1 |
| **11** | 1 | 64 | 14126.30 | 7.075083 | 3166.627638 | 1 |

Finally, I aggregated and grouped my clusters (Western Governors University, n.d.):

```
Patients=pd.DataFrame(red_df)
```

```
Patients.agg({
    "Children" : "median",
    "Age" : "median",
    "Income" : "median",
    "Initial_days" : "median",
    "TotalCharge" : "median",}).round(2)
```

```
Children            1.00
Age                53.00
Income          33768.42
Initial_days       35.84
TotalCharge      5213.95
dtype: float64
```

```
Patients.groupby("Cluster").agg({
     "Children" : "median",
    "Age" : "median",
    "Income" : "median",
    "Initial_days" : "median",
    "TotalCharge" : "median",})
```

| | Children | Age | Income | Initial_days | TotalCharge |
|---|---|---|---|---|---|
| **Cluster** | | | | | |
| **0** | 7.0 | 55.0 | 33620.96 | 44.435430 | 6129.585000 |
| **1** | 1.0 | 53.0 | 33932.49 | 7.810055 | 3171.543626 |
| **2** | 1.0 | 53.0 | 33662.31 | 61.234355 | 7467.432500 |

**Part V: Data Summary and Implications**

**E1. Quality of the Clustering Technique**

The silhouette score of my model is 0.36. This score means that my model has fairly distinguished clusters, but the clusters are not clearly distinguished (Towards Data Science, 2020). Ultimately, while my model is a decent fit for the data, there is definite room for improvement to create a better fitting model.

**E2. Results and Implications**

The elbow method determined that the optimal number of k clusters was 3 with a silhouette score of 0.36. The visualization confirms that k=3 clusters had the highest silhouette score, meaning it was the best fit.

Looking at my clusters:

| Cluster | Children | Age | Income | Initial_days | TotalCharge |
|---|---|---|---|---|---|
| 0 | 7.0 | 55.0 | 33620.96 | 44.435430 | 6129.585000 |
| 1 | 1.0 | 53.0 | 33932.49 | 7.810055 | 3171.543626 |
| 2 | 1.0 | 53.0 | 33662.31 | 61.234355 | 7467.432500 |

All 3 Clusters have patients with similar ages, number of children, and income, except Cluster 0 has a much higher median number of children at 7. Cluster 1 has far fewer initial days and thus initial charges, which logically makes sense, whereas Cluster 0 and 2 both have a higher number of initial days and thus a higher amount of charges.

It is interesting that patients in Cluster 1 have far less initial days, just over 1 week compared to over 44 days and over 61 days for Clusters 0 and 2. It would be ideal to obtain more information on that patients in Cluster 1 to find out the cause of such few initial days.

**E3. Limitation**

One limitation of my analysis is that I reduced the dataset to the numeric variables I thought were most important. More variables including more patient data and additional data for patients who were readmitted, would most likely be helpful in creating a better fitting model.

**E4. Course of Action**

While my model fits the data fairly well, it could be improved with more data and the addition of different variables to see how the model changes. While understanding the patients the hospital is treating is the first step in using data to make better business decisions, additional information is needed to fully find and understand which variables affect important factors such as readmission.

**Part VI: Demonstration**

**F. Panopto Vide of Code and Programs**

The link for my Panopto Video is:
https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8e0067ab-3add-4704-837c-b12e01219b84

**G. Sources for Third-Party Code**

Western Governors University. (n.d.). *Constructing and running the K-means Model_default*. WGU. [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8048d616-4d5f-4625-accd-b0ee01873eba

Western Governors University. (n.d.). *Evaluating and visualizing the model_default*. WGU. [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=9fa8783e-d7d2-4b4d-b06e-b0ee01874bea

Western Governors University. (n.d.). *Analyze and interpret K-means results_default*. WGU. [Video]. https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=3fe13831-fe4b-4c6b-a3eb-b0ee018754bc

**H. Sources**

Towards Data Science. (2018). *Understanding K-means Clustering in Machine Learning*. Towards Data Science. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

Towards Data Science. (2020). *Silhouette Coefficient*. Towards Data Science. https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

**I. Professional Communication**