D212 Task I PA

Shawn Wheeler

000364855

Table of Contents

[A1: Proposal of Question 3](#_Toc169306124)

[A2: Defined Goal 3](#_Toc169306125)

[B1: Explanation of the Clustering Technique 3](#_Toc169306126)

[B2: Summary of the Technique Assumption 3](#_Toc169306127)

[B3: Packages or Libraries List 3](#_Toc169306128)

[C1: Data Preprocessing 4](#_Toc169306129)

[C2: Data Set Variables 4](#_Toc169306130)

[C3: Steps for Analysis 5](#_Toc169306131)

[C4: Cleaned Data Set 7](#_Toc169306132)

[D1: Output and Intermediate Calculations 7](#_Toc169306133)

[D2: Code Execution 10](#_Toc169306134)

[E1: Quality of the Clustering Technique 12](#_Toc169306135)

[E2: Results and Implications 13](#_Toc169306136)

[E3: Limitations 13](#_Toc169306137)

[E4: Course of Action 13](#_Toc169306138)

[F: Panopto Video of Code 13](#_Toc169306139)

[F1: Panopto Video of Programs 14](#_Toc169306140)

[G: Sources for Third-Party Code 14](#_Toc169306141)

[H: Sources 14](#_Toc169306142)

## Part I: Research Question

### A1: Proposal of Question

Using k-means clustering, can demographic and additional information for patients be analyzed in order to determine better methods of providing service?

### A2: Defined Goal

The overall goal is this analysis is to utilize k-means clustering on the variables for initial days and additional charges. K-means clustering will allow for segmentation of the individuals into distinct groups based on their demographic and other information within the medical data set. These clusters can then be analyzed and considered for future alteration to services or methodologies as they can be approached uniquely.

## Part II: Technique Justification

### B1: Explanation of the Clustering Technique

K-means clustering is an unsupervised machine learning technique which is used to identify clusters, or groups, of data points that are most similar. The method used randomly creates centroids and considers the distance between each centroid and each data point at that position. After rearranging the centroids and groupings of data points a number of times, ideally an optimal clustering of the data will be discovered. The amount of created clusters, which is the same as the number of centroids, is predetermined. (Liberman, 2020)

In this project, k-means clustering will initially be applied to the Initial\_days and Additional\_charges variables with a k-value of 3, meaning 3 clusters will be identified. This data will then be used to determine if 3 clusters is the optimal amount or if there is a k-value that better represents that data. A cursory look at a comparison of Initial\_days and Additional\_charges leads me to expect that 3 may not be the optimal k-value due to how uniformly the data is distributed.

### B2: Summary of the Technique Assumption

An assumption of k-means clustering is that, due to the methodology of the algorithm, all clusters will be very similar in size to one another. Generally, three clusters will each reflect approximately one-third of the data, for instance. (GeeksForGeeks, 2024)

### B3: Packages or Libraries List

*Table of Libraries and Packages:*

|  |  |
| --- | --- |
| **Libraries / Packages** | **Usage** |
| numpy | Performing mathematic operations |
| pandas | Data manipulation |
| Import Series | Manipulating one-dimensional data |
| Import DataFrame | Manipulating two-dimensional data |
| seaborn | Creating visualizations |
| matplotlib.pyplot | Creating visualizations |
| sklearn.preprocessing |  |
| import StandardScaler | Normalizing data before k-means clustering |
| sklearn.cluster |  |
| import KMeans | Instantiating k-means clustering algorithm for use |
| sklearn.metrics |  |
| import silhouette\_score | Calculating average silhouette score for determining optimal k-value |
| warnings | Mitigating any potential Panopto issues |

## Part III: Data Preparation

### C1: Data Preprocessing

Outliers can be an issue for performing k-means clustering. In order to reduce the effect of outliers, data should be standardized during preprocessing. Standardization of data, in this case, refers to scaling the values of data to all fall between -1 and 1. This reduction in scale maintains an accurate distance of the data points required by the clustering algorithm while reducing the magnitude of the values. Therefore, the impact of outliers is lessened and the algorithm will perform better. (Sharma, 2019)

### C2: Data Set Variables

*Table of Variables:*

|  |  |
| --- | --- |
| **Variable** | **Data Type** |
| Children | Continuous |
| Age | Continuous |
| Income | Continuous |
| VitD\_levels | Continuous |
| Doc\_visits | Continuous |
| Full\_meals\_eaten | Continuous |
| VitD\_supp | Continuous |
| Initial\_days | Continuous |
| TotalCharge | Continuous |
| Additional\_charges | Continuous |

### C3: Steps for Analysis

1. **Import packages and libraries**

import numpy as np

import pandas as pd

from pandas import Series, DataFrame

import seaborn as seaborn

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

import warnings

warnings.filterwarnings("ignore")

%matplotlib inline

1. **Load data set for exploration and manipulation**

df = pd.read\_csv("C:/Users/Owner/medical\_clean212.csv")

1. **Data exploration**

df.head()

df.info()

1. **Histograms of all quantitative columns**

df.hist(grid=False, figsize=(16,12), layout=[5,5])

1. **Check for missing data, duplicates, outliers, and other potential issues**

# Checking for missing data

df.isnull().sum()

# Checking for duplicates

df.duplicated().value\_counts()

# Checking for outliers in quantitative variables

quant\_columns = ['Population', 'Children', 'Age', 'Income', 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp', 'Initial\_days', 'TotalCharge', 'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8']

for column in df:

if column in quant\_columns:

plt.figure()

plt.gca().set\_title(column)

df.boxplot([column])

1. **Drop columns that will not be used for analysis and verify the remainders**

# Drop unneeded columns

df = df.drop(['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Marital', 'Gender', 'ReAdmis', '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'], axis=1)

# Verify columns

df.info()

1. **Rename column to fit data dictionary**

# Rename column to fit data dictionary

df = df.rename(columns={'vitD\_supp': 'VitD\_supp'})

1. **Remove outliers and replace with median values, round columns, and recast data types**

# Children

df['Children'] = np.where(df['Children'] > 5, np.nan, df['Children'])

df['Children'].fillna(df['Children'].median(), inplace=True)

# Recast as int

df['Children'] = df['Children'].astype(int)

# Income

df['Income'] = np.where(df['Income'] > 97533, np.nan, df['Income'])

df['Income'].fillna(df['Income'].median(), inplace=True)

# Round

df['Income'] = df.Income.round(2)

# VitD\_levels

df['VitD\_levels'] = np.where(df['VitD\_levels'] >= 22, np.nan, df['VitD\_levels'])

df['VitD\_levels'].fillna(df['VitD\_levels'].median(), inplace=True)

# Round

df['VitD\_levels'] = df.VitD\_levels.round(2)

# Full\_meals\_eaten

df['Full\_meals\_eaten'] = np.where(df['Full\_meals\_eaten'] > 3, np.nan, df['Full\_meals\_eaten'])

df['Full\_meals\_eaten'].fillna(df['Full\_meals\_eaten'].median(), inplace=True)

# Recast as int

df['Full\_meals\_eaten'] = df['Full\_meals\_eaten'].astype(int)

# Additional\_charges

df['Additional\_charges'] = np.where(df['Additional\_charges'] > 27928, np.nan, df['Additional\_charges'])

df['Additional\_charges'].fillna(df['Additional\_charges'].median(), inplace=True)

# Round

df['Additional\_charges'] = df.Additional\_charges.round(2)

# Round Initial\_days

df['Initial\_days'] = df.Initial\_days.round(1)

# Round TotalCharge

df['TotalCharge'] = df.TotalCharge.round(2)

1. **Visually analyze all columns compared to Initial\_days via bivariate analysis**

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Children')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Age')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Children')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Income')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'VitD\_levels')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Doc\_visits')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Full\_meals\_eaten')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'VitD\_supp')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'TotalCharge')

ax = seaborn.scatterplot(data = df, x = 'Initial\_days', y = 'Additional\_charges')

1. **Standardize data using StandardScaler**

scaler = StandardScaler()

prep\_df = scaler.fit\_transform(df)

prep\_df = pd.DataFrame(prep\_df, columns = df.columns)

prep\_df

1. **Save .csv of now prepared data**

prep\_df.to\_csv('d212task1clean.csv', index=False)

### C4: Cleaned Data Set

*See Attached .csv:* d212task1clean.csv

## Part IV: Analysis

### D1: Output and Intermediate Calculations

k-means clustering requires a k-value to be selected first. Therefore, an initial model was created using a k-value of 3. After instantiating the model and letting it run, the model was evaluated. The number of data points per cluster, the points for the centeroids themselves, and a scatterplot organized to easily view the clusters were created and viewed.

*Values of clusters with k=3:*

*A white background with black text

Description automatically generated*

*Values of centeroids with k=3:*

*A screenshot of a number

Description automatically generated*

*Scatterplot of k=3 model with centroids:*

*A diagram of a number of dots

Description automatically generated with medium confidence*

Now that the initial model had been created, two methods were used to attempt to determine the optimal value for k. First, Within Cluster Sum of Squares (WCSS) was calculated for a range of k-values up to 12 for k-means clustering. This allowed for the creation of the graph below to visually identify the optimal k-value. The elbow method was used for that determination which is where you find the point where the bend noticeably enhances as with a human elbow bending, in this case, that value is 4.

*Within cluster sum of squares graph:*

*A graph with a blue line

Description automatically generated*

Secondly, the silhouette score was calculated for the initial model. The silhouette score is a scale from -1 to 1, with 1 being ideal, that is also used to determine quality of the models tested. It is a representation the distances between clusters and the tightness of the cluster grouping themselves. The value shown is the average of the silhouette scores of the data points.

*Silhouette score of initial model:*

**

Then the average silhouette score of a range of k-values up to 12 were calculated and graphed. That graph shows the highest average silhouette score at k=4.

*Silhouette score graph:*

*A graph with a blue line

Description automatically generated*

Both tests have shown that the optimal k-value is 4. With that knowledge, the same process as the initial mode was duplicated except with a k-value of 4 instead of 3. The number of data points in each cluster, the points of the centeroids themselves, and a visual representation of the clusters within the graph are shown below. Furthermore, the silhouette score is shown below which is a value closer to 1 than the initial model.

*Values of clusters with k=4:*

*A screenshot of a computer

Description automatically generated*

*Values of centeroids for k=4:*

*A screenshot of a calculator

Description automatically generated*

*Scatterplot graph of clusters with k=4:*

*A screenshot of a graph

Description automatically generated*

*Silhouette score with k=4:*

**

### D2: Code Execution

*See attached code:* d212Task1.ipynb

*See code below:*

# KMeans (Course material, n.d.)

# Select columns to compare

prep\_df = prep\_df[['Initial\_days', 'Additional\_charges']]

# Verify dataframe

prep\_df.info()

# Create and fit KMeans model

first\_model = KMeans(n\_clusters = 3, n\_init = 40, random\_state = 49)

first\_model.fit(prep\_df)

# Evaluate model

pd.Series(first\_model.labels\_).value\_counts()

ax = seaborn.scatterplot(data=prep\_df, x = 'Additional\_charges', y = 'Initial\_days')

# View centroids

centeroid = pd.DataFrame(first\_model.cluster\_centers\_, columns = prep\_df.columns)

centeroid

# Visualize centroids and data

plt.title('Clusters for k=3')

ax = seaborn.scatterplot(data = prep\_df, x = 'Initial\_days', y = 'Additional\_charges', hue = first\_model.labels\_, palette = 'muted', legend = True)

ax = seaborn.scatterplot(data = centeroid, x = 'Initial\_days', y = 'Additional\_charges', hue = centeroid.index, palette = 'muted', s = 1000, legend = False)

for i in range(len(centeroid)):

plt.text(x = centeroid.Initial\_days[i], y = centeroid.Additional\_charges[i], s = i, horizontalalignment = 'center', verticalalignment = 'center', size = 13, color = 'black')

# Within Cluster Sum of Squares

wcss = []

for k in range (2, 12):

wcss\_model = KMeans(n\_clusters = k, random\_state = 251)

wcss\_model.fit(prep\_df)

wcss.append(wcss\_model.inertia\_)

wcss\_series = pd.Series(wcss, index = range(2, 12))

ax = seaborn.lineplot(y = wcss\_series, x = wcss\_series.index)

ax = seaborn.scatterplot(y = wcss\_series, x = wcss\_series.index)

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

plt.ylabel('Within Cluster Sum of Square (WCSS)')

plt.title('WCSS Graph for Optimal k')

# Silhouette Score (-1 - +1, where 1 is best)

sil\_score = silhouette\_score(prep\_df, first\_model.labels\_)

print('Average Silhouette Score: ', sil\_score)

# Average sil score and optimal clusters visual

sil\_obj = []

for k in range(2, 12):

sil\_model = KMeans(n\_clusters = k, random\_state = 173)

sil\_model.fit(prep\_df)

sil\_obj.append(silhouette\_score(prep\_df, sil\_model.labels\_))

sil\_series = pd.Series(sil\_obj, index = range(2, 12))

ax = seaborn.lineplot(y = sil\_series, x = sil\_series.index)

ax = seaborn.scatterplot(y = sil\_series, x = sil\_series.index)

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

plt.ylabel('Average Silhouette Score')

plt.title('Silhouette Score Graph for Optimal k')

# Analysis with optimal k (k=4)

optimal = KMeans(n\_clusters = 4, random\_state = 133)

optimal.fit(prep\_df)

# Eval

pd.Series(optimal.labels\_).value\_counts()

# Centeroids for graph

centeroid = pd.DataFrame(optimal.cluster\_centers\_, columns = prep\_df.columns)

centeroid

# Cluster graph for optimal k

plt.title('Clusters for k=4')

ax = seaborn.scatterplot(data = prep\_df, x = 'Initial\_days', y = 'Additional\_charges', hue = optimal.labels\_, palette = 'muted', legend = True)

ax = seaborn.scatterplot(data = centeroid, x = 'Initial\_days', y = 'Additional\_charges', hue = centeroid.index, palette = 'muted', s = 1000, legend = False)

for i in range(len(centeroid)):

plt.text(x = centeroid.Initial\_days[i], y = centeroid.Additional\_charges[i], s = i, horizontalalignment = 'center', verticalalignment = 'center', size = 13, color = 'black')

# Silhouette score for optimal model

sil\_score = silhouette\_score(prep\_df, optimal.labels\_)

print('Average Silhouette Score: ', sil\_score)

## Part V: Data Summary and Implications

### E1: Quality of the Clustering Technique

In order to determine the quality of the k-value chosen, the silhouette score was calculated and compared. The silhouette score ranges from -1 to 1 with 1 being ideal. When the silhouette value of a data point is closer to 1, it is part of a more compact cluster which is simultaneously further away from other clusters which is ideal. The average silhouette score for a k-value then represents the overall quality of that k-value for clustering purposes. For the initial model (k=3) the average silhouette score was 0.5477. With a k-value of 4, the optimal value, the average silhouette score was shown to be 0.5526 which is closer to 1 than the initial model. Therefore, the model with k=4 has a higher quality than k=3. Looking at the graph in D1 that was created for average silhouette scores for each set of clusters up to 12, we see that k=4 had the highest point on the line graph which again indicates that it is the value with the highest quality.

### E2: Results and Implications

After creating the initial model, the quality of k=3 was displayed by the average silhouette score of 0.5477. By plotting the Within Sum of Squares (WCSS) as shown in D1, the elbow method could be used to determine the optimal number for k. Based on that graph, the optimal value was k=4. After that, the average silhouette scores were calculated for a range of k-values which also showed that the closest value to 1 was at k=4. The average silhouette score for k=4 was shown to be 0.5526, a superior value than k=3.

Once k-means clustering was applied to the data set for k=4, the four centroids were plotted at (-0.96, -0.45), (0.96, -0.43), (0.95, 1.59), and (-0.96, 1.58). There were 3905, 3925, 1077, and 1093 data points in each group respectively. For the first group, centroid (-0.96, -.045), the data would represent patients with both low initial days in their stay and low additional costs. The second group, (0.96, -0.43), seems to be patients with longer initial stays but low additional charges. For the third centroid, (0.95, 1.59), which would seem to be a group that consists for a low initial stay yet high additional charges. Finally, the fourth centroid, (-0.96, 1.58) contains patients with both high initial days of their hospital visit and high additional charges.

### E3: Limitations

One limitation encountered in this project was simply that only one set of variables was compared. It can be quite difficult to draw conclusions with only those results. Since the overall goal theoretically is to lower readmission rates, it would have been preferable to be able to compare both more variables and also variables that are perhaps more closely correlated to readmission rates.

### E4: Course of Action

With the goal from A1 being to gain additional insight into potential improvements to service, the results in E2 gave clustering information for groups in two variables, Initial\_days and Additional\_charges. Both of these variables have been shown in the past to have some correlation with readmission rates which are to be mitigated as much as possible as per the directive. These groups should be further examined to see if there is any useful insight to be gleaned, more data points should be collected in order to allow for analysis more closely related to readmission rates themselves, and more samples should be collected. Since additional charges is a broad term, for instance, it could be broken down into subcategories of additional charges for which those could be compared to initial days for analysis.

## Part VI: Demonstration

### F: Panopto Video of Code

*See Panopto Link:* https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=fc76f806-37b8-43df-9dfd-b192014376d5

### F1: Panopto Video of Programs

*See Panopto Link:* https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=fc76f806-37b8-43df-9dfd-b192014376d5

### G: Sources for Third-Party Code

Course Material (n.d.)

### H: Sources

GeeksforGeeks. (2024). *Demonstration of K-Means Assumptions*. (2023, December 9). GeeksforGeeks. https://www.geeksforgeeks.org/demonstration-of-k-means-assumptions/

Liberman, N. (2020, May 21). *K Means Clustering Explained Easily*. Medium. https://medium.com/@neil.liberman/k-means-clustering-e00408493a40

Sharma, P. (2019, August 25). *Why is scaling required in KNN and K-Means?* Medium. https://medium.com/analytics-vidhya/why-is-scaling-required-in-knn-and-k-means-8129e4d88ed7