**PERFORMANCE ASSESSMENT TASK 1: Clustering Techniques**

**D212 – Data Mining II**

Michael Lawson

Western Governors University

D212 – Data Mining II

PA Task 1: Clustering Techniques

Kesselly Kamara

09/12/2022

Michael C. Lawson | Student ID: #001479557 | Program: MS Data Analytics (5/1/21) |

Program Mentor: Emil Stoica | 859-265-1888 (EST) | [mlaws77@wgu.edu](mailto:mlaws77@wgu.edu)

**Part I:**

A.  Purpose of data mining report:

1.  O**ne** real-world question.

Will K-Means reveal groups in the ‘Initial\_days’ and ‘TotalCharge’ columns?

2.  **One** goal of the analysis.

The goal of the data analysis is to use K-means to cluster data in the ‘Initial\_days’ and ‘TotalCharge’ columns to identify hidden groups in the data.

**Part II:**

B.  Reasons for K-means:

1.  How K-means analyzes the dataset and expected outcomes.

K-means is a centroid clustering method that requires a pre-set number of clusters. K-means is the most common clustering covered in machine-learning for beginners (Das, 2020). The goal of K-means is to find groups within the data, which happens to line up with the goal set for the analysis. Sometimes the limitation of K-means is that the number of groups must be set around a chosen number of centroids before the model is run.

The elbow method will be used to plot the sum of squared errors and determine the best value for k.

The expected outcome is that K-Means will be able to correctly group the test data into the number of groups determined by the elbow method.

2.  **One** assumption of K-means.

K-means clustering assumes that data points have something in common, and that the common thing is something that predicts which centroid the data is closest to. Patients in the medical data set that share similar attributes will also fall into similar categories.

3.  Python packages / libraries:

Libraries used

1. Pandas – fast and powerful data analysis and manipulation library.

2. NumPy – wide range of math functions.

3. matplotlib

a. pyplot – visualizations of data.

b. inline – included so visualizations are included in Jupyter Notebook next to code.

4. Warnings.filterwarnings - loaded to remove filter warnings

5. scipy – algorithms for optimization and stats.

6. seaborn – more visualization tools

7. from yellowbrick.cluster import SilhouetteVisualizer – used to create Silhouette Plot

8. sklearn – efficient tools for predictive analysis.

a. cluster

KMeans – used to create K-means model.

b. preprocessing

i. LabelEncoder – used to convert categorical variables to numerical.

ii. MinMaxScaler – used to scale columns to help create a better model.

**Part III:**

C.

1.  **One** data preprocessing goal.

Preprocessing for k-means will include plotting the sum of squared errors to determine the best value for k using the elbow method. The point where the plot has the most angle at the “elbow” of the curve helps determine the best value for k.

2.  Identify initial variables & label as continuous or categorical.

|  |  |  |
| --- | --- | --- |
| Initial\_days |  | Continuous |
| TotalCharge |  | Continuous |
|  |  |  |
|  |  |  |
|  |  |  |

**3.  Preparation Steps and Code**

*# Import packages and libraries*

**import** pandas **as** pd

**import** numpy **as** np

**from** sklearn.cluster **import** KMeans

**from** sklearn.preprocessing **import** LabelEncoder

**from** sklearn.preprocessing **import** MinMaxScaler

**import** seaborn **as** sns

**import** scipy.stats **as** scs

**import** matplotlib.pyplot **as** plt

**%matplotlib** inline

**from** yellowbrick.cluster **import** SilhouetteVisualizer

**import** warnings

warnings**.**filterwarnings('ignore')

*#Import data*

med\_file **=** r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D212\medical\_clean.csv"

med\_load **=** pd**.**read\_csv(med\_file)

df **=** med\_load[['Initial\_days','TotalCharge']]

# Scatterplot of ‘Initial\_days’ vs ‘TotalCharge’

plt**.**scatter(df['Initial\_days'],df['TotalCharge'])

*Chart

Description automatically generated*

# Calculate Sum of Squared Errors

k\_rng **=** range(1,10)

sse**=**[]

**for** k **in** k\_rng:

km**=**KMeans(n\_clusters**=**k)

km**.**fit(df[['Initial\_days','TotalCharge']])

sse**.**append(km**.**inertia\_)

sse

A picture containing text, person

Description automatically generated

# Plot the Sum of Squared Errors to find the optimal value for k

plt**.**xlabel('k')

plt**.**ylabel('Sum of Squared Error')

plt**.**plot(k\_rng, sse)

*Chart, line chart

Description automatically generated*

**4.  Copy of cleaned data sets.**

# Export prepared train and test data sets

df**.**to\_csv(r"C:\Users\mlaws\OneDrive - Western Governors University\Documents\WGU\D212\medical\_kmeans.csv")

**Part IV: Analysis**

D.

1.  Describe the analysis technique.

K-means clustering is a machine learning algorithm (unsupervised) used to divide a dataset into k clusters, where k is the number of clusters that must be predetermined by the analyst (datacamp). In this analysis, data points are clustered around two centroids (k=2). The Sum of Squared Errors was plotted to use the Elbow Method to determine that the best clusters will be formed if the value of k is set at 2. The k-means algorithm calculates the distance between data points and the two centroids, and clusters the data points according to the centroid the data point is closest to. The algorithm will continue updating the clusters as different variables are considered, until the model no longer changes, and the clusters cannot be updated any more.

2. Code.

# The “elbow” reveals that the best value for k is 2

# Run the K-means algorithm for 2 clusters

km **=** KMeans(n\_clusters**=**2)

km

Text

Description automatically generated with medium confidence

y\_pred **=** km**.**fit\_predict(df[['Initial\_days','TotalCharge']])

y\_pred



# Add a column for the cluster values to use for a scatter plot

df['Cluster'] **=** y\_pred

# Check that new “Cluster” column is in data frame

df**.**head()

Text

Description automatically generated with medium confidence

# Create a scatter plot of the clusters formed by K-Means

df1 **=** df[df**.**Cluster**==**0]

df2 **=** df[df**.**Cluster**==**1]

plt**.**scatter(df1**.**Initial\_days,df1['TotalCharge'], color **=** 'orange')

plt**.**scatter(df2**.**Initial\_days,df2['TotalCharge'], color **=** 'lightblue')

plt**.**xlabel('Initial\_days')

plt**.**ylabel('TotalCharge')

plt**.**legend(['Total Charge', 'Total Charge'])

*Chart, line chart

Description automatically generated*

# Create a Silhouette Score Plot to further verify the correctness of the K-Means clusters.

fig, ax **=** plt**.**subplots(2, 2, figsize**=**(15,8))

**for** i **in** [2, 3, 4, 5]:

'''

Create KMeans instance for different number of clusters

'''

km **=** KMeans(n\_clusters**=**i, init**=**'k-means++', n\_init**=**10, max\_iter**=**100, random\_state**=**42)

q, mod **=** divmod(i, 2)

'''

Create SilhouetteVisualizer instance with KMeans instance

Fit the visualizer

'''

visualizer **=** SilhouetteVisualizer(km, colors**=**'yellowbrick', ax**=**ax[q**-**1][mod])

visualizer**.**fit(df)

Chart

Description automatically generated

**Part V: Data Summary and Implications**

E.

1.  Accuracy of clustering technique.

The scatterplot of the clusters formed by the K-Means algorithm reveals that the two clusters formed clean and separate using the features ‘Initial\_days’ and ‘TotalCharge’. The Silhouette Score Plot shows that two was the most correct value for K because the plots are close in size and stronger than the average silhouette score (the dotted red line). You can see that separating into 3 clusters results in clusters that are not similar in size. The scatterplot reveals that K-Means correctly clustered the data points into separate groups at a high percentage, but not perfect since the two groups do merge slightly and are not separated completely or far apart.

*Chart, line chart

Description automatically generated*Chart

Description automatically generated

2.  Discuss the results.

The elbow rule determined that two is the most correct number of clusters for ‘Initial\_days’ and ‘TotalCharge’. K-Means correctly separated the data into two clusters as can be seen in the scatterplot. The data points follow a linear pattern, and while the clusters are not mixed, they do meet very close in the middle of the scatterplot. A better data set would have created further separation between the clusters around the centroids.

3.  **One** limitation.

The K-means clustering algorithm can only use continuous data, and the medical dataset is mostly categorical. While tools like label encoding can help integrate categorical values into the clustering method, a data set consisting mostly of Boolean variables are not useful for performing K-Means clustering.

4.  Course of action.

The recommendation is more data or completely new data is needed for the ‘Initial\_days’ and ‘TotalCharge’ columns. Data needs to be collected and added, or this data needs to be thrown out and a new pair of features examined. The K-Means algorithm was able to distinctly cluster the data into two groups, but the clusters are not formed tight around the centroids and the clusters meet in the center of the scatterplot.

**Part VI: Demonstration**

F.

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=6fd8cfbf-29fe-4c64-b507-af0f0019678f>

**References**

Das, Vikash Kumar. 11 October, 2020. *K-MEANS CLUSTERING VS HIERARCHAL CLUSTERING.* <https://www.globaltechcouncil.org/clustering/k-means-clustering-vs-hierarchical-clustering/>

datacamp. *K-Means Clustering in Python with scikit-learn.* [*https://www.datacamp.com/tutorial/k-means-clustering-python*](https://www.datacamp.com/tutorial/k-means-clustering-python)*.*

Kumar, Ajitesh. *KMeans Silhouette Score Explained With Python Example.* <https://dzone.com/articles/kmeans-silhouette-score-explained-with-python-exam>