**Nefisa Hassen**

**D 212 PA-Task 1**

**Part I: Research Question**

A. Describe the purpose of your data mining report by doing the following:

1. Propose **one** question relevant to a real-world organizational situation that you will answer using **one** of the following clustering techniques:

Can K-means clustering be used to identify patterns and group similar locations based on the additional charges incurred by patients?

2. Define **one** goal of the data analysis. Ensure your goal is reasonable within the scope of the selected scenario and is represented in the available data.

In this analysis, I will explore how to cluster the additional charges patients pay using their locations. The aim of this project is to group locations based on the similarity in the additional charges paid by patients.

**Part II: Technique Justification**

B. Explain the reasons for your chosen clustering technique from part A1 by doing the following:

1. Explain how the clustering technique you chose analyzes the selected data set. Include expected outcomes.

I chose K-means clustering to analyze the selected data set because all the variables are continuous, making K-means clustering the most appropriate technique. K-means clustering works by partitioning the data into distinct groups (clusters) based on similarities in the data points. The algorithm assigns data points to clusters by minimizing the variance within each cluster. The expected outcomes include the identification of distinct clusters of data points, which in this case will help to group the additional charges patients pay by their locations. This will enable us to identify patterns and make informed decisions based on the clustering results.

2. Summarize **one** assumption of the clustering technique.

K-means clustering assumes that clusters are spherical and isotropic, meaning their radii are roughly equal in all directions. The cluster center is the mean calculated by the algorithm based on the average of the data points in the cluster. This assumption makes K-means less effective for non-spherical or elongated clusters.

3. List the packages or libraries you have chosen for Python or R, and justify how *each* item on the list supports the analysis.

* **Pandas :**Provides data structures and data analysis tools for handling structured data.
* **NumPy :**Provides support for large multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
* **SciPy Stats :** Provides functions for statistical analysis. The zscore function specifically computes the z-score of each value in an array, indicating how many standard deviations an element is from the mean.
* **Seaborn** It provides a high-level interface for drawing attractive and informative statistical graphics.
* **preprocessing: Provides** tools for data preprocessing and normalization.
* **KMeans** Provides tools for clustering analysis.

**Part III: Data Preparation**

C. Perform data preparation for the chosen data set by doing the following:

1. Describe **one** data preprocessing goal relevant to the clustering technique from part A1.

One of the important goals of data preprocessing for clustering is identifying and treating outliers. In this analysis, outliers are treated using the Z-score method.

2. Identify the initial data set variables you will use to perform the analysis for the clustering question from part A1, and label *each* as continuous or categorical.

For this analysis i will be using three variables which are all continues variables addtional\_charges, Lat, and Lng

3. Explain *each* of the steps used to prepare the data for the analysis. Identify the code segment for *each* step.

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

2

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

3

**from** scipy **import** stats

In [19]:

1

med\_data **=** pd.read\_csv('medical\_clean.csv')

In [20]:

1

2

med\_data.columns *#looking at the columns*

Out[20]:

Index(['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'City', 'State',  
 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job',  
 'Children', 'Age', 'Income', 'Marital', 'Gender', 'ReAdmis',  
 'VitD\_levels', 'Doc\_visits', 'Full\_meals\_eaten', 'vitD\_supp',  
 'Soft\_drink', 'Initial\_admin', 'HighBlood', 'Stroke',  
 'Complication\_risk', 'Overweight', 'Arthritis', 'Diabetes',  
 'Hyperlipidemia', 'BackPain', 'Anxiety', 'Allergic\_rhinitis',  
 'Reflux\_esophagitis', 'Asthma', 'Services', 'Initial\_days',  
 'TotalCharge', 'Additional\_charges', 'Item1', 'Item2', 'Item3', 'Item4',  
 'Item5', 'Item6', 'Item7', 'Item8'],  
 dtype='object')

In [22]:

1

In [23]:

1

med\_data.dtypes *# looking at datatypes for each variable.*

Out[23]:

CaseOrder int64  
Customer\_id object  
Interaction object  
UID object  
City object  
State object  
County object  
Zip int64  
Lat float64  
Lng float64  
Population int64  
Area object  
TimeZone object  
Job object  
Children int64  
Age int64  
Income float64  
Marital object  
Gender object  
ReAdmis object  
VitD\_levels float64  
Doc\_visits int64  
Full\_meals\_eaten int64  
vitD\_supp int64  
Soft\_drink object  
Initial\_admin object  
HighBlood object  
Stroke object  
Complication\_risk object  
Overweight object  
Arthritis object  
Diabetes object  
Hyperlipidemia object  
BackPain object  
Anxiety object  
Allergic\_rhinitis object  
Reflux\_esophagitis object  
Asthma object  
Services object  
Initial\_days float64  
TotalCharge float64  
Additional\_charges float64  
Item1 int64  
Item2 int64  
Item3 int64  
Item4 int64  
Item5 int64  
Item6 int64  
Item7 int64  
Item8 int64  
dtype: object

In [24]:

1

med\_data.isnull().sum() *# checking for missing data*

2

Out[24]:

CaseOrder 0  
Customer\_id 0  
Interaction 0  
UID 0  
City 0  
State 0  
County 0  
Zip 0  
Lat 0  
Lng 0  
Population 0  
Area 0  
TimeZone 0  
Job 0  
Children 0  
Age 0  
Income 0  
Marital 0  
Gender 0  
ReAdmis 0  
VitD\_levels 0  
Doc\_visits 0  
Full\_meals\_eaten 0  
vitD\_supp 0  
Soft\_drink 0  
Initial\_admin 0  
HighBlood 0  
Stroke 0  
Complication\_risk 0  
Overweight 0  
Arthritis 0  
Diabetes 0  
Hyperlipidemia 0  
BackPain 0  
Anxiety 0  
Allergic\_rhinitis 0  
Reflux\_esophagitis 0  
Asthma 0  
Services 0  
Initial\_days 0  
TotalCharge 0  
Additional\_charges 0  
Item1 0  
Item2 0  
Item3 0  
Item4 0  
Item5 0  
Item6 0  
Item7 0  
Item8 0  
dtype: int64

In [25]:

1

med\_data.duplicated().any() *# checking for duplicates*

2

Out[25]:

False

In [26]:

1

*# separating the variables that will be used for clustering*

2

new\_data **=** med\_data[['Lat','Lng','Additional\_charges']].copy()

In [27]:

1

new\_data *# these are the columns I will be working with*

Out[27]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lat** | **Lng** | **Additional\_charges** |
| **0** | 34.34960 | -86.72508 | 17939.403420 |
| **1** | 30.84513 | -85.22907 | 17612.998120 |
| **2** | 43.54321 | -96.63772 | 17505.192460 |
| **3** | 43.89744 | -93.51479 | 12993.437350 |
| **4** | 37.59894 | -76.88958 | 3716.525786 |
| **...** | ... | ... | ... |
| **9995** | 36.42886 | -78.23716 | 8927.642000 |
| **9996** | 39.43609 | -74.87302 | 28507.150000 |
| **9997** | 36.36655 | -87.29988 | 15281.210000 |
| **9998** | 44.10354 | -102.01590 | 7781.678000 |
| **9999** | 40.49998 | -80.19959 | 11643.190000 |

10000 rows × 3 columns

In [28]:

1

new\_data.describe()

Out[28]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Lat** | **Lng** | **Additional\_charges** |
| **count** | 10000.000000 | 10000.000000 | 10000.000000 |
| **mean** | 38.751099 | -91.243080 | 12934.528587 |
| **std** | 5.403085 | 15.205998 | 6542.601544 |
| **min** | 17.967190 | -174.209700 | 3125.703000 |
| **25%** | 35.255120 | -97.352982 | 7986.487755 |
| **50%** | 39.419355 | -88.397230 | 11573.977735 |
| **75%** | 42.044175 | -80.438050 | 15626.490000 |
| **max** | 70.560990 | -65.290170 | 30566.070000 |

In [29]:

1

2

**from** scipy.stats **import** zscore

In [30]:

1

*# identifying and treating outliers*

2

3

*# Calculate Z-scores*

4

z\_scores **=** new\_data.apply(zscore)

5

6

*# Filter out rows where any column's Z-score is greater than 3 or less than -3*

7

clean\_data **=**new\_data [(z\_scores.abs() **<** 3).all(axis**=**1)]

8

9

print(clean\_data)

Lat Lng Additional\_charges  
0 34.34960 -86.72508 17939.403420  
1 30.84513 -85.22907 17612.998120  
2 43.54321 -96.63772 17505.192460  
3 43.89744 -93.51479 12993.437350  
4 37.59894 -76.88958 3716.525786  
... ... ... ...  
9995 36.42886 -78.23716 8927.642000  
9996 39.43609 -74.87302 28507.150000  
9997 36.36655 -87.29988 15281.210000  
9998 44.10354 -102.01590 7781.678000  
9999 40.49998 -80.19959 11643.190000  
  
[9853 rows x 3 columns]

*# Scaling the data*

3

scaler **=** StandardScaler()

4

scaled\_data **=** scaler.fit\_transform(clean\_data)

5

print (scaled\_data)

[[-0.9196121 0.28826796 0.76364581]  
 [-1.65400618 0.39730563 0.71377025]  
 [ 1.00699386 -0.43422125 0.69729727]  
 ...  
 [-0.49694158 0.24637329 0.35746711]  
 [ 1.12441619 -0.8262134 -0.78848042]  
 [ 0.36925688 0.76388255 -0.19843158]]

In [31]:

1

med\_data.to\_csv('MSDA212\_PA\_cleanData.cvs')

**Part IV: Analysis**

D. Perform the data analysis, and report on the results by doing the following:

1. Determine the optimal number of clusters in the data set, and describe the method used to determine this number.

2. Provide the code used to perform the clustering analysis technique.

1

*# looking that the data before clustering*

2

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

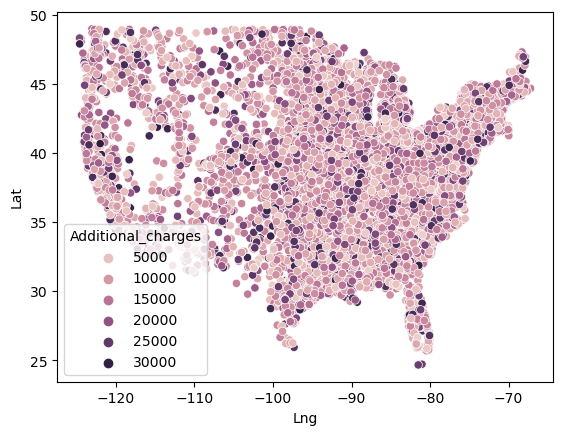
3

4

sns.scatterplot(data**=**clean\_data ,x**=**'Lng', y **=**'Lat', hue**=**'Additional\_charges')

Out[86]:

<Axes: xlabel='Lng', ylabel='Lat'>



In [87]:

1

2

**from** sklearn.preprocessing **import** StandardScaler

3

**from** scipy.cluster.vq **import** kmeans, vq

4

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

In [88]:

1

2

*# Scaling the data*

3

scaler **=** StandardScaler()

4

scaled\_data **=** scaler.fit\_transform(clean\_data)

5

print (scaled\_data)

[[-0.9196121 0.28826796 0.76364581]  
 [-1.65400618 0.39730563 0.71377025]  
 [ 1.00699386 -0.43422125 0.69729727]  
 ...  
 [-0.49694158 0.24637329 0.35746711]  
 [ 1.12441619 -0.8262134 -0.78848042]  
 [ 0.36925688 0.76388255 -0.19843158]]

In [90]:

1

*# K-means clustering*

2

centroids, \_ **=** kmeans(scaled\_data, 3)

3

cluster\_labels, \_ **=** vq(scaled\_data, centroids)

4

5

scaled\_data **=** clean\_data.copy()

6

7

*# Adding cluster labels to the original DataFrame*

8

scaled\_data['cluster\_labels'] **=** cluster\_labels

9

10

Now, I will calculate the distortion for cluster ranges between 2 and 11. Distortion measures how well the clusters fit the data; it decreases as the number of clusters increases. As shown below, there is a significant drop in distortion from 2 to 3 clusters, indicating that the optimal number of clusters might be 3. I will plot the elbow method for better visualization.

In [91]:

1

*#Calculatting distortions for different numbers of clusters*

2

3

distortions **=** []

4

num\_clusters **=** range(2, 11)

5

6

**for** i **in** num\_clusters:

7

centroids, distortion **=** kmeans(scaled\_data, i)

8

distortions.append(distortion)

9

cluster\_labels, \_ **=** vq(scaled\_data, centroids)

10

11

print(f"Number of clusters: {i}, Distortion: {distortion}")

12

Number of clusters: 2, Distortion: 2958.8603608596145  
Number of clusters: 3, Distortion: 1852.771780753345  
Number of clusters: 4, Distortion: 1501.1789544693047  
Number of clusters: 5, Distortion: 1168.830973074431  
Number of clusters: 6, Distortion: 973.7290661833605  
Number of clusters: 7, Distortion: 840.8853850173479  
Number of clusters: 8, Distortion: 729.3103032290724  
Number of clusters: 9, Distortion: 659.1108696731136  
Number of clusters: 10, Distortion: 586.0325665872089

In [92]:

1

*# The elbow plot*

2

3

plt.figure(figsize**=**(8, 4))

4

plt.plot(num\_clusters, distortions, marker**=**'o')

5

plt.title('Elbow Method for Optimal Number of Clusters')

6

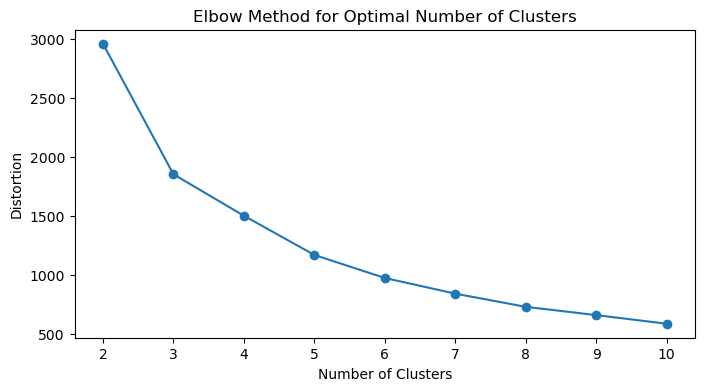
plt.xlabel('Number of Clusters')

7

plt.ylabel('Distortion')

8

plt.show()



1

Looking at the plot, there is a significant drop in distortion from 2 to 3 clusters, and the graph continues to decrease. The elbow plot clearly shows that 3 is the optimal number of clusters for analyzing additional charges paid across the country.

1

2

The scatter plot below illustrates three distinct groups, each representing different amounts paid for additional charges. To enhance the clarity of this visualization, I will compute a cluster summary.

3

4

As shown in the cluster summary code below, the average additional charge per patient is $10,683.01 in Cluster 0. In Clusters 1 and 2, the average additional charges per patient are $9,916.44 and $23,714.35, respectively.

In [71]:

1

plt.figure(figsize**=**(8, 6))

2

sns.scatterplot(x**=**'Lng', y**=**'Lat', hue**=**'cluster\_labels', data**=**scaled\_data)

3

plt.title('K-means Clustering Results')

4

plt.xlabel('Lng')

5

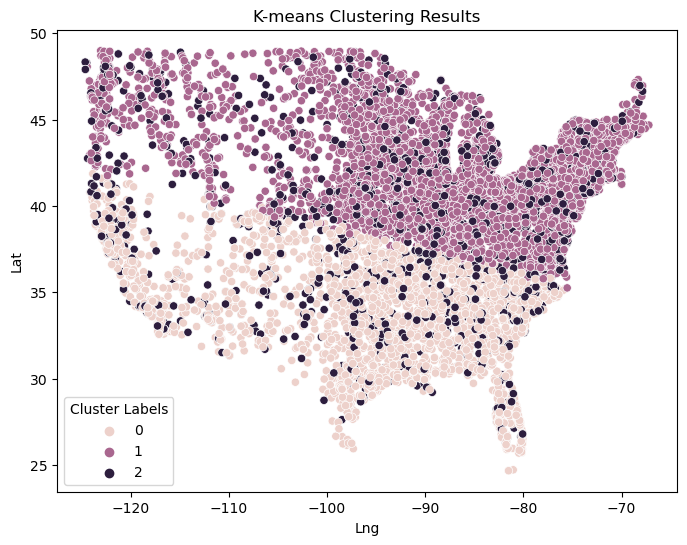
plt.ylabel('Lat')

6

plt.legend(title**=**'Cluster Labels')

7

plt.show()



In [70]:

1

cluster\_summary **=** scaled\_data.groupby('cluster\_labels')['Additional\_charges'].describe()

2

3

cluster\_summary

Out[70]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
| **cluster\_labels** |  |  |  |  |  |  |  |  |
| **0** | 3031.0 | 10683.007144 | 3997.207152 | 3132.25999 | 7605.522182 | 10560.780000 | 13463.824975 | 25608.58362 |
| **1** | 4830.0 | 9916.443931 | 3493.837939 | 3125.70300 | 7012.790847 | 9927.300912 | 12729.369395 | 19289.53000 |
| **2** | 1992.0 | 23714.347560 | 3457.125901 | 16172.56000 | 21039.692500 | 23884.801360 | 26587.915000 | 30566.07000 |

In [ ]:

1

**Part V: Data Summary and Implications**

E. Summarize your data analysis by doing the following:

1. Explain the quality of the clusters created.

I used K-means clustering for this analysis, which is an effective technique for clustering continuous variables such as latitude, longitude, and additional charges. The optimal number of clusters was determined to be 3. The model generated three distinct regions for the additional charges that patients pay based on their locations.

2. Discuss the results and implications of your clustering analysis.

The clustering analysis reveals that additional charges vary significantly based on location. This insight can be crucial for healthcare providers or businesses in adjusting their pricing strategies based on regional factors. Businesses can optimize their services by understanding regional demand and cost variations. This can lead to more efficient operations and better customer satisfaction.

3. Discuss **one** limitation of your data analysis.

One limitation I found in my analysis is that determining the optimal number of clusters is challenging. The Elbow method offers guidance but is not foolproof. Analysts need domain knowledge to interpret these results effectively and make the best decision.

4. Recommend a course of action for the real-world organizational situation from part A1 based on the results and implications discussed in part E2.

Implement a dynamic pricing model that adjusts additional charges based on the geographical regions identified by the clusters. The analysis shows significant variability in additional charges based on location. A dynamic pricing strategy can help balance affordability for customers and profitability for the organization.

Allocate more resources to regions with higher additional charges to provide financial support or alternative payment options. This can help mitigate the financial burden on customers in high-charge regions, improving access to services and customer satisfaction.

**Part VI: Demonstration**

G. <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=67fcc0a2-197a-45b7-aac5-b1bb0139b51c>

H.

DataCamp. (n.d.). K-means clustering in Python. \*DataCamp\*. <https://www.datacamp.com/tutorial/k-means-clustering-python>

GeeksforGeeks. (n.d.). Demonstration of K-means assumptions. GeeksforGeeks. Retrieved from <https://www.geeksforgeeks.org/demonstration-of-k-means-assumptions/>

McDonald, A. (2021, June 3). How to use unsupervised learning to cluster well log data using Python. \*Towards Data Science\*. <https://towardsdatascience.com/how-to-use-unsupervised-learning-to-cluster-well-log-data-using-python-a552713748b5>

I.