D212 Task 1 Clustering Techniques

Western Governs University

Table of Contents

Part I: Research Question

A1. Proposal of Question

"Is it possible to identify specific customers groups by analyzing their purchasing behaviors and demographics, enabling marketing strategies for more effective customer targeting using the k-means clustering technique?"

A2. Defined Goal

K-means clustering can be used to help identify groups of customers with similar characteristics. K-means can be used to segment customers into groups based on characteristics such as usage patterns, demographic information, or other relevant features. The goal of this analysis to identify distinct customer segments based on age and monthly charge in order to develop targeted marketing strategies and pricing plans.

Part II: Technique Justification

B1. Explanation of Clustering Technique

K-means clustering analyzes data by grouping similar data points together in clusters based on their features. The techniques partitions the data into k clusters, where k is a pre-defined number of clusters. The algorithm works by randomly selecting k data points from the dataset to be the initial centroids for the clusters. Each data points is then assigned to the nearest centroid based on its Euclidean distance. Lastly, the centroids are recalculated as the mean of the data points assigned to its cluster.

By analyzing the data points within each cluster, you can gain insights into the characteristics, behaviors, or patterns that define each cluster. These insights can help identify meaningful segments within the dataset and the structure and composition of the dataset.

B2. Summary of Technique Assumption

One assumption of k-means clustering is all the clusters have the same size, which means that the technique will assign an equal number of data points to each cluster (Bishop, n.d.).

B3. Packages or Libraires List

The following packages/libraires were used for k-means cluster:

Package/Library	Description
numpy	For numerical computing in Python. It provides support for arrays and matrices, as well as mathematical functions.
pandas	For data manipulation and analysis. It provides data structures for efficient handling of data and tools for data cleaning, merging, and transformation.
Series	For one-dimensional labeled data
DataFrame	For two-dimensional labeled data
seaborn	A visualization library for statistical graphics plotting
matplotlib.pyplot	For creating static, animated, and interactive visualizations
scipy	For scientific computing in Python.
StandardScaler	A preprocessing module from scikit-learn that standardizes features by removing the mean and scaling to unit variance.
metrics	A module from scikit-learn that provides various metrics for evaluating the performance of machine learning models.
KMeans	A clustering algorithm from scikit-learn that partitions data into k clusters based on their similarity.
silhouette_score	Used to evaluate the quality of clustering results.

Part III: Data Preparation

C1. Data Prepossessing

Standardization is important in k-means clustering because the algorithm is based on distance between data points, which is sensitive to the scale of the variables. Outliers can have a significant impact on the clustering result by distorting the cluster centers. Real data often has some level of noise and outliers that can make it challenging to identify meaningful clusters (Ryzhkov, 2021). Therefore, techniques to handle outliers and noise is needed. Transformation of data to a normal distribution can help to reduce the impact of outliers and non-normality (Ryzhkov, 2021). Standardization is important in k-means clustering, and handling outliers and noise is a critical preprocessing step to ensure the clustering results are meaningful and useful for the analysis.

C2. Dataset Variables

The following variables were used to preform k-means clustering:

Variable	Data Type
Children	Continuous
Age	Continuous
Income	Continuous
Outage_sec_perweek	Continuous
Email	Continuous
Contacts	Continuous
Yearly_equip_failure	Continuous
Tenure	Continuous
MonthlyCharge	Continuous
Bandwidth_GB_Year	Continuous

All continuous variables were used in this analysis because k-means clustering in order to simply the clustering process and reduce the risk of overfitting.

C3. Steps For Analysis

1. Import packages and/or libraries going to be used in the analysis

```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.pipeline import make_pipeline
%matplotlib inline
from sklearn.cluster import KMeans
```

2. Import data file

```
file_path = "/Users/igmark/Desktop/WGU Data Files/D212_churn_clean.csv"
df = pd.read_csv(file_path)
```

3. View data headers

```
df.head()
```

4. View descriptive statistics

```
df.describe()
```

5. Detect missing values

```
df.isnull().sum()
```

6. Detect duplicate values

```
df.duplicated()
```

7. Check for and remove outliers

```
print(df.shape)
df = df[(np.abs(stats.zscore(df.select_dtypes(include=np.number))) < 3).all(axis=1)]
print(df.shape)
```

8. Copy original data frame

```
data_orig = df.copy()
```

9. View columns in data set

```
df.columns
```

10. Remove uncessesary columns

```
df=df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State',
'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone', 'Job', 'Marital', 'Gender', 'Churn',
'Techie', 'Contract', 'Port_modem', 'Tablet', 'InternetService',
'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup',
'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',
'PaperlessBilling', 'PaymentMethod', 'Item1', 'Item2', 'Item3', 'Item4', 'Item5',
'Item6', 'Item7', 'Item8'])
```

11. View remaining columns in data set

```
df.columns
```

12. Create histograms of each variable in data set (done for visualization of the distribution of each variable)

```
df.hist(figsize = (15,15))
```

12. Create boxplots for each variable (visualize distribution)

```
df.boxplot(figsize = (15,15))
```

13. View scatterplots for bivariant analysis of variables in data set

```
sns.scatterplot(x=df['Outage_sec_perweek'],y=df['Tenure'])
plt.show()

sns.scatterplot(x=df['MonthlyCharge'],y=df['Tenure'])
plt.show()

sns.scatterplot(x=df['Bandwidth_GB_Year'],y=df['MonthlyCharge'])
plt.show()

sns.scatterplot(x=df['MonthlyCharge'],y=df['Income'])
plt.show()

sns.scatterplot(x=df['Income'],y=df['Tenure'])
plt.show()

sns.scatterplot(x=df['Bandwidth_GB_Year'],y=df['Tenure'])
plt.show()
```

14. Perform standardization on dataset

```
sc = StandardScaler()
sc.fit(df)
scaled_data_array = sc.transform(df)
scaled_data = pd.DataFrame(scaled_data_array, columns = df.columns)
scaled_data.head()
```

15. Set prepared dataset to a new date frame for k-means clustering

```
df.to_csv('D212_prepared_task1.csv')
```

C4. Cleaned Dataset

Cleansed dataset attached and titled:

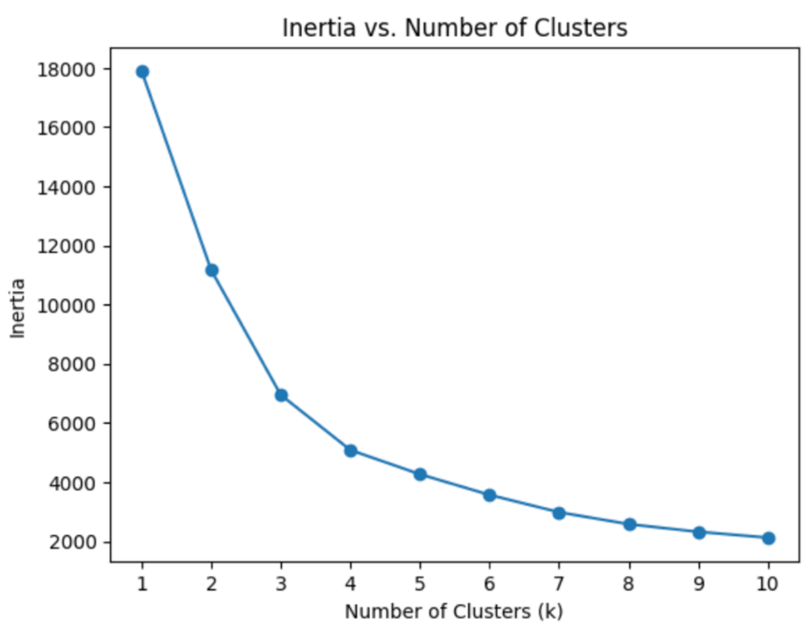
```
df.to_csv('D212_prepared_task1.csv')
```

Part IV: Analysis

D1. Output and intermediate Calculations

K-means clustering analyzes data by grouping similar data points together in clusters based on their features. The techniques partitions the data into k clusters, where k is a pre-defined number of clusters. The elbow method was used to determine the optimal number of clusters (k). This method plots the value of the sum of squared distances between the data points and their assigned clusters against the number of clusters used in the algorithm. The resulting curve should resemble an arm with the "elbow" point on the curve representing the optimal number of clusters (Onumanyi et al., 2022).

```
selected_columns = ['Age','MonthlyCharge']
selected_data = scaled_data[selected_columns]

ks = range(1, 11)
inertias = []

for k in ks:
    model = KMeans(n_clusters=k)
    model.fit(selected_data)
    inertias.append(model.inertia_)

plt.plot(ks, inertias, '-o')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('Inertia')
plt.title('Inertia vs. Number of Clusters')
plt.xticks(ks)
plt.show()
```



D2. Code Execution

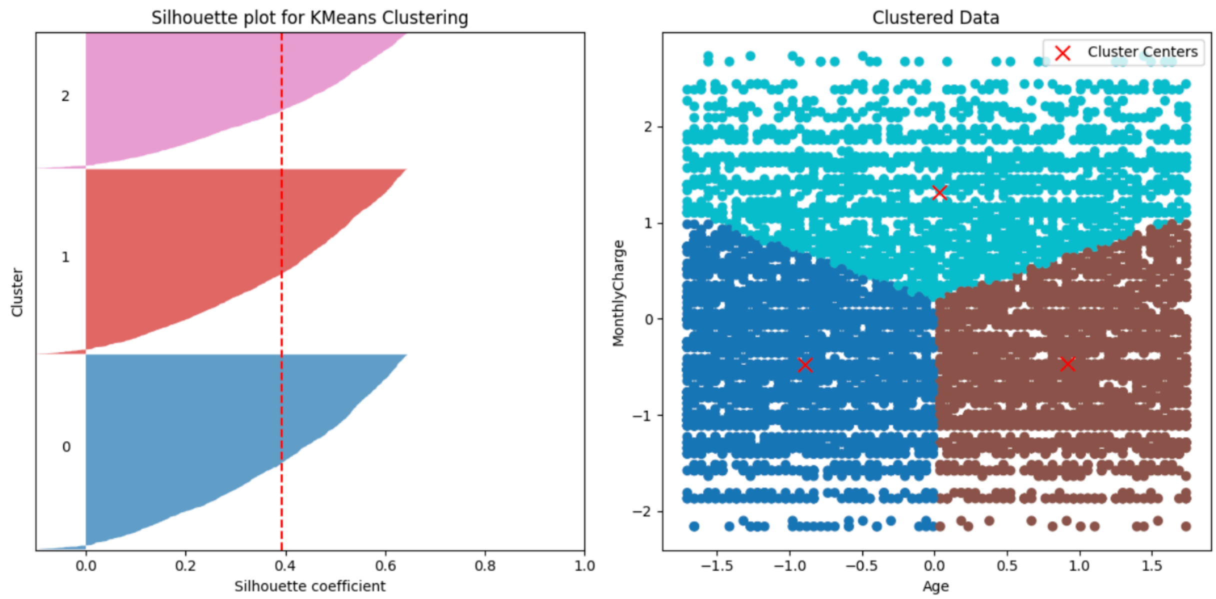
Code used to preform k-means clustering:

```
#performs k-means clustering
model = KMeans(n_clusters=3)
model.fit(selected_data)
print(model.labels_)
```

```
#coordinates of the centers of the clusters obtained after fitting the KMeans model on the data.
model.cluster_centers_

selected_columns = ['Age', 'MonthlyCharge']
selected_data = scaled_data[selected_columns]

k = 3  # Number of clusters

# Perform KMeans clustering
model = KMeans(n_clusters=k)
labels = model.fit_predict(selected_data)

# Calculate silhouette scores
silhouette_vals = silhouette_samples(selected_data, labels)

# Sort the silhouette scores and cluster labels
sorted_vals = silhouette_vals.argsort()
sorted_labels = labels[sorted_vals]

# Compute the silhouette score for the entire dataset
silhouette_avg = silhouette_vals.mean()
print("Average silhouette score:", silhouette_avg)

# Create a subplot with 1 row and 2 columns
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

# Set the limits of the y-axis for individual silhouette plots
y_lower = 10

# Iterate over clusters to create silhouette plots
for i in range(k):
   # Aggregate the silhouette scores for samples in the cluster
   cluster_vals = silhouette_vals[sorted_labels == i]
   cluster_size = cluster_vals.shape[0]

   # Sort the silhouette scores within the cluster
   cluster_vals.sort()

   # Calculate the upper limit of the silhouette plot for the cluster
   y_upper = y_lower + cluster_size

   # Fill the silhouette plot with the corresponding cluster color
   color = plt.cm.get_cmap("tab10")(float(i) / k)
   ax[0].fill_betweenx(np.arange(y_lower, y_upper), 0, cluster_vals, facecolor=color, alpha=0.7)
```

```
    # Label the silhouette plot with the cluster number
    ax[0].text(-0.05, y_lower + 0.5 * cluster_size, str(i))

    # Update the lower limit of the y-axis for the next cluster plot
    y_lower = y_upper + 10


# Set labels and limits for the silhouette plot
ax[0].set_xlabel("Silhouette coefficient")
ax[0].set_ylabel("Cluster")
ax[0].set_title("Silhouette plot for KMeans Clustering")
ax[0].set_xlim([-0.1, 1])
ax[0].set_ylim([0, len(selected_data) + (k + 1) * 10])

# Draw a vertical line at the average silhouette score
ax[0].axvline(x=silhouette_avg, color="red", linestyle="--")
ax[0].set_yticks([])  # Clear y-axis ticks


# Plot the cluster centers
cluster_centers = model.cluster_centers_
ax[1].scatter(selected_data['Age'], selected_data['MonthlyCharge'], c=labels, cmap='tab10')
ax[1].scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:, 1], marker='x', color='red',
s=100, label='Cluster Centers')
ax[1].set_xlabel('Age')
ax[1].set_ylabel('MonthlyCharge')
ax[1].set_title('Clustered Data')
ax[1].legend()

plt.tight_layout()
plt.show()
```

```
Average silhouette score: 0.391951046190778
```



Part VI: Data Summary and Implications

E1. Accuracy of Clustering Technique

To evaluate the accuracy of the k-means cluster the Silhouette method was used evaluate how similar the data in each cluster is and how well the clusters are separated. The silhouette score is the average distance between a data point and other data points within its own cluster and the average distance between the data point and data points in the nearest neighboring cluster (Kumar, 2023). The silhouette score ranges from -1 to 1. A score closer to -1 indicates that the clusters are not well separated. A score closer to 1 indicates the clusters are well separated. Higher values indicate better clusters. The average silhouette score of this analysis is 0.343. This score suggest that clustering has moderate to fair separation and compactness and there is from for improvement.

E2. Results and Implications

The elbow method was used to determine the optimal value of k, which was 3. The silhouette method also showed that the best value for k is 3 (For n_clusters = 3, the silhouette score is 0.3919683875030648) which was the highest value in the given range of clusters. Thus the optimal cluster number is 3. This was used to complete the k-means cluster and graphing a scatter plot of those clusters. The cluster centers are given by the array `[[ 0.91748404, -0.46470801], [-0.89173728, -0.47454673], [ 0.03352804,  1.32009394]]`. These values represent the centroid coordinates for each cluster. The first cluster center has a higher value for age and a lower value for monthly charge. The cluster may be interpreted as customers who are older and have lower monthly charges. The second cluster center has a lower value for both age and monthly charge variables. This may be interpreted as customers who are younger may also have lower monthly charges. The third cluster center has high values for both age and monthly charge. This may be interpreted as those customers who are older have a higher monthly charge. However, when examining the scatterplot there is an even distribution and clear patterns cannot be drawn. There may not be a strong relationship or association between the variables age and monthly charge.

E3. Limitation

One limitation to this analysis is that the analysis only focuses on two variables, age and monthly charges. The limitation by no capture the full complexity of customer behaviors or preferences. More variables such as customer demographics, usage patterns, or service preferences could provide more insights.

E4. Course of Action

The next step in this analysis would be to incorporate additional variables for a more comprehensive view of customer behavior and preferences. Refine the clustering technique to capture the underlying patterns of the data. Also gathering data could be helpful. Age and monthly charge appear to be important variables in examining telecommunication data. It would be of benefit to gather more data for an increased sample size. With an increased sample size it may be possible to gain definite relationship in specific age and monthly charges to increase marketing strategies.

Part V. Demonstration

F. Panopto Recording

Link to recording: https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=4a4c8fbb-701a-4ff2-9bbf-b00c0004805f

G. Sources for Third-Party Code

sklearn.metrics.accuracy_score. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html

H. Sources

Bishop, J. W. a. C. (n.d.). *Model-Based Machine Learning (Early Access): Chapter 8. How to Read a Model*. MBML Book. https://www.mbmlbook.com/ModelAnalysis_K-means_Clustering.html

Demonstration of k-means assumptions. (n.d.). Scikit-learn. https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_assumptions.html

k-Means Advantages and Disadvantages. (n.d.). *Google Developers*. https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

K-Means Cluster Analysis. (2023, March 13). Columbia University Mailman School of Public Health. https://www.publichealth.columbia.edu/research/population-health-methods/k-means-cluster-analysis

Kumar, A. (2023, April 27). *KMeans Silhouette Score Python Example - Data Analytics*. Data Analytics. https://vitalflux.com/kmeans-silhouette-score-explained-with-python-example/

Onumanyi, A. J., Molokomme, D. N., Isaac, S. J., & Abu-Mahfouz, A. M. (2022). AutoElbow: An Automatic Elbow Detection Method for Estimating the Number of Clusters in a Dataset. *Applied Sciences*, *12*(15), 7515. https://doi.org/10.3390/app12157515

Ryzhkov, E. (2021, December 15). 5 Stages of Data Preprocessing for K-means clustering. *Medium*. https://medium.com/@evgen.ryzhkov/5-stages-of-data-preprocessing-for-k-means-clustering-b755426f9932