Telecom Churn Analysis
Clustering Techniques

Javier Lopez

D212 - Data Mining II

Table of Contents

Part I: Research Question

A1. Research Question

What are the distinct customer segments within the telecommunications company's customer base?

A2. Objective

Our goal is to segment telecom customers based on their characteristics by performing k-means clustering. Segmenting telecom customers will allow us to better understand the similarities and differences between different groups of customers.

Part 2: Technique Justification

B1. Clustering Technique

We used k-means clustering to segment our customers based on their characteristics. K-means is an unsupervised learning algorithm that groups data points in a dataset into a specified number of clusters. The algorithm starts by randomly selecting data points to serve as cluster centroids for the specified amount of clusters, then it assigns iteratively each data point to the nearest centroid by calculating the Euclidean between them and recalculates the centroid for each cluster based on the newly assigned data points. The process continues until the cluster centroids no longer change.

The resulting clusters represent groups of telecom customers with similar characteristics. We're expecting to get a clear set of customer segments, each with a central point called a

centroid, which will help us understand what makes that group unique. K-means is a useful way to break down our customer data and see what patterns emerge.

B2. Assumption

The k-means clustering algorithm assumes that all variables have an equal variance, or standard deviation. The reason for the assumption is because k-means uses Euclidean distance to measure the distance between data points and the centroids, and Euclidean distance is sensitive to differences in the scale of variables. To comply with this assumption, we normalized our data using Sklearn's StandardScaler function. Data normalization ensures that all variables are measured using the same scale.

B3. Packages

We used a total of five libraries and six packages to complete our analysis. Below is a list of the packages and their respective libraries including their contribution to our analysis.

1. **NumPy**

   The NumPy library provides tools that allow us to work with arrays and perform mathematical operations. We used NumPy to create an array that stored the inertia values to be plotted into the Elbow Method graph.

2. **Pandas**

   The Pandas library provides tools to organize and manipulate data into tables. We used Pandas to import our data into a table and create multiple other tables to store calculations.

3. **Seaborn**

   The Seaborn library provides tools that allow us to visualize our data. We used Seaborn

to plot our inertia values for the Elbow Method into a line graph and the churn rates for clusters into a scatter plot.

4. **Matplotlib**

   The Matplotlib library provides tools that support the Seaborn library in visualizing our data. We used Matplotlib to add titles and grid lines to our visualizations.

5. **Sklearn**

   The Sklearn library provides tools that supports data preprocessing and the implementation of machine learning algorithms. We used the preprocessing package to scale our data, the cluster package to implement the k-means unsupervised learning algorithm, and the metrics package to calculate the silhouette score for each value of k.

Part III. Data Preparation

C1. Data Preprocessing

The k-means clustering algorithm only takes in continuous variables, which meant that we needed to remove all categorical variables before feeding it to the algorithm.

C2. Variables

Variable Name	Data Type	Data Level
Income	float	continuous
Outage_sec_perweek	float	continuous
Tenure	float	continuous
MonthlyCharge	float	continuous
Bandwidth_GB_Year	float	continuous

C3. Data Preparation Cadence

1. Import the data into a Pandas dataframe.

   ```
   df = pd.read_csv('churn_clean.csv')
   ```

2. Isolate churn and the continuous variables that will be fed to the k-means clustering algorithm.

   ```
   keep = []
   for col in df.columns:
      if df[col].dtype == float:
         keep.append(col)
   df = df[keep]
   ```

3. Drop location-based longitude and latitude columns.

   ```
   df.drop(['Lat', 'Lng',], axis=1, inplace=True)
   ```

4. Scale the data to be fed to the k-means clustering algorithm.

   ```
   scaler = StandardScaler()
   scaled_df = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
   ```

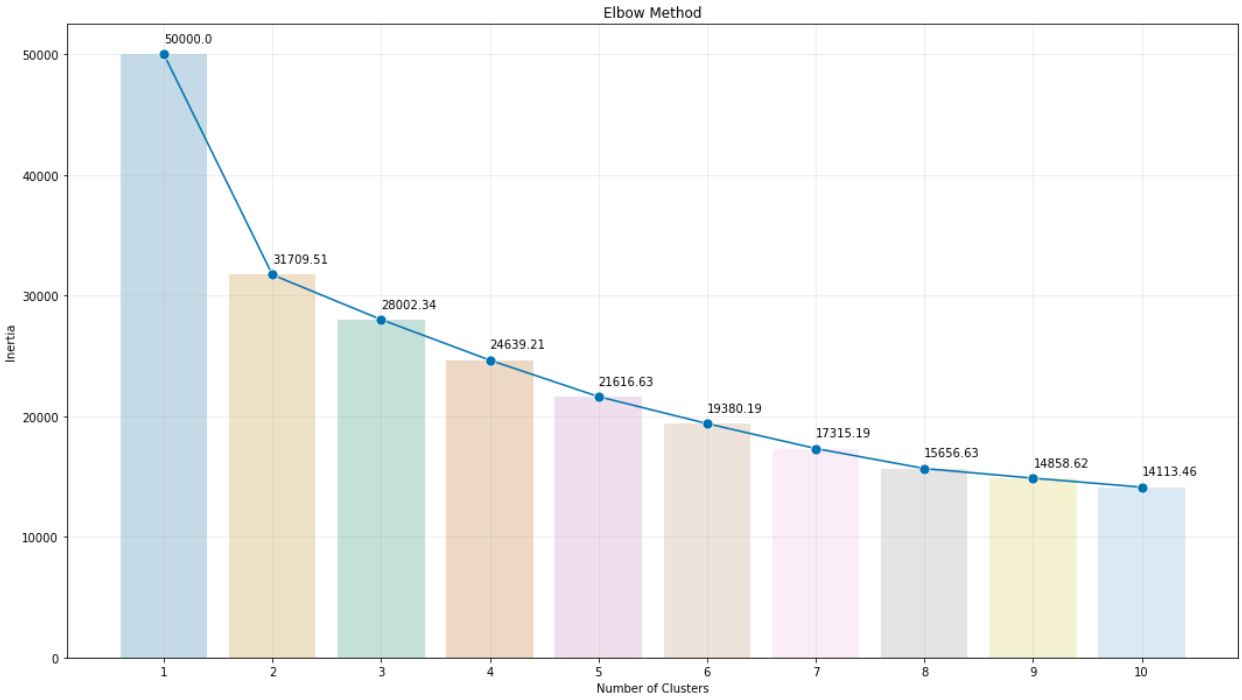
5. Store the clean dataset.

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

Part IV: Analysis

D1. Analysis Technique

Our analysis technique involved testing multiple k-means clustering models with different k values between one and ten. We got the inertia scores for each model and the amount of clusters, k, for each and plotted them. We then used the Elbow Method to visually determine the best k value for our final model, pictured in the graph above. We then scored our final model by calculating the silhouette score.




```
## Get the predicted labels
predicted_labels = kmeans.fit_predict(scaled_df)
## Calculate the silhouette score
silhouette = silhouette_score(scaled_df, predicted_labels)
print(f'Silhouette Score: {silhouette}, {n_clusters} clusters')
```

```
Silhouette Score: 0.35763272219672476, 2 clusters
```

D2. Clustering Analysis Code

```
## Determine the best value for k
inertia = np.array([])
k_vals = range(1,11)

for k in k_vals:
    kmeans = KMeans(n_clusters=k, random_state=10)
    kmeans.fit(scaled_df)
    inertia = np.append(inertia, kmeans.inertia_)

inertia_vals = pd.DataFrame(inertia, index=k_vals, columns=['Inertia'])
## Plot the inertia values
sns.barplot(x=inertia_vals.index, y=inertia_vals.Inertia, alpha=0.25)
sns.lineplot(x=inertia_vals.index-1, y=inertia_vals.Inertia, marker='o', markersize=9, legend=False)
```

```
plt.title('Elbow Method')
plt.xticks(inertia_vals.index-1)
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
for i in inertia_vals.index:
    plt.text(
        x=i-1,
        y=inertia_vals.Inertia[i]+1000,
        s=round(inertia_vals.Inertia[i], 2)
    )
plt.grid(alpha=0.25)
## Perform kmeans clustering
n_clusters = 2
kmeans = KMeans(n_clusters=n_clusters, random_state=10)
kmeans.fit(scaled_df)
## Get the predicted labels
predicted_labels = kmeans.fit_predict(scaled_df)
## Calculate the silhouette score
silhouette = silhouette_score(scaled_df, predicted_labels)
print(f'Silhouette Score: {silhouette}, {n_clusters} clusters')
## Assign cluster labels
df['Cluster'] = kmeans.labels_
## Calculate cluster summary
cluster_summary = df.groupby('Cluster').agg(['mean', 'median', 'std']).transpose()
cluster_summary.columns = ['Cluster 1', 'Cluster 2']
cluster_summary
## Get centroids
centroids = pd.DataFrame(
    scaler.inverse_transform(kmeans.cluster_centers_),
    index=['Cluster 1', 'Cluster 2'],
    columns=df.columns[:-1]
)
## View centroids
print('Cluster centroids:\n')
for i in centroids.index:
    print(i)
    for col in centroids.columns:
```

```
        print(f'{col}: {round(centroids[col][i], 2)}')
    print()
## Plot centroids
centr_x = [sum(df[df.Cluster==1].index)/len(df[df.Cluster==1]),
sum(df[df.Cluster==2].index)/len(df[df.Cluster==2])]
titles = [
    'Income ($)',
    'Weekly Outage Duration (sec)',
    'Tenure (months)',
    'Monthly Charge ($)',
    'Annual Bandwidth Usage (GB)'
]
y_labels = [
    'Income ($)',
    'Outage Duration (sec)',
    'Tenure (months)',
    'Charge Amount ($)',
    'Bandwidth Usage (GB)'
]
fig,ax = plt.subplots(1,5)
plt.suptitle('Cluster Characteristics and Centroids')
for i in range(5):
    sns.scatterplot(x=df.index, y=df[df.columns[i]], hue=df.Cluster, ax=ax[i], legend=False)
    sns.scatterplot(x=centr_x, y=centroids[centroids.columns[i]], s=150, ax=ax[i])
    ax[i].set_title(titles[i])
    ax[i].set_ylabel(y_labels[i])
ax[4].legend(['Cluster', 'Centroid'], title='Legend', loc='lower right')
plt.tight_layout()
```

Part V. Data Summary and Implications

E1. Accuracy

The accuracy of clustering techniques can be measured using the inertia, or more popularly used, the silhouette score. The silhouette score measures how well each data point fits within its assigned cluster and how distinct it is from other clusters. A silhouette score ranges from -1 to 1, -1 being a poor score, and 1 being a perfect fit. Our final model had a silhouette score of 0.358 and inertia of 31,709.51, which can be considered a moderate score. Multiple factors can play a role into the silhouette scores, including the similarity between data points.

E2. Results and Implications

Our clustering analysis resulted in two customer groups with distinct centroids. Cluster 1 and 2 had similar centroids for Income, Weekly Outage Duration, and Monthly Charge. However, there was a significant difference in the Tenure and Annual Bandwidth Usage between the two clusters. Cluster 1 had an average tenure of 9.13 months, while Cluster 2 had an average tenure of 59.93 months, a difference of 50.9 months.

Additionally, Cluster 1 had an average annual bandwidth usage of 1,312.29 GB, while Cluster 2 had an average annual bandwidth usage of 5,473.23 GB, a difference of 4,160.94. The difference between the two clusters in tenure and bandwidth usage imply that customers from Cluster 1 are churning before completing a 12-month tenure with the organization, likely resulting in a lower than average annual bandwidth usage. Another implication is that the organization can use the cluster analysis to make decisions regarding resource allocation,

create targeted marketing campaigns to customers based on their cluster characteristics, and focus their customer retention efforts.

E3. Limitation

The biggest limitation in our clustering analysis was the process of selecting the value of k. Unlike other analysis techniques, the methods of arriving at the best value of k for the k-means clustering technique is very subjective. One can use the Elbow Method, where the value of k and the inertia are plotted into a graph to visually assess the best value, or by calculating the silhouette score for various values of k. Not having a formulation to determine the best value of k can leave it up to the interpretation of the analyst, which can vary from one analyst to another. Resulting in potentially different conclusions based on who conducted the analysis.

E4. Recommendation

Based on the centroids of each cluster, we recommend that the telecom company focus their efforts on increasing the tenure of customers in Cluster 1 through focused marketing campaigns and targeted incentives. We also recommend that the telecom company ensure that customers in Cluster 2 feel valued, as they have spent on average four more years with the company than customers in Cluster 1. One way to achieve this can be through rewards programs or targeted discounts on additional services.

Part VI. Demonstration

G. Sources

No third-party code or in-text citations were used to complete our research assessment.