*Sspen82@wgu.edu*

OFM3 — OFM3 TASK 1: CLUSTERING TECHNIQUES

DATA MINING II — D212

PRFA — OFM

Sam Spencer

DATA ANALYTICS

STUDENT NUMBER: 000926323

COLLEGE OF INFORMATION TECHNOLOGY, WESTERN GOVERNORS’ UNIVERSITY

NAME OF INSTRUCTOR: Keiona Middleton

Monday, May 15, 2023

Section A: Description of the Report

Section A1:

How do different demographic variables (such as age, gender, income) affect customer churn, and how can K-means clustering be used to identify customer segments with high churn risk based on these interactions?

To answer this question, we could use a clustering technique such as hierarchical clustering or K-means clustering to group customers based on their demographic characteristics (e.g., age, gender, income), as well as their past behavior (e.g., frequency of purchases, time since last purchase, complaints, etc.). The clustering analysis could help identify different segments of customers who are more likely to churn and allow the organization to develop targeted strategies to retain each segment. For example, the organization could offer personalized promotions or discounts to customers who are at higher risk of churn, based on their identified characteristics and preferences.

Section A2: Goal of the data analysis

Our goal is to develop a predictive model to identify customers who are at high risk of churning, based on their past behavior and demographics characteristics, and recommend targeted retention strategies to prevent them from leaving, which in turn would increase revenue.

Part II: Technique Justification

Section B1: How K-Means Clustering Technique Analyzes Churn Dataset

K-means clustering is a type of unsupervised machine learning algorithm used for grouping similar data points in a dataset into a predetermined number of clusters. The K in the name of the algorithm represents the number of clusters. K-means works by iteratively assigning data points to a cluster and then computing the centroid of each cluster. The centroid is the mean of all the points in the cluster. The algorithm continues this process until the centroids no longer change or reach a predetermined number of iterations.

In the context of customer churn analysis, K-means clustering can be used to group customers with similar characteristics into clusters. The algorithm looks for patterns in the customer data and creates clusters based on those patterns. The expected outcome of the analysis is the identification of groups of customers that have similar characteristics such as demographics, usage behavior, and payment history. This information can be used to identify high-risk customers who are likely to churn, as well as opportunities for targeted marketing and retention strategies.

Section B2: Assumption

One assumption of the K-means clustering technique is that the clusters are spherical, equally sized, and evenly distributed. This means that the variance of each cluster is roughly the same and that the data points in each cluster are uniformly distributed. Additionally, the technique assumes that each data point belongs to one or only one cluster. These assumptions can affect the accuracy of the clustering results if the dataset does not meet these criteria.

Section B3: Justification of Chosen packages or Libraries

* NumPy: A fundamental package for scientific computing with support for arrays and mathematical operations.
* pandas: A powerful library for data manipulation and analysis, providing data structures like DataFrame.
* matplotlib.pyplot: A plotting library for creating visualizations, used to plot the elbow plot.
* seaborn: A data visualization library based on matplotlib, often used for statistical graphics.
* sklearn.cluster.KMeans: The KMeans class from scikit-learn used for K-means clustering.
* sklearn.preprocessing.StandardScaler: A utility class from scikit-learn for standardizing features by removing the mean and scaling to unit variance.
* sklearn.metrics.silhouette\_score: A metric used to evaluate the quality of clustering, measuring the compactness and separation of the clusters.
* yellowbrick.cluster.KElbowVisualizer: A visualizer class from the Yellowbrick library for determining the optimal number of clusters using the elbow method.

Part III: Data Preparation

Section C1: Data Preprocessing goal

One relevant data preprocessing goal for K-means clustering is to standardize or normalize the features before running the algorithm. This goal is important because K-means is sensitive to the scale of the variables used in the clustering process.

Standardizing or normalizing the data ensures that each feature contributes equally to the clustering process and prevents variables with larger scales from dominating the clustering results. By applying techniques such as standardization (subtracting the mean and dividing by the standard deviation) or normalization (scaling values to a specified range), we can achieve a more balanced representation of the data and improve the accuracy and reliability of the clustering analysis.

Section C2: Initial Dataset Variables

* Continuous Variables
* Population
* Children
* Age
* Income
* Tenure
* Bandwidth\_GB\_Year

Categorical Variables

* Marital
* Gender
* Churn

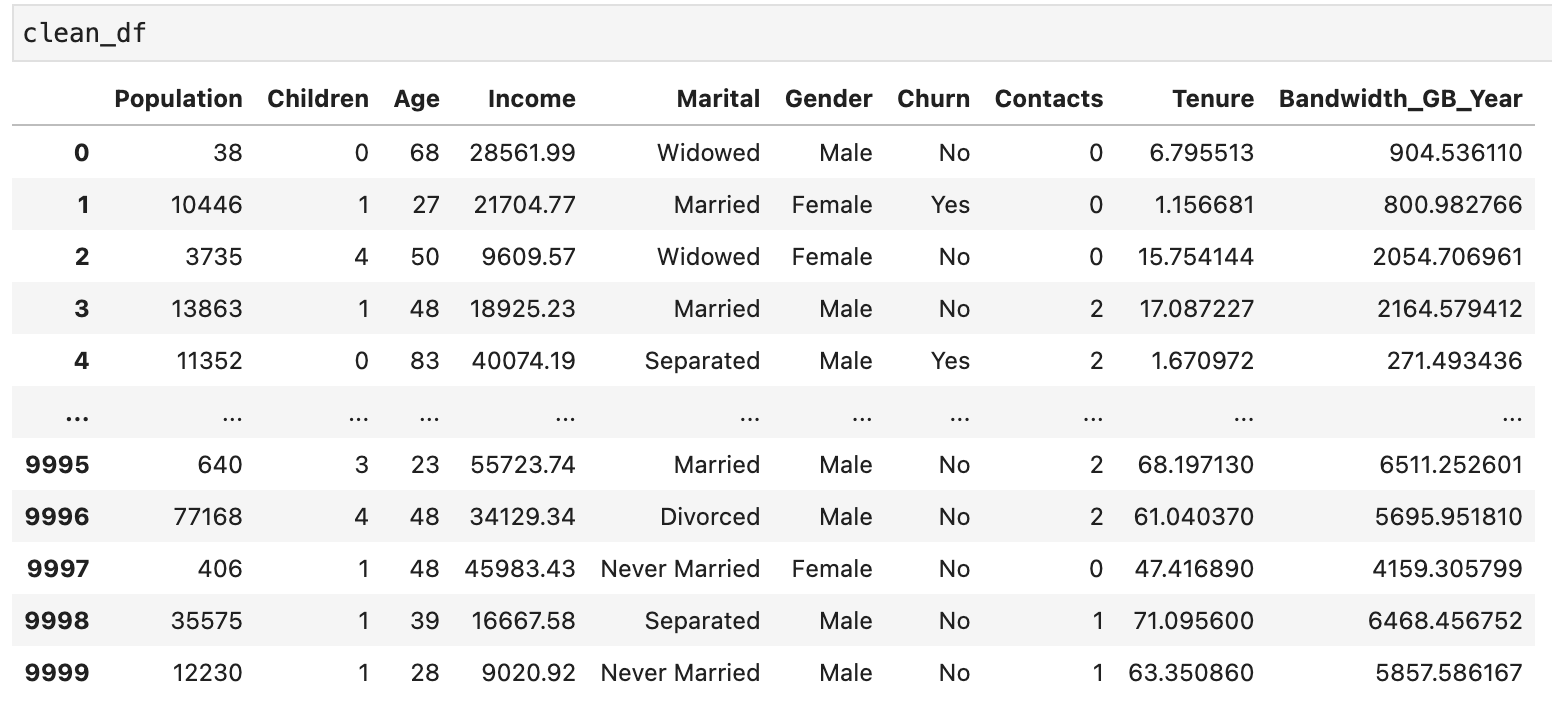


Figure 1 Variables used and the code that produced it.

Section C3: Data Preparation Steps and Codes Used

Preparing data for analysis involves several key steps: data collection, inspection, cleaning, transformation, and integration. These steps ensure that the data is accurate, reliable, and usable for analysis. The code segments for each step may involve functions from libraries like Pandas to read, clean, transform, and merge data. Overall, data preparation is crucial for gaining insights and making informed decisions from data.

To perform our analysis, we have categorical variables that require conversion into numerical format. This step is crucial as it allows us to perform mathematical operations on the data and gain a deeper understanding of the relationships between the variables. By converting these categorical variables into a numerical format, we can transform the data into a more suitable format for analysis, facilitating better interpretation and more meaningful insights.

#converting catergorical values into numerical

# Churn Yes = 1 and No = 0

clean\_df['Churn']=df.Churn.map(dict(Yes=1, No=0))

df['Gender'].value\_counts()

clean\_df['Gender']=df.Gender.map(dict(Female=1, Male=2, Nonbinary=0))

marital\_dict = {'Married': 0, 'Divorced': 1, 'Widowed': 2, 'Separated': 3, 'Never Married': 4}

clean\_df['Marital'] = clean\_df['Marital'].replace(marital\_dict)

A picture containing text, screenshot, font, black and white

Description automatically generated

Figure 2 the code and the dataframe

Section C4: Copy of the Cleaned Dataset

Cleaned dataset is attached with submission.

Part IV: Analysis

Section D1: Description and Screenshots of the Analysis Technique

The analysis technique we used is called KMeans clustering. “K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. It follows a simple procedure of classifying a given data set into several clusters, defined by the letter “k,” which is fixed beforehand. The clusters are then positioned as

points and all observations or data points are associated with the nearest.

cluster, computed, adjusted and then the process starts overusing the new

adjustments until a desired result is reached.” (Rouse, K-means clustering 2016)

The KMeans algorithm works by randomly selecting k initial centroids, where k is the number of clusters specified by the user. Then, for each data point, the algorithm calculates the distance to each centroid and assigns the data point to the cluster with the closest centroid.

Next, the algorithm updates the position of each centroid by calculating the mean of all the data points assigned to that cluster. Then, the process of assigning data points to the closest centroid and updating the centroid positions is repeated until the algorithm converges, meaning that the cluster assignments and centroid positions no longer change significantly.

Once the algorithm has converged, the resulting clusters are determined by the final assignments of the data points to the centroids. The quality of the clustering solution is often evaluated using metrics such as inertia, silhouette score, or Davies-Bouldin index.

The code below (figure 3) performs a loop to run KMeans clustering with a range of different numbers of clusters, from 2 to 20, and calculates the inertia value for each number of clusters. The inertia is the sum of squared distances of each data point to its assigned cluster center. It is a measure of how internally coherent the clusters are. A lower inertia value indicates better clustering.

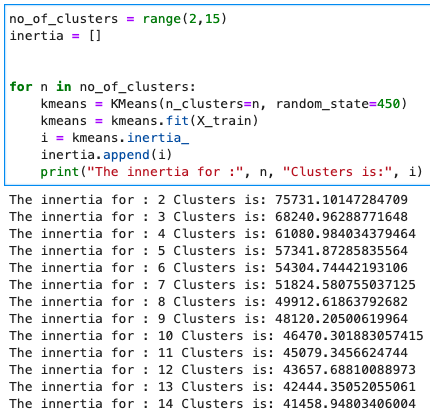


Figure 3

The code below (figure 4) plots an elbow plot to visualize the optimal number of clusters for the given data. An elbow plot is a visualization tool used in clustering analysis to identify the optimal number of clusters for a given dataset. It plots the inertia values, which measure the internal coherence of the clusters, against the number of clusters. The "elbow point" on the plot represents the optimal number of clusters where the inertia value starts to level off. This helps us selecting an appropriate number of clusters to ensure effective clustering of the data.

The `no\_of\_clusters` variable is a range object from 2 to 15. The `inertia` variable contains a list of inertia values for each number of clusters from 2 to 15. These values were likely obtained from running the KMeans algorithm on the data with a range of cluster numbers and storing the inertia values in a list.

The `plt.plot` function is used to create a line plot of the inertia values against the number of clusters. The `marker` argument is set to 'o' to display markers at each data point. The `plt.xlabel`, `plt.ylabel`, and `plt.title` functions are used to add labels to the x-axis, y-axis, and the title of the plot, respectively.

The resulting plot shows a curve that initially decreases steeply and then levels off. The "elbow point" is the point at which the curve starts to level off. In this case, it appears that the optimal number of clusters for the given data is around 4 or 5, as this is the point where the curve starts to level off. This decision on the optimal number of clusters is based on visual inspection of the plot and some domain expertise may also be needed to make a final decision.

import matplotlib.pyplot as plt

no\_of\_clusters = range(2, 15)

inertia = [75731.10147284709, 68240.96288771648, 61080.984034379464, 57341.87285835564, 54304.74442193106, 51824.580755037125, 49912.61863792682, 48120.20500619964,

46470.301883057415, 45079.3456624744, 43657.68810088973, 42444.35052055061, 41458.94803406004]

plt.plot(no\_of\_clusters, inertia, marker='o')

plt.xlabel('Number of Clusters')

plt.ylabel('Inertia')

plt.title('Elbow Plot')

plt.show()

A picture containing text, screenshot, line, plot

Description automatically generated

Figure 4

The code below (figure 5) uses the KElbowVisualizer from the yellowbrick library to plot an elbow plot and determine the optimal number of clusters for the given dataset.

The KElbowVisualizer takes a KMeans model, and a range of cluster numbers to evaluate, and fits the data to the model. It then calculates the inertia value for each number of clusters and plots the results as an elbow plot.

The timings parameter is set to True to display the time taken to fit the model for each number of clusters. The figsize parameter sets the size of the plot. Finally, the show() method is called to display the elbow plot.

A picture containing text, line, diagram, plot

Description automatically generated

Figure 5 the ideal clusters would be 6.

Section D2: The Codes Used to Perform the Clustering TechniqueA screenshot of a computer screen

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with low confidence

Part V: Data Summary and Implications

Section E1: Accuracy

The accuracy of the elbow point at 6 clusters suggests that this number provides a good balance between capturing meaningful patterns in the data and avoiding overfitting or excessive fragmentation.

At the elbow point, the inertia values show a significant reduction compared to the initial clusters, indicating that the clustering algorithm is effectively grouping similar data points together within each cluster. This suggests that the resulting clusters at 6 can capture distinct patterns or segments within the data.

However, it is important to note that the interpretation of the elbow point as the optimal number of clusters is not a definitive measure of accuracy. It is a heuristic approach that relies on the assumption that a significant reduction in inertia corresponds to meaningful cluster separation. The optimal number of clusters can still vary depending on the specific dataset, context, and desired outcome of the analysis.

To further evaluate the accuracy of the clustering at 6 clusters, it is recommended to conduct additional analysis, such as examining the distribution of data points within each cluster, evaluating the coherence and separability of the clusters using silhouette score or other metrics, and assessing the practical usefulness and interpretability of the resulting clusters in the specific domain.

Section E2: Results and Implications

The clustering analysis using KMeans with 6 clusters reveals distinct customer segments based on their characteristics. These segments can be used for targeted marketing, personalized recommendations, and tailored strategies. The analysis provides insights into customer behavior, preferences, and satisfaction, enabling businesses to optimize their efforts. Additionally, the clusters can serve as a baseline for evaluating business performance and predicting future customer behavior. It is important to validate and interpret these results in the specific context of the dataset and business objectives.

Section E3: Limitation

One limitation of the data analysis is that it relies on the assumption that the chosen features and variables adequately capture the underlying patterns and characteristics of the data. If important features are missing or the chosen variables do not sufficiently represent the underlying structure, the clustering results may be suboptimal or misleading. Therefore, it is crucial to carefully select and preprocess the input data to ensure that it adequately represents the underlying phenomenon being studied. Additionally, the interpretation of the clustering results should be done with caution and in consideration of any limitations or biases inherent in the data used for analysis.

Section E4: Recommendation

Our telecom company can enhance the accuracy of its customer variable prediction by conducting numerous tests using different numbers of clusters and diverse datasets. This iterative approach is crucial in obtaining more reliable and robust results.

By experimenting with various numbers of clusters, the organization can assess how the clustering analysis performs under different scenarios. This helps in finding the optimal balance between granularity and interpretability of the customer segments. Evaluation metrics such as inertia, silhouette score, or domain-specific criteria can be utilized to measure the effectiveness of the clustering results.

Moreover, testing the analysis with different datasets allows the organization to examine the generalizability of the identified customer segments. By incorporating diverse datasets that cover various time periods or customer cohorts, the organization gains valuable insights into the stability and consistency of the identified variables.

Part V: Panopto Video presentation

Section F:   
  
You can view the session using the following link:  
<https://wgu.hosted.panopto.com/Panopto/Pages/Sessions/List.aspx#folderID=%222365f8f2-5418-4a6d-9310-adb10176bcae%22>

Section G: Sources

Pandas. pandas. (n.d.). https://pandas.pydata.org/

Rouse, M. (Ed.). (2016, November 30). K-means clustering. Techopedia. https://www.techopedia.com/definition/32057/k-means-clustering

Sklearn.preprocessing.StandardScaler. scikit. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Statistical Data Visualization. seaborn. (n.d.). https://seaborn.pydata.org/

What is NumPy?. What is NumPy? - NumPy v1.24 Manual. (n.d.). https://numpy.org/doc/stable/user/whatisnumpy.html

Yellowbrick: Machine learning visualizationℑ. Yellowbrick: Machine Learning Visualization - Yellowbrick v1.5 documentation. (n.d.). https://www.scikit-yb.org/en/latest/