Customer Segmentation using K-means Clustering Project Report

Amelia Tang

2023/07/30 (updated: 2023-07-30)

Summary

Customer segmentation involves dividing customers into groups that share similar characteristics. The main objective of customer segmentation is to strategize how to engage with customers in each category, ultimately maximizing the profitability of the business from each customer (Tabianan, Velu, and Ravi 2022).

In a 2023 research, a thorough overview of the ACS literature was conducted and `K-means clustering` was identified as the most frequently used algorithm for customer segmentation (Salminen et al. 2023). Other popular algorithms including variations of `K-means clustering` , `K-means with other algorithms` , `Fuzzy Algorithm` and `Latent Class Analysis Model` (Salminen et al. 2023).

This figure shows the popularity rank of algorithms for customer segmentation from the same research.:

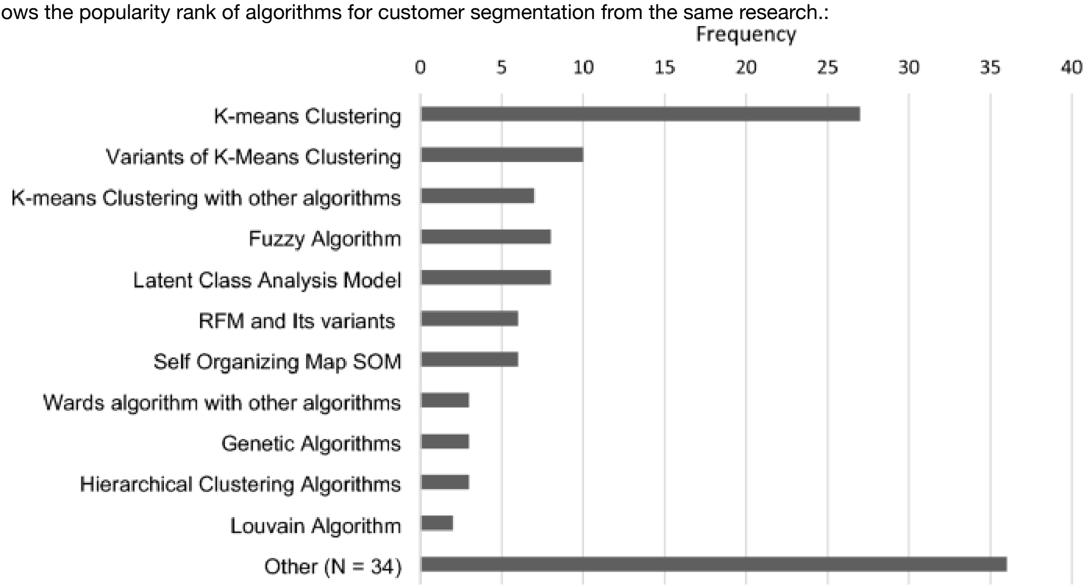


Figure 1. Popular Algorithms for Customer Segmentation

In this project, I implemented K-means clustering in R (R Core Team 2021) and drew an elbow plot to identify the number of cluster `K` using the `factoextra` package (Kassambara 2021). Lastly, I visualized the clusters in a 2-D cluster plot using `clusplot()` in the `cluster` library (Maechler et al. 2022).

Methods

Data Collection and Pre-processing

The data set used in this project was a subset of the `Customer Segmentation` for an automobile company on kaggle.com and available here (https://www.kaggle.com/datasets/vetrirah/customer). Each row of the data represents a used customer and his/her gender, martial information, age, graduation status, profession, work experience, spending score and family size.

To clean the data for EDA, I manipulated the data using `tidyverse` (Wickham et al. 2019). After, removing all the NAs, 6,718 rows of non-null data remained.

To preprocess the data for implementing the K-means clustering model, I created dummy variables using `fastDummies` for categorical features, as the algorithm does not take in categorical data (Jacab Kaplan Year of the package version). Then, I scaled the data using the `scale()` function in base R (R Core Team 2021). By default, it returns scaled data with zero mean and unit variance. Here is the formula for a scaled value:

```
scaled_value = (original_value - mean) / standard_deviation
```

Scaling the data is important for K-means clustering because the algorithm calculates distances between data points to form clusters. When features have different scales, those with larger values can dominate the distance calculation, leading to biased cluster assignments. Scaling ensures that all features contribute equally to the clustering process, resulting in more meaningful and balanced clusters. Additionally, scaling helps K-means converge faster and prevents numerical instability during the optimization process.

Exploratory Data Analysis (EDA)

To conduct exploratory data analysis, I visualized the data using `ggplot2` (**ggplot2?**). The full EDA report can be viewd here (Customer_Segmentation_EDA_Report.pdf)

According to Figure 2, unmarried customers consistently show a low spending score, while married customers display a broader range of spending scores, encompassing high, low, and average levels of spending.

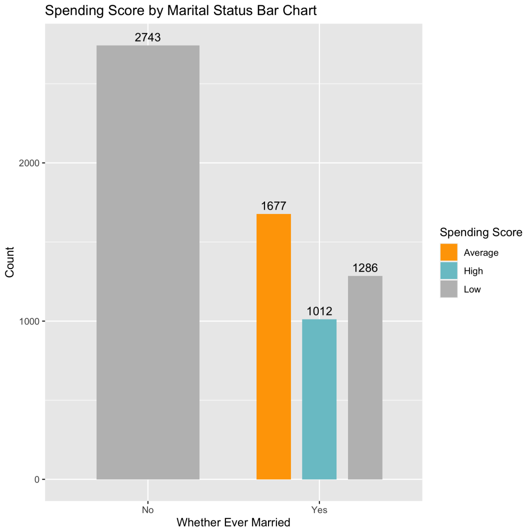


Figure 2. Spending Score by Marital Status

According to Figure 3, the boxplot reveals that a significant portion of healthcare, medical (doctor), and marketing professionals in the dataset are relatively young (below 40 years old). On the other hand, law professionals tend to be relatively older (above 60 years old).

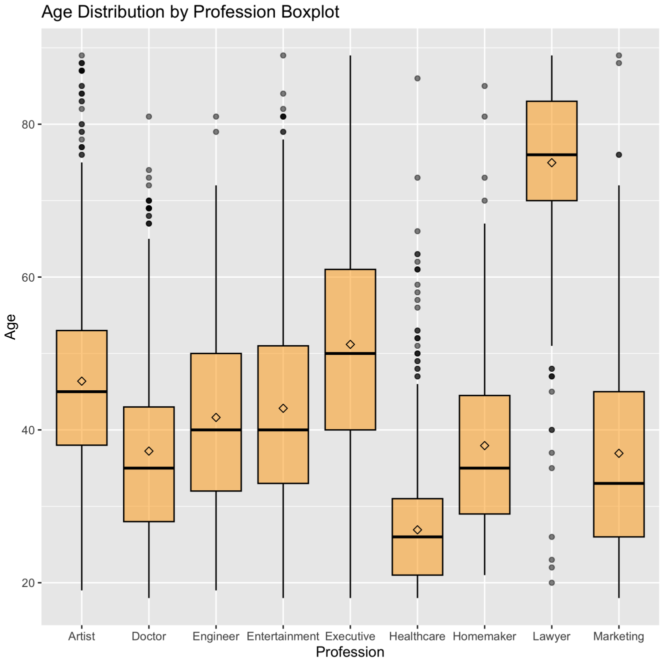


Figure 3. Used Car Prices by Brands

K-means Clustering Algorithm

K-means clustering is a popular unsupervised machine learning algorithm used for clustering data points into K distinct clusters based on their similarity. It aims to partition the data into clusters in a way that minimizes the sum of squared distances between data points and their respective cluster centroids.

One of the most important hyper-parameters for K-means is the number of clusters K. The standard literature utilizes the elbow method to optimize K (Patankar et al. 2021).

The elbow method is a technique used to determine the optimal number of clusters, K, in K-means clustering. It involves plotting the sum of squared distances (SSD) between data points and their assigned cluster centroids for different values of K. As K increases, the SSD typically decreases, as each data point is closer to its cluster centroid. However, beyond a certain point, adding more clusters does not lead to a significant reduction in SSD, and the curve in the plot starts to level off, resembling an elbow. The optimal number of clusters is usually identified at this point, as it represents the “elbow” or the point of diminishing returns. The K value corresponding to the elbow is considered the optimal choice for balancing the trade-off between model complexity and clustering performance, leading to more meaningful and interpretable clusters.

According to the elbow plot for this project shown below, I chose 6 as the number of clusters (K) because when K = 7, the SSD does not decrease significantly, and instead, it increases.

I initially set the maximum number of iterations to 10 give the relatively small data size to see if any meaningful results could be generated for this project.

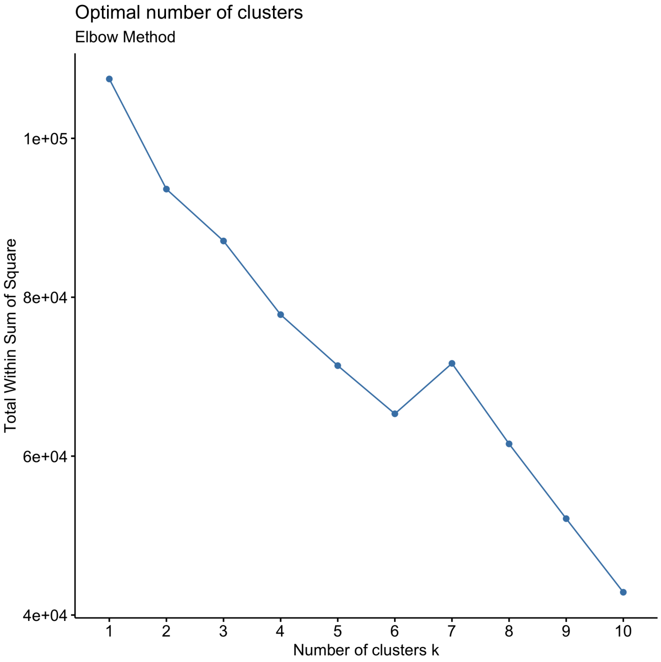
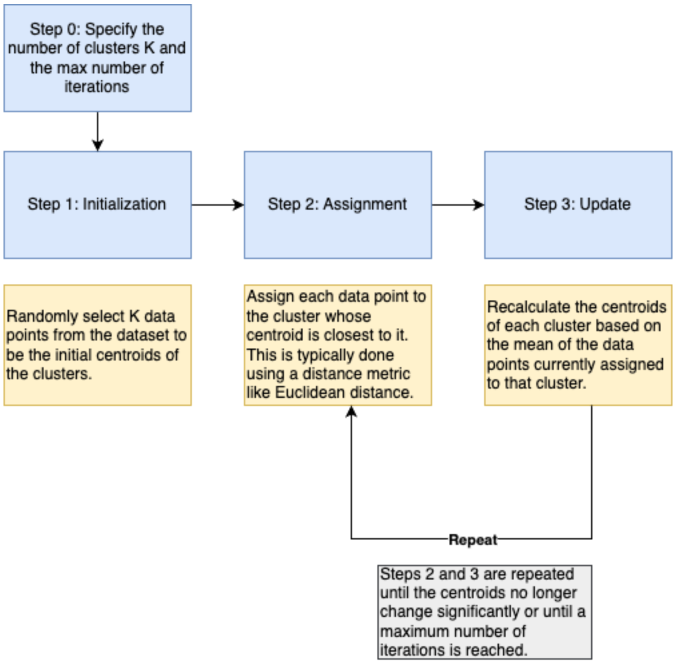


Figure 4. The Elbow Method

The K-means clustering algorithm starts by randomly selecting K data points from the dataset as initial cluster centroids. It then iteratively assigns each data point to the cluster with the closest centroid, using a distance metric like Euclidean distance. After the assignment, the centroids are updated by calculating the mean of the data points currently assigned to each cluster. This process is repeated until the centroids no longer change significantly or until a maximum number of iterations is reached. The algorithm converges when the centroids stabilize, and data points are assigned to the closest clusters based on these final centroids. The flowchart below illustrates how K-means Clustering Algorithms work step by step:


Step 0: Specify the number of clusters K and the max number of iterations
Step 1: Initialization
Step 2: Assignment
Step 3: Update
Repeat

Randomly select K data points from the dataset to be the initial centroids of the clusters.

Assign each data point to the cluster whose centroid is closest to it. This is typically done using a distance metric like Euclidean distance.

Recalculate the centroids of each cluster based on the mean of the data points currently assigned to that cluster.

Steps 2 and 3 are repeated until the centroids no longer change significantly or until a maximum number of iterations is reached.

Figure 5. Kmeans Clustering Algorithm Steps

Results & Discussion

I trained the K-means clustering model with the number of clusters K = 6 and the maximum number of iteration of 10. The below cluster plot demonstrates two major components that explain the clusters.

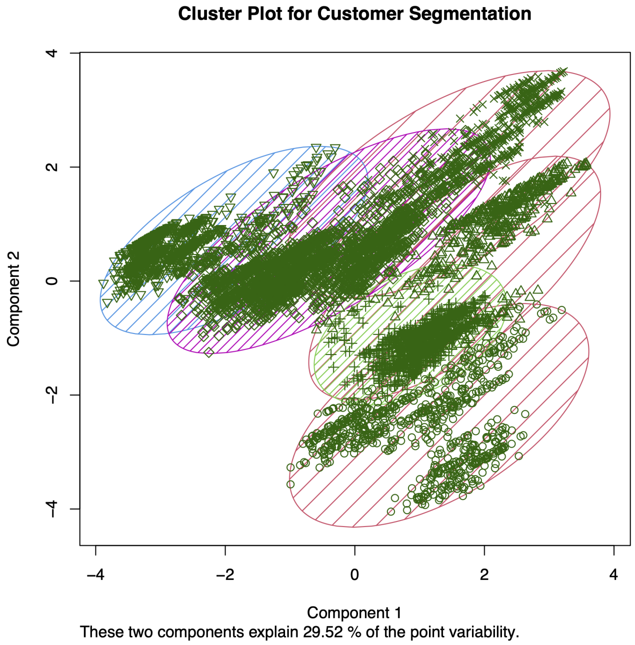


Figure 6. 2-D Cluster Plot

Each customer is assigned a cluster and the results are saved in the `results` folder. Here is a snippet of the cluster assignment:

Age	Work_Experience	Family_Size	Gender_Female	Ever_Married_No	Graduated_No	Profession_Doctor	Profession_Engineer
-1.3034647	-0.4786562	0.7603344	-0.9025373	1.2037141	1.3240158	-0.3114177	-0.3091117
1.4216419	-0.4786562	-1.2090499	1.1078226	-0.8306384	-0.7551656	-0.3114177	3.2345952
1.4216419	-0.7723730	-0.5525885	-0.9025373	-0.8306384	-0.7551656	-0.3114177	-0.3091117
0.7555047	-0.7723730	-0.5525885	-0.9025373	-0.8306384	1.3240158	-0.3114177	-0.3091117
-0.6978855	-0.4786562	0.1038729	-0.9025373	1.2037141	-0.7551656	-0.3114177	-0.3091117
-0.6373275	-0.4786562	0.1038729	1.1078226	1.2037141	-0.7551656	-0.3114177	-0.3091117

Profession_Entertainment	Profession_Executive	Profession_Healthcare	Profession_Homemaker	Profession_Lawyer	Profession_Marke
-0.3715439	-0.2862963	2.2733660	-0.1649639	-0.2844665	-0.1895
-0.3715439	-0.2862963	-0.4398109	-0.1649639	-0.2844665	-0.1895
-0.3715439	-0.2862963	-0.4398109	-0.1649639	3.5148294	-0.1895
-0.3715439	-0.2862963	-0.4398109	-0.1649639	-0.2844665	-0.1895
-0.3715439	-0.2862963	2.2733660	-0.1649639	-0.2844665	-0.1895
-0.3715439	-0.2862963	2.2733660	-0.1649639	-0.2844665	-0.1895

Spending_Score_Average	Spending_Score_High	clusters
-0.5767344	-0.4211066	6
-0.5767344	-0.4211066	1
-0.5767344	2.3743421	4

Spending_Score_Average	Spending_Score_High	clusters
1.7336423	-0.4211066	3
-0.5767344	-0.4211066	6
-0.5767344	-0.4211066	6

References

Jacab Kaplan, Benhamin Schlegel. Year of the package version. *fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from Categorical Variables*. https://github.com/jacobkap/fastDummies (https://github.com/jacobkap/fastDummies).

Kassambara, Alboukadel. 2021. *Factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. https://CRAN.R-project.org/package=factoextra (https://CRAN.R-project.org/package=factoextra).

Maechler, Martin, Peter Rousseeuw, Anja Struyf, Mia Hubert, and Kurt Hornik. 2022. *Cluster: Cluster Analysis Basics and Extensions*. https://CRAN.R-project.org/package=cluster (https://CRAN.R-project.org/package=cluster).

Patankar, Nikhil, Soham Dixit, Akshay Bhamare, Ashutosh Darpel, and Ritik Raina. 2021. “Customer Segmentation Using Machine Learning.” IOS Press; released under the conditions of Creative Commons Attribution ….

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/ (https://www.R-project.org/).

Salminen, Joni, Mekhail Mustak, Muhammad Sufyan, and Bernard J. Jansen. 2023. “How Can Algorithms Help in Segmenting Users and Customers? A Systematic Review and Research Agenda for Algorithmic Customer Segmentation.” *Journal of Marketing Analytics*. https://doi.org/10.1057/s41270-023-00235-5 (https://doi.org/10.1057/s41270-023-00235-5).

Tabianan, Kayalvily, Shubashini Velu, and Vinayakumar Ravi. 2022. “K-Means Clustering Approach for Intelligent Customer Segmentation Using Customer Purchase Behavior Data.” *Sustainability* 14 (12). https://doi.org/10.3390/su14127243 (https://doi.org/10.3390/su14127243).

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, arrett Grolemund, et al. 2019. “Welcome to the tidyverse.” *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686 (https://doi.org/10.21105/joss.01686).