Customer Segmentation Using K-means Clustering

Miftahul Labiib Syam

2023-07-27

In this notebook, we will perform customer segmentation using the K-Means clustering algorithm. Customer segmentation is a powerful technique that allows businesses to divide their customer base into distinct groups based on their shared characteristics. This helps in better understanding customer behavior and tailoring marketing strategies to target specific segments more effectively.

1. Dataset

Dataset that we use in this project is collected from kaggle which contains the following columns:

1. `Customer ID`: A unique identifier for each customer.
2. `Gender`: The gender of the customer.
3. `Age`: The age of the customer.
4. `Annual Income`: The annual income of the customer.
5. `Spending Score`: Score assigned by the shop, based on customer behavior and spending nature.
6. `Profession`: The profession of the customer.
7. `Work Experience`: The number of years of work experience of the customer.
8. `Family Size`: The size of the customer's family.

2. Read Data

```
#read data
customer <- read.csv("data_input/customers.csv")
head(customer)
```

```
##   CustomerID Gender Age Annual.Income.... Spending.Score..1.100.    Profession
## 1          1   Male  19             15000                     39    Healthcare
## 2          2   Male  21             35000                     81      Engineer
## 3          3 Female  20             86000                      6      Engineer
## 4          4 Female  23             59000                     77        Lawyer
## 5          5 Female  31             38000                     40 Entertainment
## 6          6 Female  22             58000                     76        Artist
##   Work.Experience Family.Size
## 1               1           4
## 2               3           3
## 3               1           1
## 4               0           2
## 5               2           6
## 6               0           2
```

After load the dataset, we have to check the datatype of each columns to ensure that our data is stored in appropriate type. We will use `glimpse()` from `dplyr`:

```
library(dplyr)
glimpse(customer)
```

```
## Rows: 2,000
## Columns: 8
## $ CustomerID             <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
## $ Gender                 <chr> "Male", "Male", "Female", "Female", "Female", "~
## $ Age                    <int> 19, 21, 20, 23, 31, 22, 35, 23, 64, 30, 67, 35,~
## $ Annual.Income....      <int> 15000, 35000, 86000, 59000, 38000, 58000, 31000~
## $ Spending.Score..1.100. <int> 39, 81, 6, 77, 40, 76, 6, 94, 3, 72, 14, 99, 15~
## $ Profession             <chr> "Healthcare", "Engineer", "Engineer", "Lawyer",~
## $ Work.Experience        <int> 1, 3, 1, 0, 2, 0, 1, 1, 0, 1, 1, 4, 0, 1, 0, 1,~
## $ Family.Size            <int> 4, 3, 1, 2, 6, 2, 3, 3, 3, 4, 3, 4, 5, 1, 1, 2,~
```

- `Gender` and `Profession` is stored in character and we will convert it to factor.
- `CustomerID` is an identifier and we will convert it to character.

2.1 Change Datatype

```
customer <- customer %>%
  mutate_at(vars(Gender, Profession), as.factor) %>%
  mutate(CustomerID = as.character(CustomerID))
glimpse(customer)
```

```
## Rows: 2,000
## Columns: 8
## $ CustomerID             <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "1~
## $ Gender                 <fct> Male, Male, Female, Female, Female, Female, Fem~
## $ Age                    <int> 19, 21, 20, 23, 31, 22, 35, 23, 64, 30, 67, 35,~
## $ Annual.Income....      <int> 15000, 35000, 86000, 59000, 38000, 58000, 31000~
## $ Spending.Score..1.100. <int> 39, 81, 6, 77, 40, 76, 6, 94, 3, 72, 14, 99, 15~
## $ Profession             <fct> Healthcare, Engineer, Engineer, Lawyer, Enterta~
## $ Work.Experience        <int> 1, 3, 1, 0, 2, 0, 1, 1, 0, 1, 1, 4, 0, 1, 0, 1,~
## $ Family.Size            <int> 4, 3, 1, 2, 6, 2, 3, 3, 3, 4, 3, 4, 5, 1, 1, 2,~
```

2.2 Check missing values and Duplicate

```
colSums(is.na(customer))
```

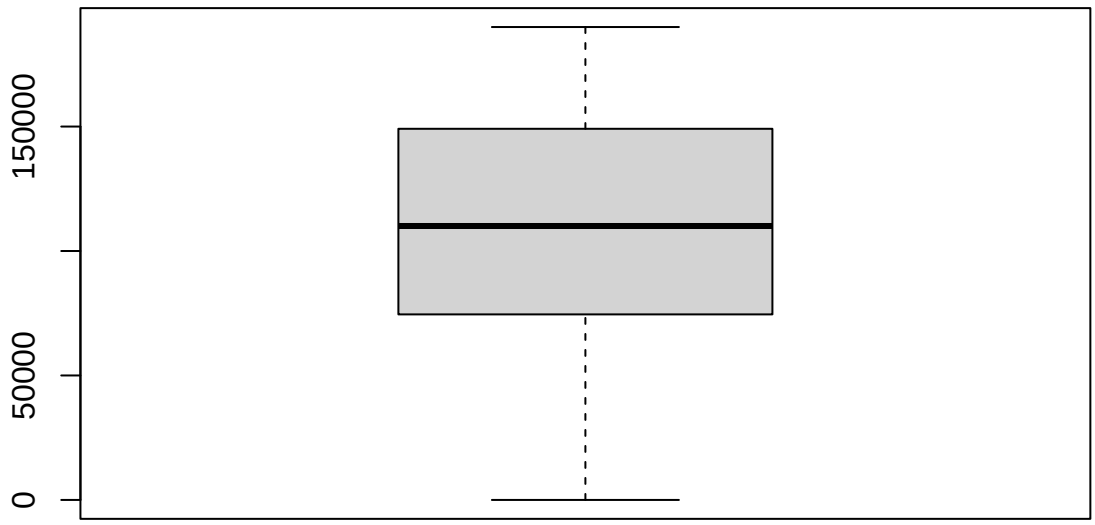
```
##             CustomerID                 Gender                    Age
##                      0                      0                      0
##      Annual.Income.... Spending.Score..1.100.             Profession
##                      0                      0                      0
##        Work.Experience            Family.Size
##                      0                      0
```

```
sum(duplicated(customer))
```

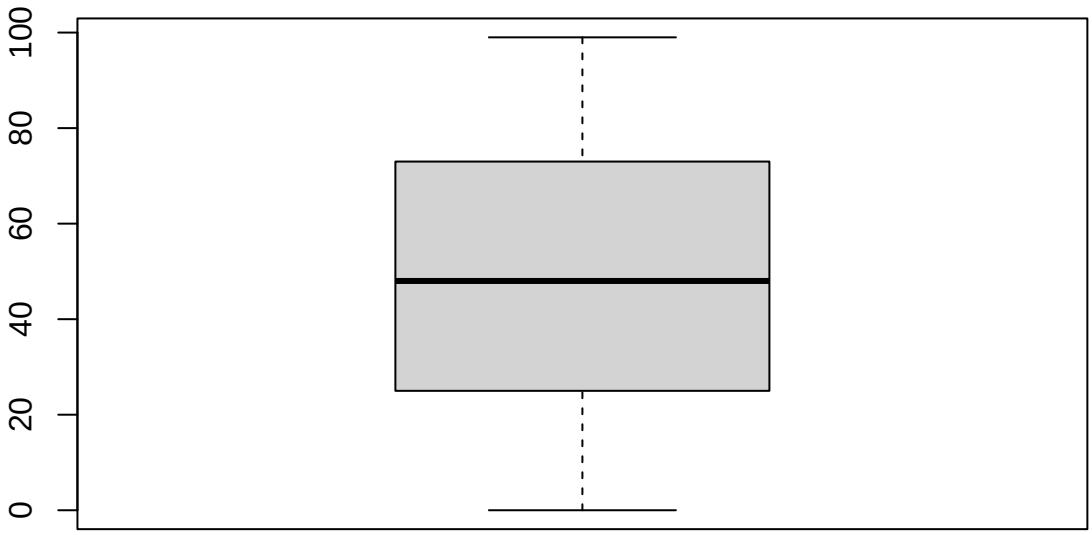
```
## [1] 0
```

3. EDA

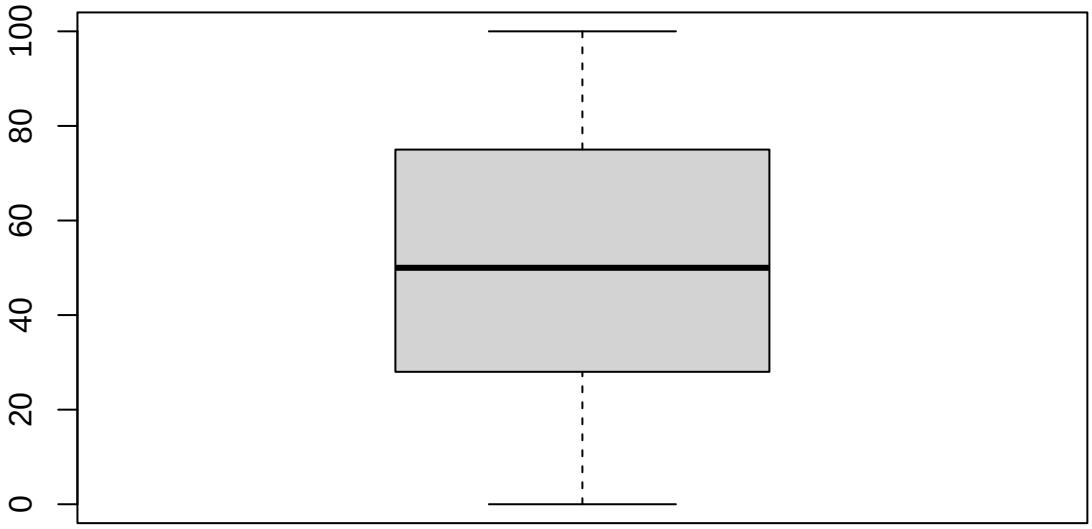
```
boxplot(customer$Annual.Income....)
```



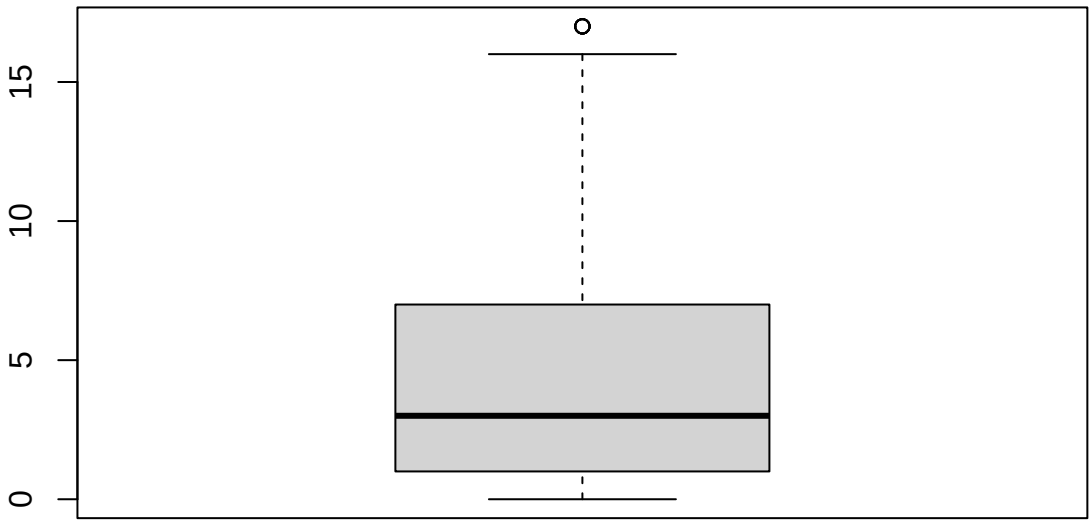

```
boxplot(customer$Age)
```




```
boxplot(customer$Spending.Score..1.100.)
```




```
work_experience <- boxplot(customer$Work.Experience)
```



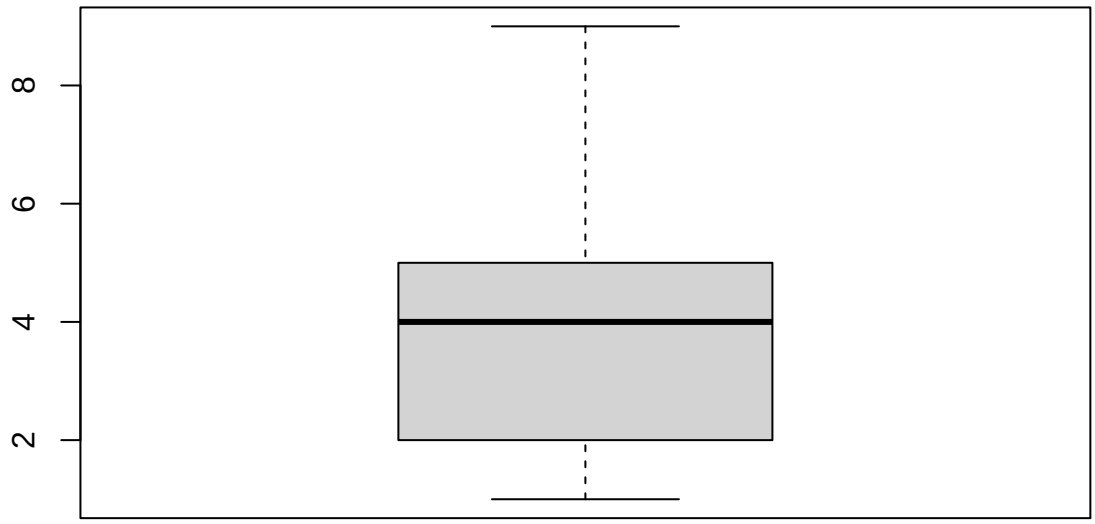
There is one outlier in this column, let's check it out

```
work_experience$out
```

```
## [1] 17 17 17 17 17
```

We have to remember that outlier has big influence in k-means clustering, so it is very important to determine what step we will do to handle the otlier. In this case, we will not remove it because we think that this outlier is the real data.

```
boxplot(customer$Family.Size)
```




```
unique(customer$Profession)
```

```
##  [1] Healthcare    Engineer      Lawyer        Entertainment Artist
##  [6] Executive     Doctor        Homemaker     Marketing
## 10 Levels:  Artist Doctor Engineer Entertainment Executive ... Marketing
```

4. Scaling data

Scaling data is crucial when performing k-means clustering because the algorithm calculates distances between data points to form clusters based on their proximity. If the features in the data have different scales, those with larger magnitudes will dominate the clustering process. As a result, features with smaller scales may be ignored, leading to suboptimal or biased clustering results. By scaling the data, we ensure that all features contribute equally to the clustering process, improving the accuracy and fairness of the final clusters produced by k-means.

```
customer_scaled <- customer %>%
  mutate(across(where(is.numeric), scale))
head(customer_scaled)
```

```
##   CustomerID Gender        Age Annual.Income.... Spending.Score..1.100.
## 1          1   Male -1.0538258        -2.0929775             -0.4282314
## 2          2   Male -0.9834769        -1.6557190              1.0752771
## 3          3 Female -1.0186513        -0.5407099             -1.6095596
```

```
## 4          4 Female -0.9131281        -1.1310089              0.9320858
## 5          5 Female -0.6317327        -1.5901303             -0.3924336
## 6          6 Female -0.9483025        -1.1528718              0.8962880
##      Profession Work.Experience Family.Size
## 1    Healthcare      -0.7910093   0.1174681
## 2      Engineer      -0.2810919  -0.3899534
## 3      Engineer      -0.7910093  -1.4047962
## 4        Lawyer      -1.0459680  -0.8973748
## 5 Entertainment      -0.5360506   1.1323109
## 6        Artist      -1.0459680  -0.8973748
```

5. Clustering Using Two Features : Annual Income and Spending Score

in the first part of our analysis, we will consider only two features - Annual Income and Spending Score - to perform customer segmentation. This will help us visualize the clusters in a 2-dimensional space and gain insights into basic customer behaviors.

```
#subset data
customer_subset1 <- customer_scaled %>%
  select(Annual.Income...., Spending.Score..1.100.)
```

```
set.seed(100)
customer_cluster <- kmeans(x= customer_subset1,
                           centers = 4)
```

```
customer$cluster <- as.factor(customer_cluster$cluster)
clustering <- customer %>%
  group_by(cluster) %>%
  summarize(average_income = mean(Annual.Income....),
            average_spending_score = mean(Spending.Score..1.100.))


clustering
```

```
## # A tibble: 4 x 3
##   cluster average_income average_spending_score
##   <fct>            <dbl>                 <dbl>
## 1 1              151946.                  30.0
## 2 2               72313.                  25.3
## 3 3              148196.                  77.9
## 4 4               72443.                  72.3
```

```
#Return centroid to the real value
mean_inc <- mean(customer$Annual.Income....)
mean_spend <- mean(customer$Spending.Score..1.100.)
sd_inc <- sd(customer$Annual.Income....)
sd_spend <- sd(customer$Spending.Score..1.100.)
centroid <- customer_cluster$centers %>%
  as.data.frame() %>%
  summarize(centroid_income = Annual.Income....*sd_inc+mean_inc,
```

```
            centroid_spend = Spending.Score..1.100.*sd_spend+mean_spend)
#Buat Scatter Plot untuk memvisualisasikan hasil clustering
library(ggplot2)
plot_cluster <- customer %>%
  ggplot(mapping = aes(x = Annual.Income...., y = Spending.Score..1.100.)) +
  geom_point(aes(color = cluster)) +
  geom_point(data = centroid, aes(x = centroid_income, y = centroid_spend, color=  "Centroid"),
             size = 5)+
  labs(title = "Cluster",
    x= "Annual Income",
       y= "Spending Score(1-100)",
    color= "Legend")

plot_cluster
```

Conclusion:

1. Cluster 1:

- Average Annual Income: $71671.20
- Average Spending SCore: 74.01452 This cluster represents customers with low annual income and high spending score. These customers may be categorized as the "*low-income, high-spending group*".

2. Cluster 2:

- Average Annual Income: $147967.79
- Average Spending SCore: 77.89936 This cluster represents customers with high annual income and high spending score. These customers may be categorized as the "*high-income, high-spending group*".

3. Cluster 3:

- Average Annual Income: $152400.37
- Average Spending SCore: 29.99806 This cluster represents customers with high annual income and low spending score. These customers may be categorized as the "*high-income, low-spending group*".

4. Cluster 4:

- Average Annual Income: $73230.99
- Average Spending SCore: 26.90093 This cluster represents customers with low annual income and low spending score. These customers may be categorized as the "*low-income, low-spending group*".

Analyzing these clusters allows you to gain insights into the diverse spending behaviors among customer groups, enabling customized marketing approaches. Visualizations like scatter plots or bar charts aid in comprehending cluster distributions and inter-feature relationships. Remember, the conclusions are drawn from mean cluster values, and individual profiles may differ. Further data exploration provides deeper understanding of customer segments and their traits. Now, we'll cluster customers based on 'Age' to uncover insights into behavior and preferences, optimizing marketing strategies and product offerings.

6. Clustering Using Three Features : Annual Income, Spending Score, and Age

```
customer_subset2 <- customer_scaled %>%
  select(Annual.Income...., Spending.Score..1.100., Age)
head(customer_subset2)
```

```
##   Annual.Income.... Spending.Score..1.100.        Age
## 1        -2.0929775             -0.4282314 -1.0538258
## 2        -1.6557190              1.0752771 -0.9834769
## 3        -0.5407099             -1.6095596 -1.0186513
## 4        -1.1310089              0.9320858 -0.9131281
## 5        -1.5901303             -0.3924336 -0.6317327
## 6        -1.1528718              0.8962880 -0.9483025
```

```
set.seed(100)
customer_six_cluster <- kmeans(x = customer_subset2, centers = 6)
```

```
customer_six_cluster$centers
```

```
##   Annual.Income.... Spending.Score..1.100.        Age
## 1        -0.1181409             -0.7736295  1.0786351
## 2        -0.8922092             -0.9029084 -0.6112949
## 3         0.6513948              1.0249151 -1.0411202
## 4        -1.0259115              0.8367486 -0.1380150
## 5         0.9869072             -0.7756385 -0.6793008
## 6         0.7907362              0.6880481  0.9202999
```

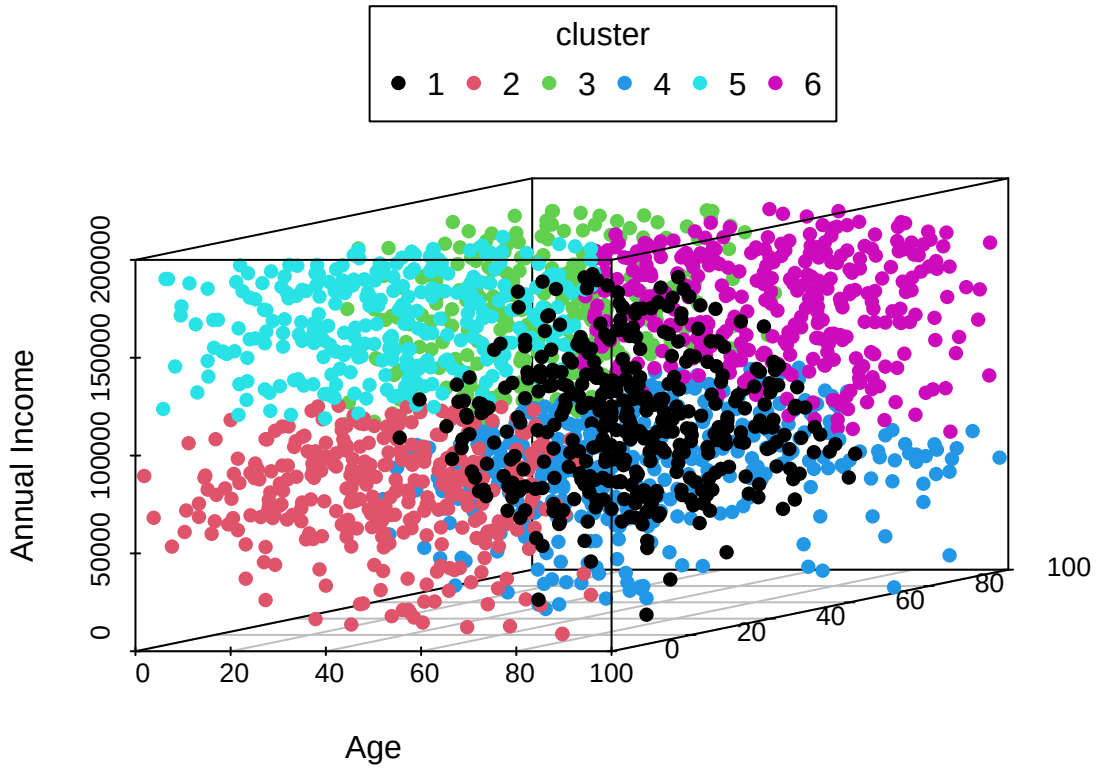
```
customer_six_cluster$betweenss/customer_six_cluster$totss
```

```
## [1] 0.6602934
```

```
customer$cluster_2 <- as.factor(customer_six_cluster$cluster)
```

```
#library(rgl)
mean_age <- mean(customer$Age)
sd_age<- sd(customer$Age)
centroid_cluster <- customer_six_cluster$centers %>%
  as.data.frame() %>%
  summarize(centroid_income = Annual.Income....*sd_inc+mean_inc,
            centroid_spend = Spending.Score..1.100.*sd_spend+mean_inc,
            centroid_age = Age*sd_age+mean_age)
```

```
library(scatterplot3d)
plot_cluster_six <- plot_cluster_six <- scatterplot3d(
  x = customer$Age,
  y = customer$Spending.Score..1.100.,
  z = customer$Annual.Income....,
  pch=16,
  color = customer$cluster_2,
  xlab = "Age",
  ylab = "Spending Score(1-100)",
  zlab = "Annual Income",
  angle = 15,
  grid = T,
  box = T
)
legend("top", legend = levels(customer$cluster_2), col = levels(customer$cluster_2), pch = 16, title =
```




```
clustering_2 <- customer %>%
 group_by(cluster_2)%>%
  summarize(avg_income = mean(Annual.Income....),
            avg_spending = mean(Spending.Score..1.100.),
            avg_age = mean(Age)) %>%
  rename(cluster = cluster_2)
clustering_2
```

```
## # A tibble: 6 x 4
##   cluster avg_income avg_spending avg_age
##   <fct>        <dbl>        <dbl>   <dbl>
## 1 1          105328.         29.4    79.6
## 2 2           69923.         25.7    31.6
## 3 3          140526.         79.6    19.4
## 4 4           63807.         74.3    45.0
## 5 5          155872.         29.3    29.6
## 6 6          146900.         70.2    75.1
```

Conclusion: After applying clustering to the customer dataset, we obtained six distinct clusters based on their characteristics. Each cluster shows different patterns of behavior among the customers. Here are the key conclusions based on the average values of each cluster:

1. Cluster 1 (Senior Frugal Spenders):

- Customers in this cluster have a moderate annual income.

- They exhibit a conservative spending behavior with a low spending score.
- The average age of the customers is the highest among all clusters, indicating a predominantly senior population.

2. Cluster 2 (Young Frugal Earners):

- Customers in this cluster have a low annual income.
- They are prudent spenders, with the lowest average spending score among all clusters.
- The average age of the customers is relatively young, suggesting a focus on younger individuals or young earners.

3. Cluster 3 (High-Spending Young Consumers):

- Customers in this cluster have a relatively high annual income.
- They are enthusiastic spenders, reflected by the highest spending score among all clusters.
- The average age of the customers is the youngest compared to all clusters, indicating a group of young and affluent consumers.

4. Cluster 4 (Middle-Aged Moderate Spenders):

- Customers in this cluster have the lowest annual income among all clusters.
- They display a balanced spending behavior with a relatively high spending score.
- The average age of the customers is moderate, suggesting a middle-aged demographic.

5. Cluster 5 (Affluent Young Savers):

- Customers in this cluster have the highest average annual income among all clusters.
- They are cautious spenders, indicated by their relatively low spending score.
- The customers' age is relatively young, implying a group of affluent and financially responsible young individuals.

6. Cluster 6 (Affluent Senior Shoppers):

- Customers in this cluster have a relatively high annual income.
- They are enthusiastic shoppers, as evidenced by their high spending score.
- The average age is relatively old, suggesting an affluent and older customer segment.

These insights provide valuable information for marketing and business strategies. The clustering results can help target specific customer segments, tailor products or services to meet their needs, and optimize marketing efforts to maximize revenue and customer satisfaction. Additionally, the clusters can serve as a basis for further analysis and exploration of customer behavior and preferences.