****

**VIRGINIA COMMONWEALTH UNIVERSITY**

**Statistical analysis and modelling (SCMA 632)**

**A4- Multivariate Analysis and Business Analytics Applications**

**Part-2**

**Bala Vignesh Aravindan**

**V01106579**

**Date of Submission: 07-07-2024**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Title** | **Page No.** |
| **1.** | Introduction | **1** |
| **2** | Results | **2-3** |
| **3.** | Interpretations | **3** |
| **4.** | Recommendations | **3** |
| **5.** | Codes | **3-11** |

**Cluster Analysis to characterize respondents based on background variables**

**Introduction**

The purpose of this study is to segment a dataset of individuals based on their demographic and preference characteristics. The dataset consists of 33 features, including income, age, proximity to various amenities, and other factors. By applying KMeans clustering, we aim to identify distinct groups within the population and understand their characteristics.

**Objectives**

The objectives of this study are:

* To identify distinct clusters within the dataset based on demographic and preference characteristics
* To describe the characteristics of each cluster
* To understand the business significance of each cluster
* To provide recommendations for targeting each cluster effectively

**Business Significance**

* Segmenting the population into distinct clusters can have significant business implications.
* By understanding the characteristics of each cluster, businesses can tailor their marketing strategies, product offerings, and customer service to meet the specific needs of each group.
* This can lead to increased customer satisfaction, loyalty, and ultimately, revenue growth.

**Results**

The KMeans clustering algorithm was applied to the dataset, and three distinct clusters were identified. The cluster centroids are presented in the table above.

**Interpretation**

Here's a brief interpretation of the results:

**Cluster 0:**

Higher income (185,526) and budgets (117.76)

Older age group (53.24)

Higher proximity to city (4.05), schools (3.89), transport (4.16), and work place (3.84)

Higher appreciation potential (4.32) and profile of neighbourhood (4.47)

Larger size (1894.74) and higher EMI (72,105.26)

**Cluster 1:**

Middle-income group (75,000) and budgets (52.33)

Middle-aged group (44.38)

Moderate proximity to city (3.7), schools (3.5), transport (4.0), and work place (3.97)

Moderate appreciation potential (4.27) and profile of neighbourhood (3.83)

Medium size (946.67) and EMI (42,666.67)

**Cluster 2:**

Lower-income group (55,000) and budgets (32.5)

Younger age group (36.19)

Lower proximity to city (3.14), schools (2.95), transport (4.1), and work place (3.67)

Lower appreciation potential (3.90) and profile of neighbourhood (3.29)

Smaller size (666.67) and EMI (27,500)

These clusters seem to represent different segments of the population based on their income, age, and preferences for proximity to various amenities. Cluster 0 appears to be the most affluent group, while Cluster 2 is the most budget-conscious. Cluster 1 falls in between, representing a middle-class segment.The three clusters can be interpreted as follows:

**Cluster 0:** Affluent individuals with high income, older age, and a preference for proximity to city centers, schools, and workplaces. They have a high appreciation potential and value a good profile of neighborhood.

**Cluster 1:** Middle-class individuals with moderate income, middle age, and a moderate preference for proximity to various amenities. They have a moderate appreciation potential and value a good profile of neighborhood.

**Cluster 2:** Budget-conscious individuals with low income, younger age, and a lower preference for proximity to various amenities. They have a lower appreciation potential and value affordability.

**Recommendations**

Based on the clustering results, we recommend the following:

* For Cluster 0: Target high-end products and services, emphasize the proximity to city centres, schools, and workplaces, and highlight the appreciation potential and good profile of neighbourhood.
* For Cluster 1: Offer mid-range products and services, emphasize the moderate proximity to various amenities, and highlight the moderate appreciation potential and good profile of neighbourhood.
* For Cluster 2: Offer affordable products and services, emphasize the value for money, and highlight the convenience and accessibility of the location.

By targeting each cluster effectively, businesses can increase their market share, customer satisfaction, and revenue growth.

**Codes:**

**import** pandas **as** pd

**import** numpy **as** np

**import** category\_encoders **as** ce

**from** sklearn.cluster **import** KMeans

**from** sklearn.preprocessing **import** StandardScaler

In [4]:

df **=** pd**.**read\_csv("C:\\Users\\Bala Vignesh.A\\Desktop\\SCMA 632\\Survey (1).csv")

In [6]:

print(df**.**columns)

# Preprocess the data

In [11]:

encoder **=** ce**.**OrdinalEncoder(cols**=**['City', 'Sex', 'Occupation', 'Monthly Household Income',

'Income', 'Planning to Buy a new house', 'Time Frame',

'Reasons for buying a house', 'what type of House', 'Number of rooms',

'Size of House', 'Budget', 'Finished/Semi Finished',

'Influence Decision', 'Maintainance', 'EMI', '1.Proximity to city',

'2.Proximity to schools', '3. Proximity to transport',

'4. Proximity to work place', '5. Proximity to shopping',

'1. Gym/Pool/Sports facility', '2. Parking space', '3.Power back-up',

'4.Water supply', '5.Security', '1. Exterior look ', '2. Unit size',

'3. Interior design and branded components',

'4. Layout plan (Integrated etc.)', '5. View from apartment',

'1. Price', '2. Booking amount', '3. Equated Monthly Instalment (EMI)',

'4. Maintenance charges', '5. Availability of loan',

'1. Builder reputation', '2. Appreciation potential',

'3. Profile of neighbourhood', '4. Availability of domestic help',

'Time', 'Size', 'Budgets', 'Maintainances', 'EMI.1', 'ages', 'sex',

'Finished/Semi Finished.1', 'Influence Decision.1'])

df\_encoded **=** encoder**.**fit\_transform(df)

# Select background variables

In [13]:

background\_vars **=** df\_encoded**.**iloc[:, **-**20:]

# Scale the data

In [14]:

scaler **=** StandardScaler()

background\_vars\_scaled **=** scaler**.**fit\_transform(background\_vars)

# Determine the optimal number of clusters using the elbow method

In [15]:

wcss **=** []

**for** k **in** range(1, 11):

kmeans **=** KMeans(n\_clusters**=**k, random\_state**=**42)

kmeans**.**fit(background\_vars\_scaled)

wcss**.**append(kmeans**.**inertia\_)

**import** matplotlib.pyplot **as** plt

In [18]:

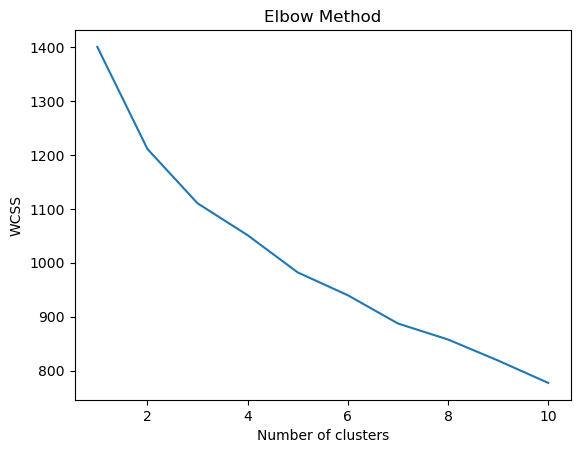
plt**.**plot(range(1, 11), wcss)

plt**.**title('Elbow Method')

plt**.**xlabel('Number of clusters')

plt**.**ylabel('WCSS')

plt**.**show()

****

# Perform K-means clustering with 3 clusters

In [19]:

kmeans **=** KMeans(n\_clusters**=**3, random\_state**=**42)

clusters **=** kmeans**.**fit\_predict(background\_vars\_scaled)

# Add the cluster labels to the original dataset

In [20]:

df['Cluster'] **=** clusters

# Analyze and interpret clusters

In [23]:

**def** numeric\_mean(group):

**return** group**.**select\_dtypes(include**=**[np**.**number])**.**mean()

print(df**.**groupby('Cluster')**.**apply(numeric\_mean))

Sex Income 1.Proximity to city 2.Proximity to schools \

Cluster

0 NaN 185526.315789 4.052632 3.894737

1 NaN 75000.000000 3.700000 3.500000

2 NaN 55000.000000 3.142857 2.952381

3. Proximity to transport 4. Proximity to work place \

Cluster

0 4.157895 3.842105

1 4.000000 3.966667

2 4.095238 3.666667

5. Proximity to shopping 1. Gym/Pool/Sports facility \

Cluster

0 3.052632 4.052632

1 2.566667 2.966667

2 2.333333 2.904762

2. Parking space 3.Power back-up ... 2. Appreciation potential \

Cluster ...

0 3.842105 3.736842 ... 4.315789

1 3.566667 3.366667 ... 4.266667

2 3.190476 3.476190 ... 3.904762

3. Profile of neighbourhood 4. Availability of domestic help \

Cluster

0 4.473684 3.789474

1 3.833333 3.000000

2 3.285714 2.761905

Time Size Budgets Maintainances EMI.1 \

Cluster

0 7.421053 1894.736842 117.763158 71578.947368 72105.263158

1 7.500000 946.666667 52.333333 31004.000000 42666.666667

2 7.000000 666.666667 32.500000 17619.047619 27500.000000

ages Cluster

Cluster

0 53.236842 0.0

1 44.383333 1.0

2 36.190476 2.0

[3 rows x 33 columns]

# Visualize the clusters

In [37]:

**import** pandas **as** pd

**from** sklearn.cluster **import** KMeans

In [38]:

**from** sklearn.preprocessing **import** StandardScaler

In [40]:

**def** extract\_median\_age(age\_range):

ages **=** age\_range**.**split('-')

**return** np**.**median([int(age) **for** age **in** ages])

In [42]:

**def** extract\_median\_age(age\_range):

**if** age\_range **==** '>60':

**return** 65 *# assume the median age for '>60' is 65*

**elif** '-' **in** age\_range:

ages **=** age\_range**.**split('-')

**return** np**.**median([int(age) **for** age **in** ages])

**else**:

**return** int(age\_range)

In [43]:

df['Age'] **=** df['Age']**.**apply(extract\_median\_age)

# Perform KMeans clustering on the preprocessed 'Age' column

In [44]:

kmeans **=** KMeans(n\_clusters**=**5)

cluster\_labels **=** kmeans**.**fit\_predict(df[['Age']])

# reate a new DataFrame with the cluster labels and ages

In [45]:

cluster\_df **=** pd**.**DataFrame({'Cluster': cluster\_labels, 'Age': df['Age']})

# Group the data by cluster and calculate the mean age for each cluster

In [46]:

cluster\_means **=** cluster\_df**.**groupby('Cluster')['Age']**.**mean()

# Create a bar plot to visualize the results

In [47]:

plt**.**figure(figsize**=**(8, 6))

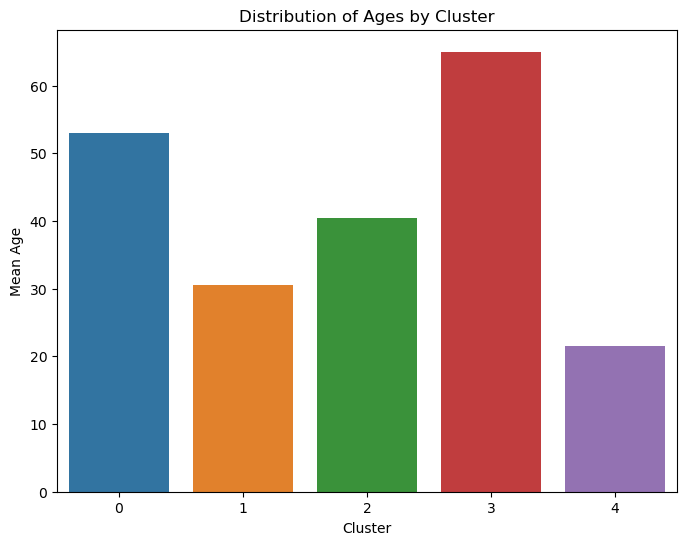
sns**.**barplot(x**=**cluster\_means**.**index, y**=**cluster\_means**.**values)

plt**.**xlabel('Cluster')

plt**.**ylabel('Mean Age')

plt**.**title('Distribution of Ages by Cluster')

plt**.**show()

****

# Load necessary libraries

library(dplyr)

library(readr)

# Read the data

df <- read\_csv("C:/Users/Bala Vignesh.A/Desktop/SCMA 632/Survey (1).csv")

# Display column names

print(colnames(df))

# Display column names

existing\_cols <- colnames(df)

print(existing\_cols)

# Define the correct columns to be encoded (modify this list based on the actual column names)

cols <- c('City', 'Sex', 'Occupation', 'Income', 'Planning to Buy a new house',

'Time Frame', 'Reasons for buying a house', 'what type of House',

'Number of rooms', 'Size of House', 'Budget', 'Finished/Semi Finished',

'Influence Decision', 'EMI', 'X1.Proximity.to.city',

'X2.Proximity.to.schools', 'X3.Proximity.to.transport',

'X4.Proximity.to.work.place', 'X5.Proximity.to.shopping',

'X1.Gym.Pool.Sports.facility', 'X2.Parking.space', 'X3.Power.back.up',

'X4.Water.supply', 'X5.Security', 'X1.Exterior.look', 'X2.Unit.size',

'X3.Interior.design.and.branded.components',

'X4.Layout.plan.Integrated.etc.', 'X5.View.from.apartment', 'X1.Price',

'X2.Booking.amount', 'X3.Equated.Monthly.Instalment.EMI',

'X4.Maintenance.charges', 'X5.Availability.of.loan',

'X1.Builder.reputation', 'X2.Appreciation.potential',

'X3.Profile.of.neighbourhood', 'X4.Availability.of.domestic.help',

'Time', 'Size', 'Budgets', 'Maintainances', 'EMI.1', 'ages', 'sex',

'Finished.Semi.Finished.1', 'Influence.Decision.1')

# Check missing columns

missing\_cols <- setdiff(cols, existing\_cols)

if (length(missing\_cols) > 0) {

print("The following columns are missing from the DataFrame:")

print(missing\_cols)

} else {

# Preprocess the data: Encode categorical variables

df\_encoded <- df %>%

mutate(across(all\_of(cols), as.numeric))

# Select background variables

background\_vars <- df\_encoded %>% select(tail(cols, 20))

# Scale the data

background\_vars\_scaled <- scale(background\_vars)

# Determine the optimal number of clusters using the elbow method

wcss <- numeric(10)

for (k in 1:10) {

set.seed(42)

kmeans\_result <- kmeans(background\_vars\_scaled, centers = k, nstart = 25)

wcss[k] <- kmeans\_result$tot.withinss

}

# Plot the elbow method

plot(1:10, wcss, type = "b", pch = 19, frame = FALSE,

xlab = "Number of clusters",

ylab = "WCSS",

main = "Elbow Method for Optimal Number of Clusters")

# Perform K-means clustering with 3 clusters

set.seed(42)

kmeans\_result <- kmeans(background\_vars\_scaled, centers = 3, nstart = 25)

df$Cluster <- kmeans\_result$cluster

# Analyze and interpret clusters

cluster\_summary <- df %>%

group\_by(Cluster) %>%

summarise(across(where(is.numeric), mean, na.rm = TRUE))

print(cluster\_summary)

# Visualize the clusters

library(ggplot2)

ggplot(df, aes(x = factor(Cluster), y = ages)) +

geom\_bar(stat = "summary", fun = "mean") +

labs(title = "Distribution of Ages by Cluster", x = "Cluster", y = "Mean Age")

}