Project Report

Python, Data Science and Machine Learning Integrated Program

Name: Pranav sharma

Registration No: 12324310

E-mail Id:sharmapranav.488@gmail.com

Dated: 29/27/2024

MALL CUSTOMER SEGMENTATION

INTRODUCTION

In today's competitive retail environment, understanding customer behavior is crucial for businesses aiming to enhance their marketing strategies and improve customer satisfaction. One effective method to achieve this is through customer segmentation. By dividing a diverse customer base into smaller, more homogenous groups, retailers can tailor their marketing efforts and services to meet the specific needs of different customer segments.

The goal of this project is to perform customer segmentation for a mall's clientele, using data-driven techniques to identify distinct customer groups. By analyzing various attributes such as age, gender, income, and spending behavior, we aim to uncover meaningful segments within the customer base. This segmentation will enable mall management to personalize marketing campaigns, optimize product placement and inventory, and enhance overall customer satisfaction and loyalty.

To achieve these objectives, we will use a comprehensive dataset containing demographic and purchasing information of the mall's customers. Our methodology involves several key steps: data preprocessing to clean and prepare the data, exploratory data analysis **(EDA)** to visualize and summarize the data, and the application of clustering algorithms like **K-Means** to identify distinct customer segments. We will then analyze each segment to provide actionable insights.

The tools and technologies employed in this project include Python for data manipulation and analysis, Pandas for data cleaning, **Matplotlib and Seaborn** for visualization, and **Scikit-Learn** for implementing clustering algorithms. By the project's conclusion, we expect to identify and describe several distinct customer segments, offering valuable insights that will help the mall enhance its marketing strategies, improve customer experiences, and drive business growth.

OBJECTIVE

The primary objective of this project is to perform customer segmentation for a mall's clientele. Using data-driven techniques, we will analyze various attributes of customers to identify distinct segments. This segmentation will enable the mall management to:

1. Personalize Marketing Campaigns:

Customer segmentation allows the mall to tailor its marketing campaigns to different groups based on their unique characteristics and preferences. For instance, one segment might consist of young professionals with high disposable income, who may respond well to promotions for upscale brands and exclusive events. Another segment could be families looking for discounts on family-friendly products and activities. By understanding these distinct needs, the mall can design targeted advertisements, offers, and promotions that resonate with each group, thereby increasing the effectiveness of marketing efforts and improving customer engagement.

2. Optimize Product Placement and Inventory:

With insights from customer segmentation, the mall can make informed decisions about product placement and inventory management. For example, if one segment is predominantly interested in high-tech gadgets and electronics, the mall can strategically position these products in areas where this segment is more likely to shop. Conversely, if another segment prefers luxury goods, placing these products in prominent locations and ensuring adequate stock levels can enhance the shopping experience for that group. This targeted approach helps in maximizing sales opportunities and reducing the risk of overstocking or stockouts.

3. Enhance Customer Loyalty and Retention:

Understanding the specific needs and preferences of different customer segments enables the mall to create tailored loyalty programs and personalized experiences. For instance, a loyalty program offering exclusive rewards and benefits to frequent shoppers in a particular segment can increase their satisfaction and encourage repeat visits. Personalized services, such as special discounts or exclusive previews for certain segments, can also strengthen customer relationships and foster long-term loyalty. By catering to the distinct needs of each segment, the mall can enhance the overall shopping experience and improve customer retention.

4. Increase Overall Sales and Profitability:

By leveraging the insights gained from customer segmentation, the mall can make data-driven decisions that boost sales and profitability. Targeted marketing campaigns and optimized product placement attract more customers and encourage higher spending. Additionally, by aligning inventory with the preferences of each segment, the mall can increase turnover and reduce waste. Enhancing customer loyalty through personalized experiences also contributes to repeat business and higher lifetime value. Overall, these strategies help in driving sales growth and improving the mall's profitability.

OVERVIEW OF THE DATASET

Dataset Description: The dataset used in this project is the "Mall Customers" dataset, which provides information about customers from a mall. The dataset contains demographic and behavioral attributes of the customers, which can be used to perform segmentation. The dataset includes the following columns:

1. **CustomerID:** Unique identifier for each customer.
2. **Gender:** Gender of the customer (Male/Female).
3. **Age:** Age of the customer.
4. **Annual Income (k$):** Annual income of the customer in thousands of dollars.
5. **Spending Score (1-100):** Spending score assigned by the mall based on customer behavior and spending nature (1 being lowest and 100 being highest).

CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72

Attributes:

- **CustomerID:** A numerical identifier unique to each customer.
- **Gender:** Categorical variable indicating the customer's gender.
- **Age:** Numerical variable indicating the customer's age.
- **Annual Income (k$):** Numerical variable indicating the customer's annual income in thousands of dollars.
- **Spending Score (1-100):** Numerical variable indicating the spending score, a metric assigned by the mall based on customer spending behavior.

Purpose of the Dataset: The dataset is used to perform customer segmentation analysis. By examining the demographic and behavioral attributes of the customers, we aim to identify distinct groups of customers who exhibit similar purchasing behaviors. These insights will enable the retail store to develop targeted marketing strategies and enhance overall customer satisfaction.

Data Source: The dataset is publicly available and can be downloaded from Kaggle at the following link: Mall Customers Dataset on Kaggle.

DATA AND METHODOLOGY:

The dataset used for this project contains demographic and purchasing information about the mall's customers, including age, gender, income, and spending scores. The methodology involves the following steps:

1. Data Preprocessing: Cleaning and preparing the data for analysis.

2. Exploratory Data Analysis (EDA): Visualizing and summarizing the main characteristics of the data to uncover patterns and insights.

3. Segmentation Techniques: Applying clustering algorithms such as K-Means to segment the customers into distinct groups.

4. Segment Analysis: Interpreting the characteristics of each segment to provide actionable insights.

TOOLS AND TECHNOLOGIES

For this project, we will utilize the following tools and technologies:

- Python: For data manipulation and analysis.
- Pandas: For data cleaning and preparation.
- Matplotlib and Seaborn: For data visualization.
- Scikit-Learn: For implementing clustering algorithms.

EXPECTED OUTCOMES

By the end of this project, we aim to identify and describe several distinct customer segments within the mall's clientele. These segments will provide valuable insights that can help the mall enhance its marketing strategies, improve customer experiences, and ultimately drive business growth.

THE ACTUAL CODE AND PRACTICAL PROOF OF THE PROJECT IS ATTACHED BELOW.

Data Collection

In [26]:
```
import pandas as pd

# Load the dataset
file_path = 'Mall_Customers.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataset
data.head(10), data.info(), data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

Out[26]:
```
(   CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
 0           1    Male   19                  15                      39
 1           2    Male   21                  15                      81
 2           3  Female   20                  16                       6
 3           4  Female   23                  16                      77
 4           5  Female   31                  17                      40
 5           6  Female   22                  17                      76
 6           7  Female   35                  18                       6
 7           8  Female   23                  18                      94
 8           9    Male   64                  19                       3
 9          10  Female   30                  19                      72,
 None,
        CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
 count  200.000000  200.000000          200.000000              200.000000
 mean   100.500000   38.850000           60.560000               50.200000
 std     57.879185   13.969007           26.264721               25.823522
 min      1.000000   18.000000           15.000000                1.000000
 25%     50.750000   28.750000           41.500000               34.750000
 50%    100.500000   36.000000           61.500000               50.000000
 75%    150.250000   49.000000           78.000000               73.000000
 max    200.000000   70.000000          137.000000               99.000000)
```

Data Cleaning

In [27]:
```
count=data.isnull().sum()
count
```

Out[27]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Replace NaN of age with mean

In [28]:
```
mean_age=data['Age'].mean()
data["Age"].fillna(mean_age,inplace=True)
data.head(10)
```

```
C:\Users\JAYDEV\AppData\Local\Temp\ipykernel_13148\1930870467.py:2: FutureWar
ning: A value is trying to be set on a copy of a DataFrame or Series through
chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work b
ecause the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.
method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  data["Age"].fillna(mean_age,inplace=True)
```

Out[28]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

Renaming columns for better readability

In [29]:
```
# Renaming columns for better readability
data.columns = ["CustomerID", "Gender", "Age", "AnnualIncome", "SpendingScore"
print(data)
```

```
     CustomerID  Gender  Age  AnnualIncome  SpendingScore
0             1    Male   19            15             39
1             2    Male   21            15             81
2             3  Female   20            16              6
3             4  Female   23            16             77
4             5  Female   31            17             40
..          ...     ...  ...           ...            ...
195         196  Female   35           120             79
196         197  Female   45           126             28
197         198    Male   32           126             74
198         199    Male   32           137             18
199         200    Male   30           137             83

[200 rows x 5 columns]
```

Replace NaN of gender with mode

In [30]:
```
mode_gender=data['Gender'].mode()[0]
type(mode_gender)
mode_gender
```

Out[30]:
```
'Female'
```

In [31]:
```
data["Gender"].fillna(mode_gender,inplace=True)
```

```
C:\Users\JAYDEV\AppData\Local\Temp\ipykernel_13148\2761098919.py:1: FutureWar
ning: A value is trying to be set on a copy of a DataFrame or Series through
chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work b
ecause the intermediate object on which we are setting values always behaves
as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.
method({col: value}, inplace=True)' or df[col] = df[col].method(value) instea
d, to perform the operation inplace on the original object.


  data["Gender"].fillna(mode_gender,inplace=True)
```

In [32]:
```
data.head(20)
```

Out[32]:

	CustomerID	Gender	Age	AnnualIncome	SpendingScore
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72
10	11	Male	67	19	14
11	12	Female	35	19	99
12	13	Female	58	20	15
13	14	Female	24	20	77
14	15	Male	37	20	13
15	16	Male	22	20	79
16	17	Female	35	21	35
17	18	Male	20	21	66
18	19	Male	52	23	29
19	20	Female	35	23	98

Removing rows with NAN Annual Income and Spending Score

In [33]:
```
data.dropna(inplace=True)
```

In [34]:
```
count=data.isnull().sum()
count
```

Out[34]:
```
CustomerID       0
Gender           0
Age              0
AnnualIncome     0
SpendingScore    0
dtype: int64
```

Convert the Gender column to numerical values using one-hot encoding or label encoding.

In [35]:
```
# Data transformation (e.g., encoding categorical variables)
data['Gender'] = data['Gender'].map({'Male': 0, 'Female': 1}).astype('int')
```

In [36]:
```
data
```

Out[36]:

	CustomerID	Gender	Age	AnnualIncome	SpendingScore
0	1	0	19	15	39
1	2	0	21	15	81
2	3	1	20	16	6
3	4	1	23	16	77
4	5	1	31	17	40
...	...	...	...	...	...
195	196	1	35	120	79
196	197	1	45	126	28
197	198	0	32	126	74
198	199	0	32	137	18
199	200	0	30	137	83

200 rows × 5 columns

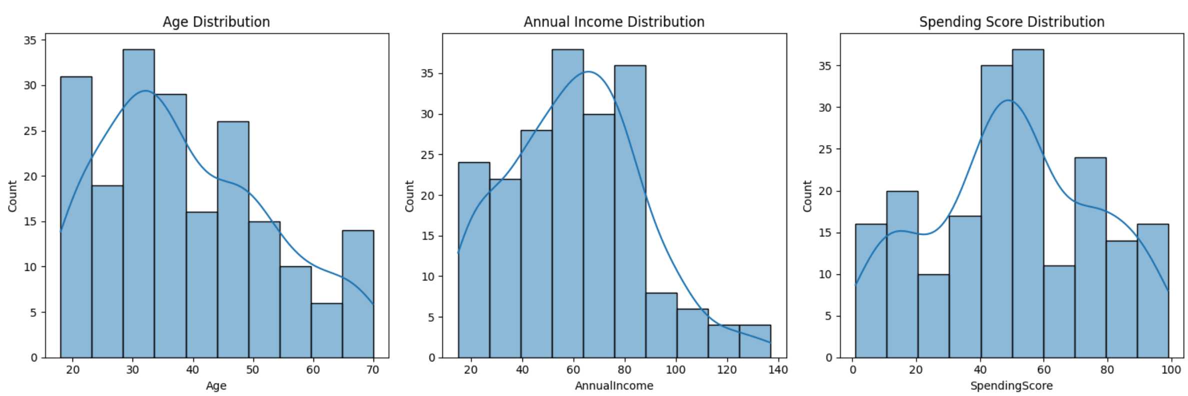
Save the cleaned dataset

In [37]:
```
cleaned_file_path = 'cleaned_mall_customers.csv'
data.to_csv(cleaned_file_path, index=False)
```

Exploratory Data Analysis (EDA)

- Calculating descriptive statistics.
- Creating histograms for age, annual income, and spending score distributions.
- Generating a scatter plot of annual income vs. spending score, colored by gender.

In [38]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate descriptive statistics
descriptive_stats = data.describe()

# Create histograms for Age, Annual Income, and Spending Score
plt.figure(figsize=(15, 5))

plt.subplot(1, 3, 1)
sns.histplot(data['Age'], bins=10, kde=True)
plt.title('Age Distribution')

plt.subplot(1, 3, 2)
sns.histplot(data['AnnualIncome'], bins=10, kde=True)
plt.title('Annual Income Distribution')

plt.subplot(1, 3, 3)
sns.histplot(data['SpendingScore'], bins=10, kde=True)
plt.title('Spending Score Distribution')

plt.tight_layout()
plt.show()

# Scatter plot of Annual Income vs. Spending Score colored by Gender
plt.figure(figsize=(10, 6))
sns.scatterplot(x='AnnualIncome', y='SpendingScore', hue='Gender', data=data, 
plt.title('Annual Income vs. Spending Score')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend(title='Gender', labels=['Female', 'Male'])
plt.show()

descriptive_stats
```



Out[38]:

	CustomerID	Gender	Age	AnnualIncome	SpendingScore
count	200.000000	200.000000	200.000000	200.000000	200.000000
mean	100.500000	0.560000	38.850000	60.560000	50.200000
std	57.879185	0.497633	13.969007	26.264721	25.823522
min	1.000000	0.000000	18.000000	15.000000	1.000000
25%	50.750000	0.000000	28.750000	41.500000	34.750000
50%	100.500000	1.000000	36.000000	61.500000	50.000000
75%	150.250000	1.000000	49.000000	78.000000	73.000000
max	200.000000	1.000000	70.000000	137.000000	99.000000

In [39]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Features to be used for clustering
features = ['Age', 'AnnualIncome', 'SpendingScore']

# Standardize the features
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data[features])

# Apply K-Means clustering with 5 clusters
kmeans = KMeans(n_clusters=5, random_state=42)
data['Cluster'] = kmeans.fit_predict(data_scaled)

# Create a scatter plot of Annual Income vs. Spending Score, colored by cluster
plt.figure(figsize=(10, 6))
sns.scatterplot(x='AnnualIncome', y='SpendingScore', hue='Cluster', palette='t
plt.title('Customer Segments: Annual Income vs. Spending Score')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend(title='Cluster')
plt.show()

data['Cluster'].value_counts()
```

Out[39]:
```
Cluster
0    58
3    45
1    40
4    31
2    26
Name: count, dtype: int64
```