Project Report

- **Project Title:** Customer Segmentation for a Retail Store
- **Project Manager:** Satyam Maheshwari
- **Start Date:** 13-07-24
- **End Date:** 17-07-24
- **Objectives:** To segment customers into distinct groups based on their purchasing behavior.

❖ **Scope:**

  - Data cleaning
  - EDA
  - Customer Segmentation using K-Means
  - Visualization using Matplotlib and Power BI

❖ **Technical Requirements Document (TRD)**

- **Data Sources:** Mall Customers dataset
- **Technologies:**
  - Python
  - Jupyter Notebook
  - Matplotlib
  - Seaborn
  - Scikit-learn
  - Power BI

- **Architecture:**
  - Data preprocessing
  - EDA
  - Clustering
  - Visualization
- **Data Flow:** Import data → Clean data → Analyze data → Segment customers → Visualize results

❖ **Deliverables:**

- Conclusions

  {Insights & Recommendations}

❖ **Tasks:**

1. **Data Loading and Initial Inspection**

The dataset 'Mall_Customers.csv' is loaded into a pandas Data Frame, and the first 10 rows are displayed to get an initial look at the data structure and values.

2. **Handling Missing Values**

- **Age Column:** Missing values in the 'Age' column are filled with the mean age.
- **Gender Column:** Missing values in the 'Gender' column are filled with the mode (most frequent value) of the column.

3. **Data Cleaning and Transformation**

- Columns are renamed for better readability: CustomerID, Gender, Age, Annual Income, Spending Score.
- The 'Gender' column is encoded numerically: 0 for Male and 1 for Female.
- All remaining missing values are confirmed to be handled.

4. Exploratory Data Analysis (EDA)

A statistical summary of the dataset is provided, giving insights into measures such as mean, median, standard deviation, etc.

5. Visualizing Distributions

- **Age Distribution:** A histogram with a kernel density estimate (KDE) overlay is created to show the age distribution of customers.
- **Annual Income Distribution:** A histogram with KDE shows how customers' annual incomes are distributed.
- **Spending Score Distribution:** A histogram with KDE illustrates the distribution of customers' spending scores.

6. Visualizing Relationships

A scatter plot is created to show the relationship between Annual Income and Spending Score, with points colored by Gender, revealing any patterns based on gender differences.

7. Clustering with K-Means

- **Feature Selection:** Relevant features (Age, Annual Income, Spending Score) are selected for clustering.
- **Standardization:** Features are standardized to ensure equal weighting during clustering.

- **K-Means Clustering:** The K-Means algorithm is applied with 5 clusters to segment customers into distinct groups.
- **Cluster Visualization:** A scatter plot visualizes the resulting customer segments, colored by cluster, highlighting distinct groupings based on spending and income.

❖ Conclusion:

Through this analysis, we have cleaned and preprocessed the data, visualized key distributions and relationships, and applied K-Means clustering to identify distinct customer segments.

The visualizations indicate clear patterns in customer spending behavior and income levels.

These insights can help tailor marketing strategies and improve customer relationship management by targeting specific segments based on their characteristics and spending habits.

Overall, the approach effectively segments customers and provides actionable insights for business decisions.

❖ Charts & Code Snippets:

+ Code + Text

```
[13] count=data.isnull().sum()
     count
```

```
CustomerID       0
Gender           0
Age              0
AnnualIncome     0
SpendingScore    0
dtype: int64
```

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

```
[15] count=data.isnull().sum()
     count
```

```
CustomerID       0
Gender           0
Age              0
AnnualIncome     0
SpendingScore    0
dtype: int64
```

```
data
```

	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

+ Code + Text

EDA

```
data.describe()
```

	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

```
[18] data
```

	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

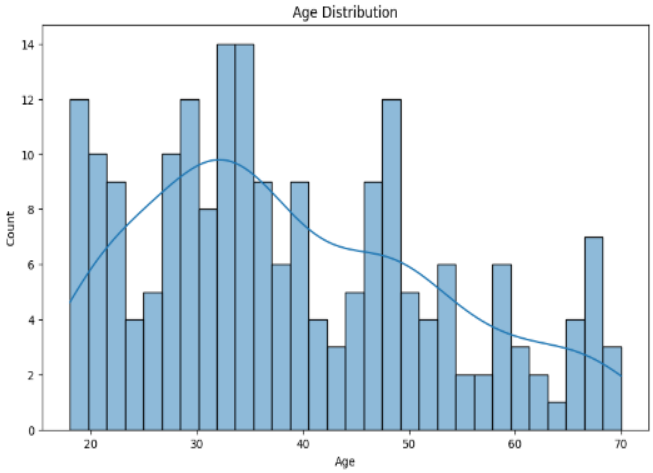
```
[19] import matplotlib.pyplot as plt
     import seaborn as sns
```

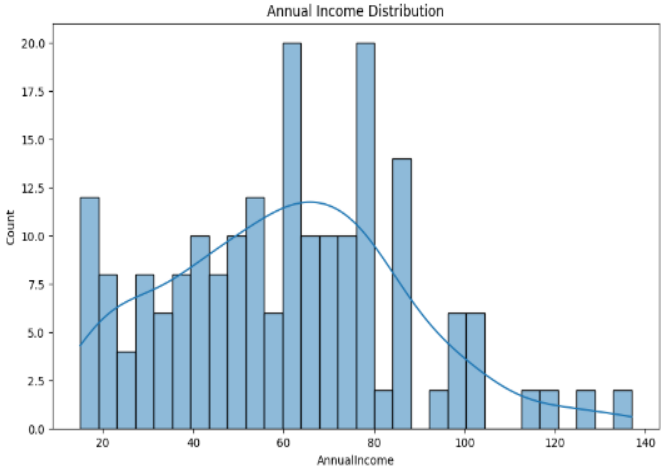
```
#histplot-Age,Annual income,Spending score
# Visualizing distributions

#----Age Distribution----
plt.figure(figsize=(10, 6))
sns.histplot(data['Age'], bins=30, kde=True)
plt.title('Age Distribution')
plt.show()

#-----Annual Income Distribution------
plt.figure(figsize=(10, 6))
sns.histplot(data['AnnualIncome'], bins=30, kde=True)
plt.title('Annual Income Distribution')
plt.show()

#-----Spending Score Distribution------
plt.figure(figsize=(10, 6))
sns.histplot(data['SpendingScore'], bins=30, kde=True)
plt.title('Spending Score Distribution')
plt.show()
```

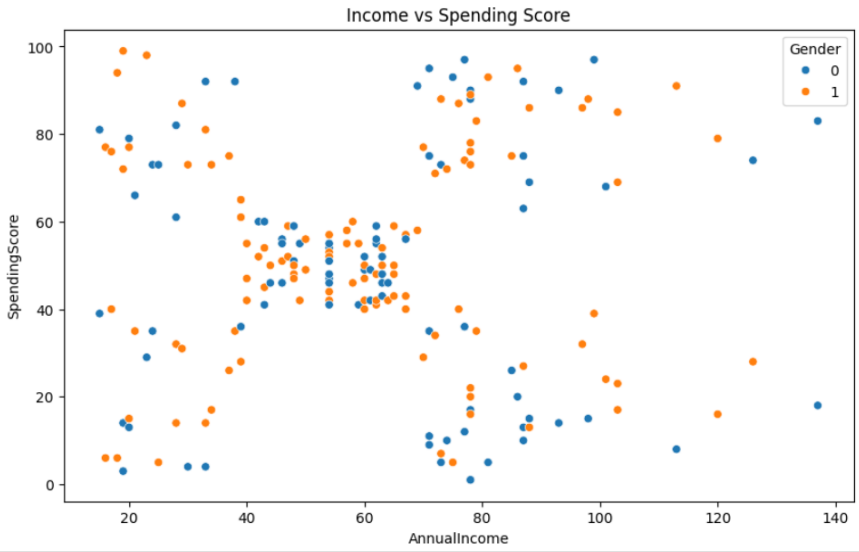

Age Distribution
Count
Age
0
2
4
6
8
10
12
14
20
30
40
50
60
70


Annual Income Distribution
Count
AnnualIncome
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
20
40
60
80
100
120
140

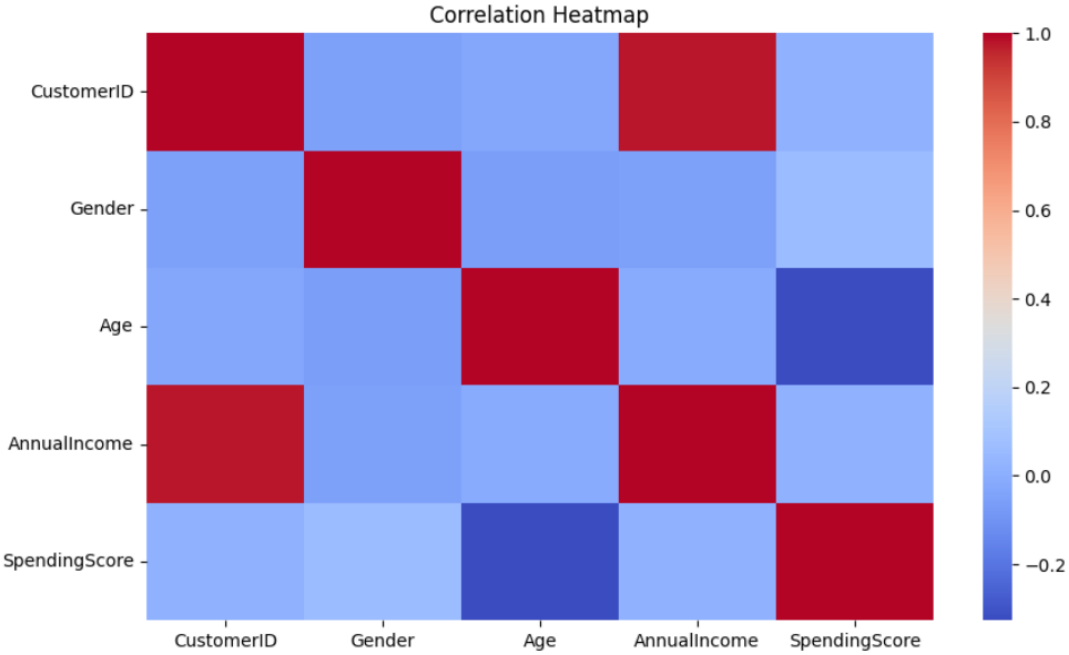

Spending Score Distribution
Count
SpendingScore
0
2
4
6
8
10
12
14
0
20
40
60
80
100

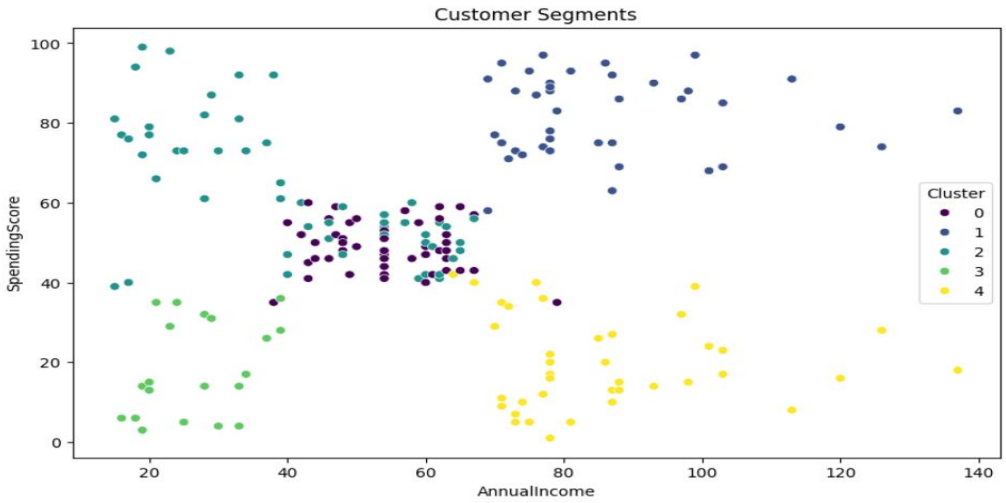
+ Code + Text

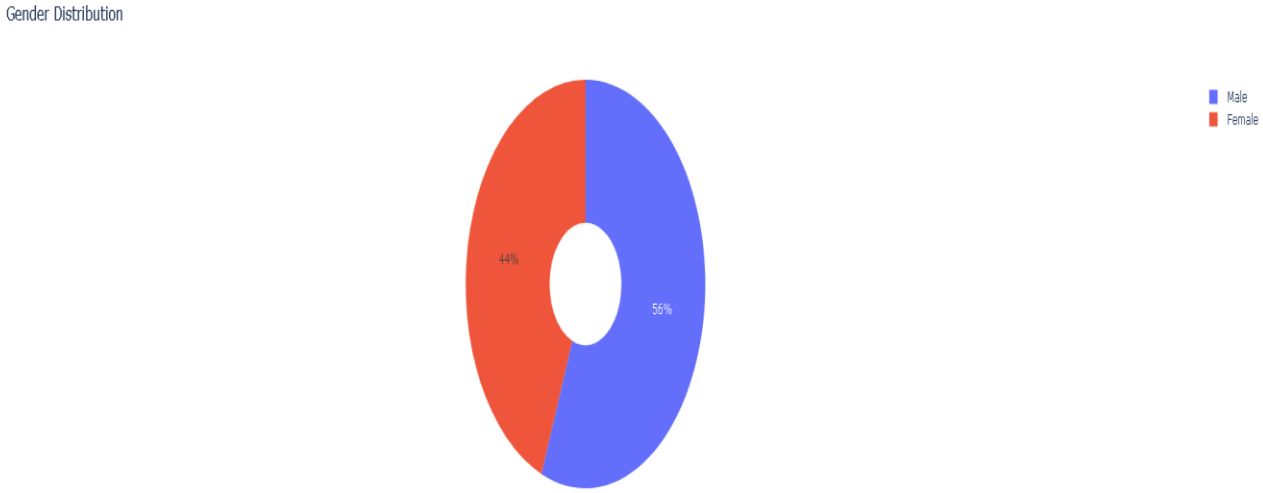
```
# Visualizing relationships
plt.figure(figsize=(10, 6))
sns.scatterplot(data=data, x='AnnualIncome', y='SpendingScore', hue='Gender')
plt.title('Income vs Spending Score')
plt.show()
```




```
# Create a heatmap to visualize correlations
plt.figure(figsize=(10, 6))
sns.heatmap(data.corr(), cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```




Customer Segments
SpendingScore
AnnualIncome
Cluster
0
1
2
3
4
0
20
40
60
80
100
20
40
60
80
100
120
140


Gender Distribution
Male
Female
44%
56%