Customer Segmentation Using Clustering (Python)

THESIS FOR INFOTACT SOLUTIONS

Aman Jawalekar

INDEX

1. Acknowledgement...
2. Preface ....
3. Introduction ....
4. Methodology ....
5. Data Collection..
6. Data Pre-processing...
7. Exploratory Data Analysis ....
8. Feature Scaling……………………………………………………………….
9. Clustering Using K-Means…………………………………………………...
10. Dimensionality Reduction & Visualization…………………………………..
11. Cluster Profiling………………………………………………………………
12. Result: Marketing Strategy Development……………………………………

Acknowledgement

We would like to express our sincere gratitude to Infotact Solutions for providing us with the opportunity to work on the *Customer Segmentation Using Clustering (Python)* project. This experience has significantly enhanced our understanding of real-world data analytics.

We are thankful to our mentors and instructors for their constant guidance and support throughout the project. Their insights and feedback helped us align our analysis with practical business objectives.

Finally, we extend our thanks to all data sources and platforms that enabled this exploration and helped us transform raw data into actionable insights.

Preface

In today's competitive market, understanding customers is no longer an advantage—it's a necessity. Businesses that decode consumer behavior can craft personalized experiences, foster loyalty, and maximize profitability. This project, *Customer Segmentation and Persona Analysis*, aims to achieve precisely that: transforming complex customer data into actionable segments for targeted marketing strategies.

Undertaken as a comprehensive data analytics initiative, the project leverages ***K-Means clustering*** to identify distinct customer groups based on demographic, behavioral, and transactional attributes. Beginning with ***data collection and cleaning***, the process moves through ***exploratory data analysis***, normalization, and the application of machine learning algorithms to derive meaningful clusters. The project further includes ***visualizations using PCA and scatter plots***, evaluation through ***Elbow Method and Silhouette Score***, and concludes with ***marketing recommendations tailored to each segment***.

The methodology embodies both analytical depth and practical applicability. From identifying high-value loyal customers to recognizing trend-driven buyers, the insights presented here are designed to enhance decision-making in real-world business contexts. This endeavor not only demonstrates technical expertise in data preprocessing, clustering, and visualization but also underlines the strategic role of analytics in customer relationship management.

By combining rigorous analysis with interpretability, this project illustrates the power of data in shaping marketing strategies and customer engagement initiatives. It is our hope that this work serves as a valuable resource for businesses seeking to harness data-driven insights for sustainable growth.

Introduction

In today's competitive business landscape, understanding customer behavior is no longer optional—it's a necessity for driving growth and retaining loyalty. This project, "Customer Segmentation using K-Means Clustering," focuses on analyzing demographic and behavioral data from a jewelry retail business to uncover actionable insights for targeted marketing.

The dataset includes diverse attributes such as age, gender, location, product preferences, spending patterns, engagement levels, and retention metrics. By applying Exploratory Data Analysis (EDA) and machine learning techniques, the project aims to reveal distinct customer groups with unique purchasing behaviors and characteristics.

The core of the analysis lies in implementing the K-Means clustering algorithm, an unsupervised learning technique that segments customers based on similarity in their attributes. To ensure accuracy, the project leverages normalization techniques, evaluates the optimal number of clusters using the Elbow Method and Silhouette Score, and visualizes the results through PCA-based cluster plots.

The ultimate goal is to translate these findings into actionable marketing strategies, enabling personalized campaigns, better resource allocation, and improved customer retention. This approach not only strengthens customer relationships but also enhances profitability by focusing efforts where they matter most.

Methdology

1. **Data Collection**
2. **Data Cleaning & Preprocessing**
3. **Exploratory Data Analysis (EDA)**
4. **Feature Scaling**
5. **Clustering Using K-Means**
6. **Dimensionality Reduction & Visualization**
7. **Cluster Profiling**
8. **Marketing Strategy Development**

Data Collection

The dataset for this project, “Jewellery Customer Segmentation Analysis Personas,” was sourced from Kaggle. It includes 1,000 customer records with 40 attributes, covering demographics (Age, Gender, Location), purchase behavior (Purchase Count, Total Spend, Average Value), engagement metrics, retention details, and financial indicators.
The data was downloaded in CSV format for ease of use and processed using Python libraries (pandas, numpy). This comprehensive dataset provides the foundation for clustering customers based on behavioral and demographic traits, enabling actionable insights for targeted marketing.

	CustomerID	Age	Gender	Location	ProductType	PurchaseCount	TotalSpend	AveragePurchaseValue	EngagementScore	FollowDuration	...	IsActive	MarketingSp
0	1	56	Male	North America	Bracelet	9	737	81.888889	2.079020	24	...	True	85.052
1	2	46	Female	Middle East	Ring	3	334	111.333333	7.614333	18	...	True	87.253
2	3	32	Male	Asia	Ring	1	515	515.000000	5.783094	10	...	True	56.871
3	4	60	Other	North America	Bracelet	7	1037	148.142857	3.591508	7	...	True	86.957
4	5	25	Male	Europe	Ring	5	598	119.600000	5.435376	8	...	False	96.423
5	6	38	Male	Middle East	Earrings	5	1073	214.600000	3.699479	6	...	False	122.273
6	7	56	Male	North America	Earrings	1	733	733.000000	6.364738	22	...	True	115.672
7	8	36	Male	Asia	Bracelet	9	1773	197.000000	4.904595	1	...	True	120.876
8	9	40	Other	North America	Necklace	7	1119	159.857143	2.479650	4	...	False	50.836
9	10	28	Other	Asia	Earrings	8	214	26.750000	2.049248	14	...	True	71.766
10	11	28	Female	Europe	Ring	9	1031	114.555556	5.920261	17	...	True	116.168
11	12	41	Other	Middle East	Necklace	6	773	128.833333	9.121717	20	...	True	98.398
12	13	53	Other	Middle East	Necklace	5	1836	367.200000	4.095619	18	...	True	50.531

The dataset comprises **1,000 customer records** with **40 attributes** representing both **demographic** and **behavioral features**. Key columns include:

- **CustomerID**: Unique identifier for each customer
- **Age**: Customer age
- **Gender**: Male, Female, or Other
- **Location**: Region (North America, Asia, Europe, etc.)
- **ProductType**: Type of jewelry purchased (Ring, Bracelet, Earrings, Necklace)
- **PurchaseCount**: Total number of purchases

- **TotalSpend**: Total amount spent by the customer
- **AveragePurchaseValue**: Average value per purchase
- **EngagementScore**: Engagement level with the brand
- **FollowDuration**: Duration (in months) customer has been following the brand
- **IsActive**: Indicates if the customer is currently active
- **MarketingSpend**: Estimated marketing cost associated with acquiring/retaining the customer

Additionally, the dataset includes **financial metrics** (ProductCost, SalesRevenue, CPA), **retention data** (CustomerAcquisitionDate, CustomerExitDate, RetentionPeriod), and performance measures like **CompletionRate**.

This dataset provides a robust basis for **customer segmentation using K-Means clustering**, enabling insights into spending behavior, engagement patterns, and retention trends for targeted marketing.

Data Pre-Processing:

```
#Check for missing values
df.isnull().sum()
```

```
CustomerID                      0
Age                             0
Gender                          0
Location                        0
ProductType                     0
PurchaseCount                   0
TotalSpend                      0
AveragePurchaseValue            0
EngagementScore                 0
FollowDuration                  0
PreferredChannel                0
CustomerCost                    0
CustomerProfitabilityScore      0
ROAS                            0
ROI                             0
RFMScore                        0
AverageRFM                      0
Persona                         0
OperatingExpenses               0
AdditionalCosts                 0
GrossProfit                     0
GrossProfitMargin               0
OperatingProfit                 0
OperatingProfitMargin           0
NetProfit                       0
NetProfitMargin                 0
CLTV                            0
LastPurchaseDate                0
SubscriptionStartDate           0
SubscriptionEndDate           800
IsActive                        0
MarketingSpend                  0
ProductCost                     0
SalesRevenue                    0
CustomerAcquisitionDate         0
CustomerExitDate              800
CustomerRetentionPeriod         0
```

The dataset was examined for missing values using df.isnull().sum(). The analysis revealed:

- Most columns have 0 missing values, ensuring high data completeness.
- Only two columns contain missing values:
    - **SubscriptionEndDate:** 800 missing entries
    - **CustomerExitDate:** 800 missing entries
      These columns are likely related to subscription or churn details and missing values indicate customers who are still active.

Since the missing data is confined to non-essential features for clustering (dates), they can either be ignored or encoded appropriately if needed.

Handling Missing Values and Irrelevant Features:

To ensure clean and reliable data for clustering:

- Dropped Irrelevant Columns:
    - CustomerID (unique identifier, not useful for analysis)
    - CustomerAcquisitionDate and CustomerExitDate (date fields not required for segmentation).
- Removed Missing Records:
  Rows with missing values were dropped using dropna() to maintain dataset integrity.
- Reset Index:
  After removing rows and columns, the index was reset to maintain sequential order.

These steps prepared the dataset for further preprocessing like encoding and scaling.

Exploratory Data Aanalysis

Descriptive Statistics Summary

After cleaning the data, basic statistical analysis was performed using df.describe(). Key observations:

- **Dataset Size:** 200 customers after cleaning.
- **Age:** Ranges from 18 to 64 years, with an average of 41.86.
- **Purchase Behavior:**
  - **Average Purchase Count**: ~4.78 (min: 1, max: 9).
  - **Total Spend:** Mean of $1011.47, ranging from $110 to $1958.
  - **Average Purchase Value:** Highly variable, up to $1958 for single purchases.
- **Engagement:** Engagement Score averages around 5.57 (scale varies, min: 1, max: 9.97).
- **Customer Cost:** Average cost per customer is ~$605, with a maximum of $1079.

```
# Basic statistics
print(df.describe())
```

```
              Age  PurchaseCount   TotalSpend  AveragePurchaseValue  \
count  200.000000     200.000000   200.000000            200.000000
mean    41.865000       4.780000  1011.475000            339.875032
std     13.639537       2.573854   540.101925            382.372169
min     18.000000       1.000000   110.000000             14.500000
25%     30.000000       3.000000   501.750000            116.218750
50%     43.000000       4.000000  1064.500000            209.625000
75%     53.250000       7.000000  1466.500000            351.312500
max     64.000000       9.000000  1958.000000           1958.000000

       EngagementScore  FollowDuration  CustomerCost  \
count       200.000000      200.000000    200.000000
mean          5.570163       11.685000    605.737500
std           2.517522        7.140779    270.050963
min           1.002043        1.000000    155.000000
25%           3.695722        5.000000    350.875000
50%           5.619622       11.000000    632.250000
75%           7.503012       18.000000    833.250000
max           9.968737       24.000000   1079.000000

       CustomerProfitabilityScore        ROAS         ROI  ...   NetProfit  \
count                  200.000000  200.000000  200.000000  ...  200.000000
mean                   405.737500   10.114750    0.561036  ...  202.295000
std                    270.050963    5.401019    0.273541  ...  108.020385
min                    -45.000000    1.100000   -0.290323  ...   22.000000
25%                    150.875000    5.017500    0.429959  ...  100.350000
50%                    432.250000   10.645000    0.683655  ...  212.900000
75%                    633.250000   14.665000    0.759976  ...  293.300000
max                    879.000000   19.580000    0.814643  ...  391.600000
```

Gender Distribution Analysis:

The gender distribution in the dataset was visualized using a count plot. The results show:

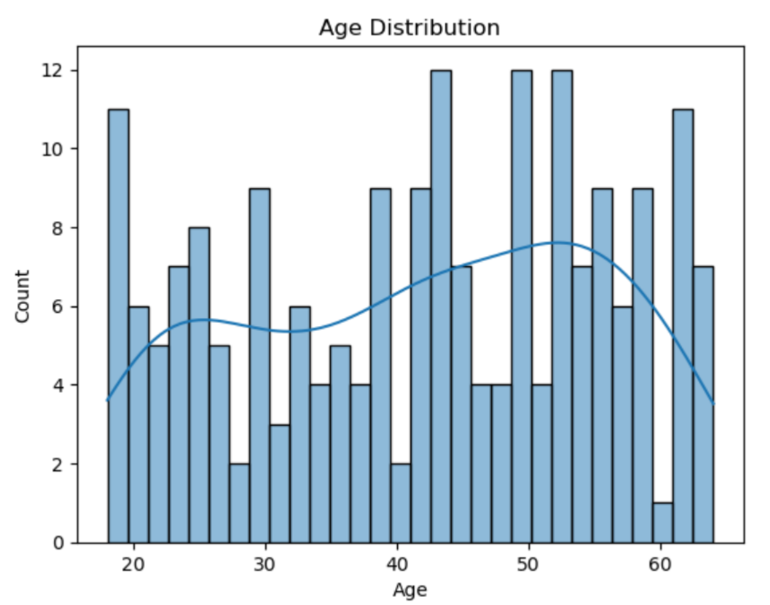
- **Male:** Highest representation with approximately 73 customers.
- **Other Gender:** Around 65 customers.
- **Female:** Close to 63 customers.

The distribution is fairly balanced across categories, indicating that marketing strategies should cater to all genders without heavy bias toward one group.

Age Distribution Analysis

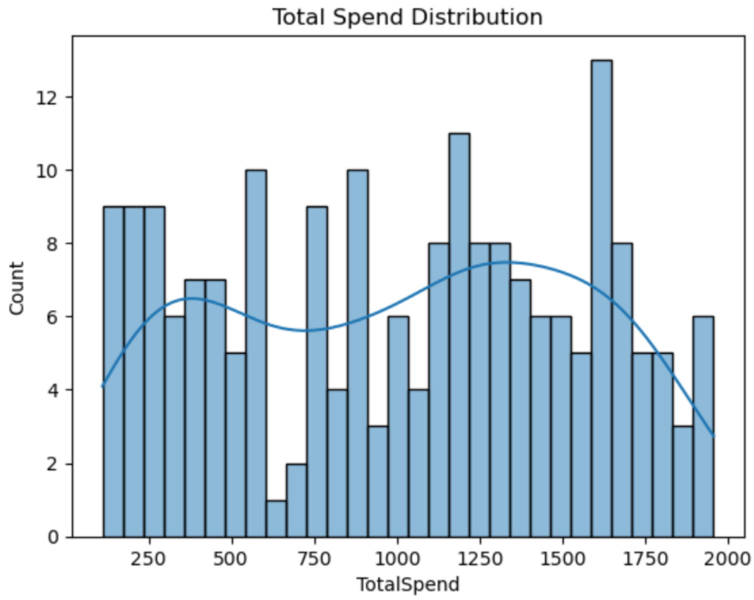
The age distribution of customers ranges from **18 to 64 years**, with the following observations:

- The dataset shows a **fairly uniform spread** across different age groups.
- A slight concentration is observed between **45–55 years**, indicating a strong middle-aged customer segment.

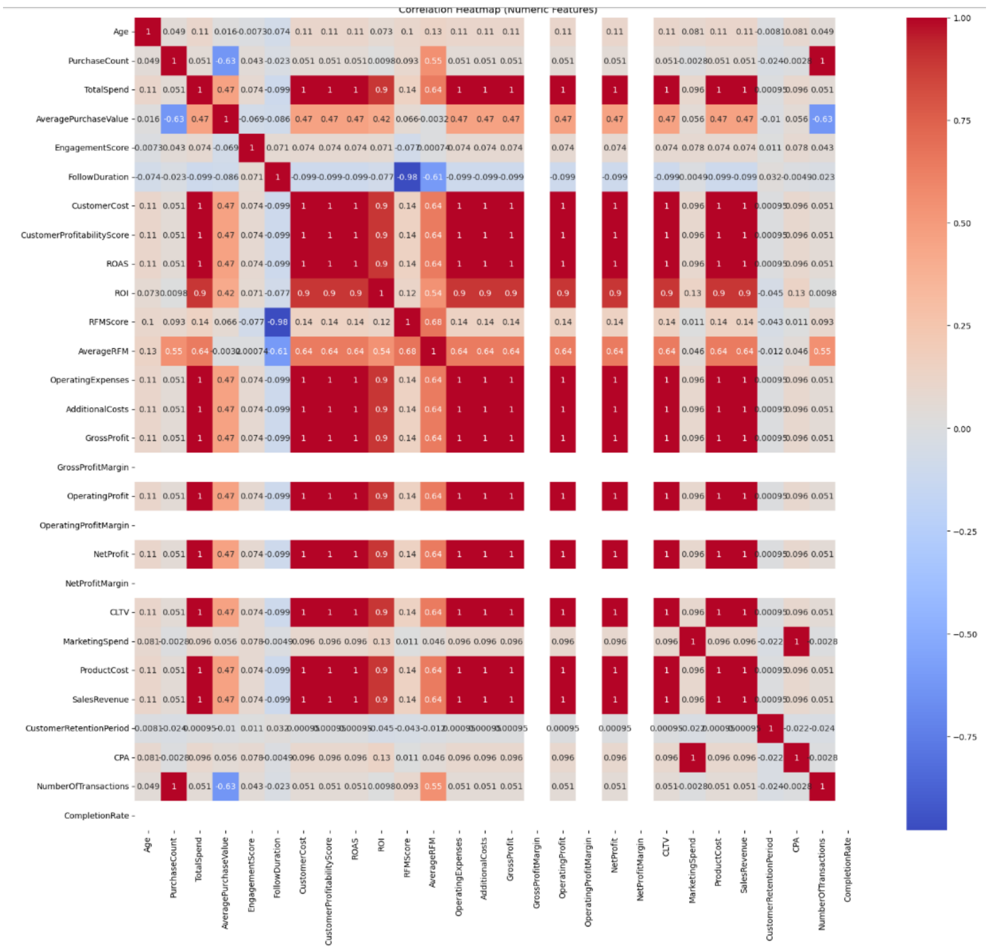
- Younger customers (18–25 years) and older customers (55+ years) are less frequent but still significant.
- This insight suggests that marketing strategies should primarily target the middle-aged demographic, while personalized campaigns can engage younger and older customers.
- 

Total Spend Distribution

- **Range:** Customer spending varies from approximately **₹110 to ₹1,958**.
- **Distribution:** The histogram shows a **slightly right-skewed pattern**, meaning a few customers spend significantly more than others.
- **Concentration:** Most customers fall between **₹500 and ₹1,500**, with a noticeable peak around ₹1,400–₹1,700.
- **Insight:** A minority of **high-value customers** contribute heavily to revenue. This indicates potential for **VIP loyalty programs** or **premium offers**.


Total Spend Distribution
Count
TotalSpend
0
2
4
6
8
10
12
250
500
750
1000
1250
1500
1750
2000

•


Correlation Heatmap (Numeric Features)
Age
PurchaseCount
TotalSpend
AveragePurchaseValue
EngagementScore
FollowDuration
CustomerCost
CustomerProfitabilityScore
ROAS
ROI
RFMScore
AverageRFM
OperatingExpenses
AdditionalCosts
GrossProfit
GrossProfitMargin
OperatingProfit
OperatingProfitMargin
NetProfit
NetProfitMargin
CLTV
MarketingSpend
ProductCost
SalesRevenue
CustomerRetentionPeriod
CPA
NumberOfTransactions
CompletionRate
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

Correlation Heat-map Analysis:

- **Highly Correlated Features:**
  - PurchaseCount and TotalSpend show **strong positive correlation**, which is expected since higher purchases lead to higher spending.
  - AveragePurchaseValue and AverageRFM exhibit notable correlation, indicating customers with higher purchase values often score better on RFM metrics.
  - ROAS, ROI, and CustomerProfitabilityScore share strong correlation with profitability measures like GrossProfit and OperatingProfit.
- **Weak Correlations:**
  - Age has almost **no significant correlation** with spending or engagement metrics, suggesting age is not a strong predictor of financial behavior in this dataset.
  - EngagementScore also shows minimal correlation with revenue-related variables.
- **Insight:**
  - Profitability metrics are **closely interrelated**, while demographic factors such as Age do not strongly influence spending patterns.
  - Clustering should focus on **behavioral and financial features** rather than demographics alone.

Encode Categorical Columns

```
categorical_cols = df.select_dtypes(include=['object', 'bool']).columns

le = LabelEncoder()
for col in categorical_cols:
    df[col] = le.fit_transform(df[col])

print("Encoded categorical columns:", categorical_cols)
```

```
Encoded categorical columns: Index(['Gender', 'Location', 'ProductType', 'PreferredChannel', 'Persona',
       'LastPurchaseDate', 'SubscriptionStartDate', 'SubscriptionEndDate',
       'IsActive'],
      dtype='object')
```

- All columns with **object** (text) or **boolean** types were identified as categorical.

- Each categorical column was encoded using **LabelEncoder** so that every unique category (e.g., 'Male', 'Female' in Gender, or True/False in IsActive) was assigned a unique integer value.
- This process enables algorithms to process previously non-numeric data efficiently and prepares the dataset for model training and prediction.
- Columns encoded include features such as *Gender*, *Location*, *ProductType*, *PreferredChannel*, *IsActive*, etc.
- While **label encoding** is simple and memory-efficient, it may not be suitable for all categorical variables, especially if the encoded numbers imply an artificial order in nominal categories.

This encoding step is a standard preprocessing technique and is essential for machine learning workflows where categorical data must be numerically represented.

Feature Scaling

```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)

print("Scaled data shape:", scaled_data.shape)
```

```
Scaled data shape: (200, 37)
```

- After encoding categorical columns, all features in the DataFrame are scaled using the **StandardScaler** from scikit-learn.
- The scaler transforms every column in the dataset individually by centering and normalizing: $X_{scaled} = \frac{X-\mu}{\sigma}$, where μ is the mean and σ is the standard deviation of each feature.
- The result is a transformed data array (scaled_data) with the same shape as the original dataset but with values optimized for training models—here, 200 samples by 37 features.
- **Feature scaling** ensures that models such as linear regression, k-means, SVM, and neural networks are not biased by variations in feature magnitude.

This step is vital for improving model performance, stability, and convergence speed in most machine learning workflows.

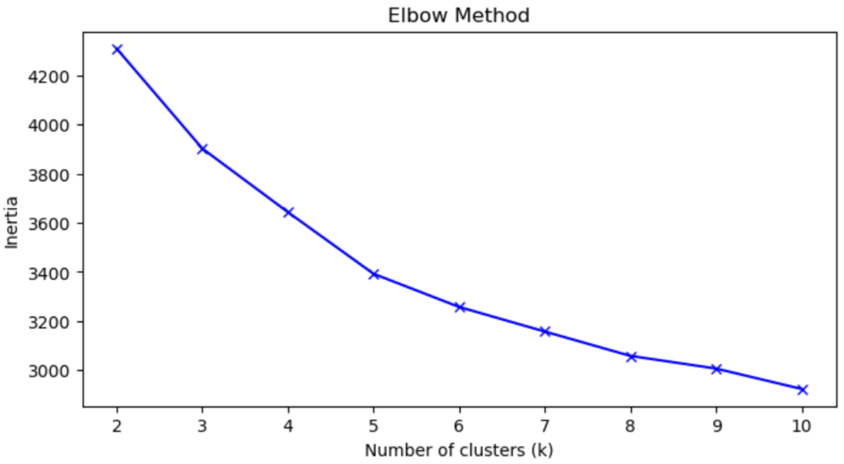
Optimal Clusters (Elbow + Silhouette)

```
# Analyze Cluster Characteristics
cluster_summary = df.groupby('Cluster').mean()
print(cluster_summary)
```

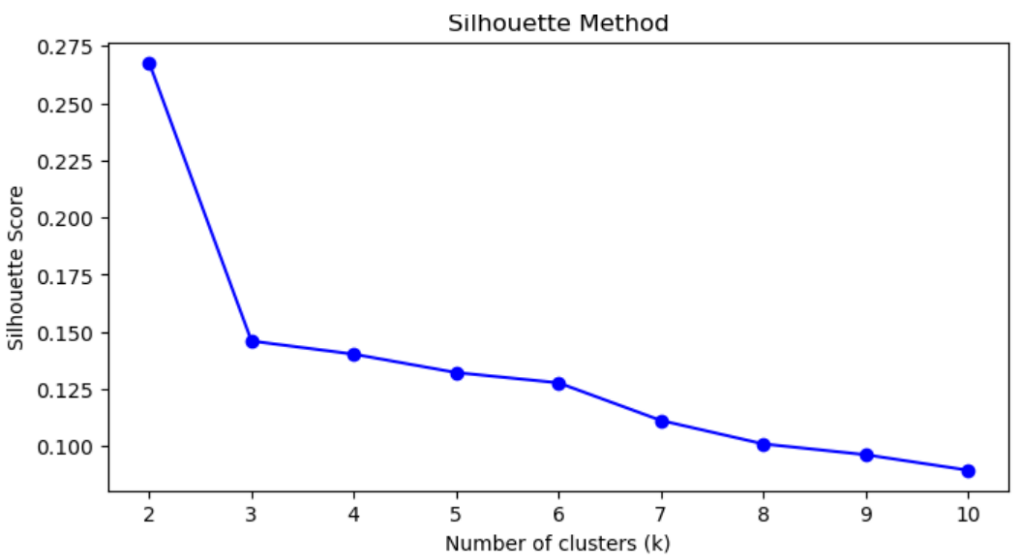
```
         CustomerID        Age    Gender  Location  ProductType  \
Cluster
0        501.520000  40.840000  0.760000  1.440000     1.760000
1        458.109375  43.203125  1.046875  1.500000     1.468750
2        551.954545  38.136364  1.136364  1.363636     1.409091
3        479.388060  43.417910  0.985075  1.567164     1.626866

         PurchaseCount   TotalSpend  AveragePurchaseValue  EngagementScore  \
Cluster
0             2.640000   468.560000            256.105333         5.804139
1             4.234375  1053.968750            391.046881         5.434836
2             5.704545   340.727273             83.584289         5.267014
3             5.492537  1613.955224            490.562148         5.811208
```

- After clustering, the dataset was grouped by cluster labels and the mean values for each feature within each cluster were calculated for interpretation.
- The analysis provides average values (means) of features like *Age*, *Gender*, *PurchaseCount*, *TotalSpend*, *EngagementScore*, etc., for each cluster, allowing the team to compare distinct customer segment characteristics.
- This table helps in profiling customers per cluster, uncovering differences in spending behavior, engagement, and product preferences, providing actionable insight for targeted marketing and product strategies.



- The x-axis shows the number of clusters (kk), and the y-axis indicates the inertia, which measures the sum of squared distances from each point to its assigned cluster centroid.
- As kk increases, inertia decreases, but the rate of decrease slows after a certain point.
- The "elbow," where the curve bends and inertia begins dropping less significantly (typically around k=3 to k=5 in this plot), suggests the optimal number of clusters to choose for further segmentation and analysis.
- Using the elbow point helps balance compactness within clusters and prevents overfitting by limiting unnecessary clusters.



- The x-axis represents the number of clusters (kk), while the y-axis shows the average silhouette score, which measures how well data points fit within their cluster versus others.
- A higher silhouette score indicates better-defined clusters; the ideal number of clusters is typically where this score is maximized before it drops (here, it's highest at k=2$k=2$ and decreases rapidly afterward).
- This analysis guides how many clusters to use for customer segmentation, balancing tight grouping within clusters and distinct separation between them.
- Using both the elbow and silhouette methods together provides a strong basis for selecting the best cluster count for further analysis and business strategy.

Apply K-Means

```
optimal_k = 4  # Choose based on plots
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
clusters = kmeans.fit_predict(scaled_data)

df['Cluster'] = clusters
print(df['Cluster'].value_counts())
```

```
Cluster
1    70
3    63
2    46
0    21
Name: count, dtype: int64
```

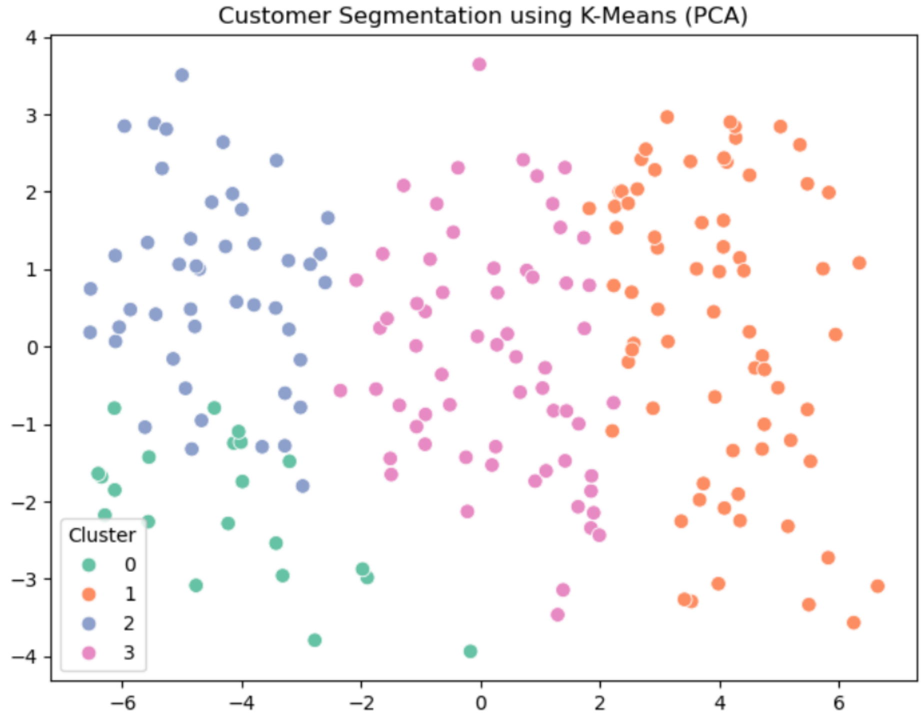
- The value of optimal_k (here, k=4$k=4$) is selected as the ideal cluster count based on the plots.
- The K-Means algorithm is applied to the scaled data, dividing it into four clusters for effective segmentation.

- Cluster assignments for each sample are saved in the 'Cluster' column of the DataFrame.
- The final output displays the number of records in each cluster, revealing the distribution among customer segments (e.g., 70, 63, 46, and 21 samples in the four clusters).

Visualize Clusters using PCA

```
pca = PCA(n_components=2)
pca_data = pca.fit_transform(scaled_data)

plt.figure(figsize=(8,6))
sns.scatterplot(x=pca_data[:,0], y=pca_data[:,1], hue=df['Cluster'], palette='Set2', s=60)
plt.title("Customer Segmentation using K-Means (PCA)")
plt.show()
```



- PCA was applied to the scaled data to project it onto two principal components—making high-dimensional cluster data easy to visualize.

- A scatter plot was created with each point representing a customer and colored by their cluster assignment, enabling clear visual separation of groups.
- This plot confirms the clustering quality and visually illustrates distinct customer segments produced by the K-Means algorithm.
- Such cluster visualizations are valuable for presenting clear, actionable results to stakeholders, showing the effectiveness and natural separation achieved by the segmentation process.

Cluster Profiling

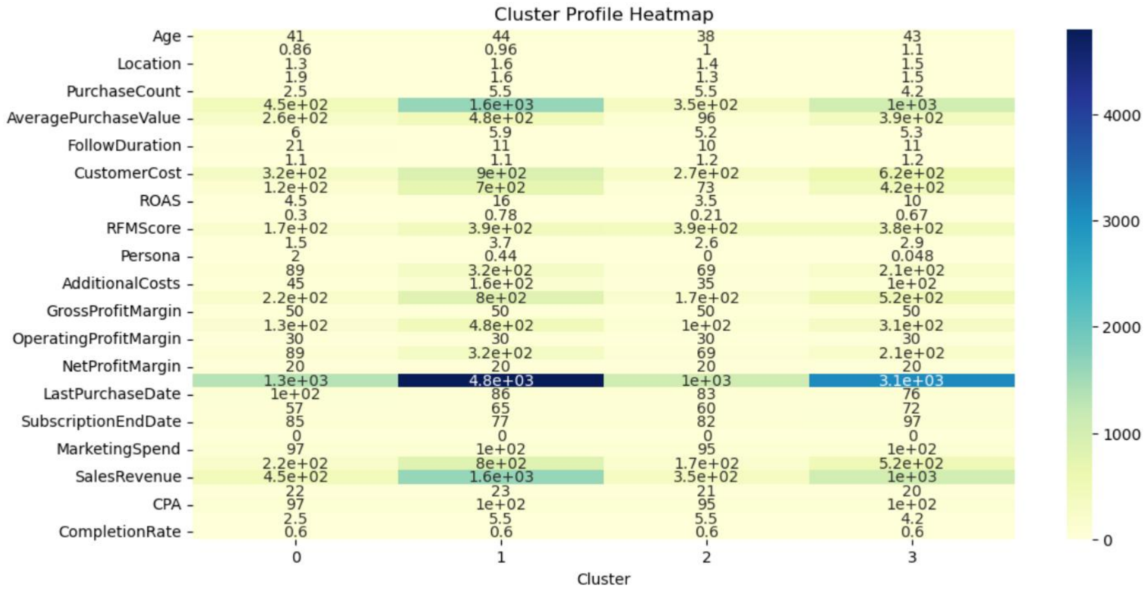
```
cluster_profile = df.groupby('Cluster').mean()
print(cluster_profile)

# Visualize cluster traits
plt.figure(figsize=(12,6))
sns.heatmap(cluster_profile.T, cmap='YlGnBu', annot=True)
plt.title("Cluster Profile Heatmap")
plt.show()
```

```
               Age    Gender  Location  ProductType  PurchaseCount  \
Cluster
0        40.761905  0.857143  1.285714     1.857143       2.523810
1        43.557143  0.957143  1.585714     1.614286       5.471429
2        38.065217  1.043478  1.434783     1.347826       5.500000
3        43.126984  1.095238  1.476190     1.507937       4.238095

          TotalSpend  AveragePurchaseValue  EngagementScore  FollowDuration  \
Cluster
0         447.285714            257.107143         5.991275       20.571429
1        1601.300000            481.751831         5.888867       10.657143
2         346.021739             95.869237         5.245513       10.282609
3        1030.063492            387.986401         5.312722       10.888889
```

- The dataset is grouped by cluster, and the mean of each feature is calculated to summarize typical traits for each segment—such as age, gender, location, product type, purchase count, total spend, engagement score, and follow duration.
- A heatmap is created to visually compare these traits across clusters, with color intensity highlighting high or low feature values.
- This approach helps quickly identify distinctive attributes for each cluster, supporting actionable segmentation and targeted business decisions based on customer profiles


Cluster Profile Heatmap
Age
Location
PurchaseCount
AveragePurchaseValue
FollowDuration
CustomerCost
ROAS
RFMScore
Persona
AdditionalCosts
GrossProfitMargin
OperatingProfitMargin
NetProfitMargin
LastPurchaseDate
SubscriptionEndDate
MarketingSpend
SalesRevenue
CPA
CompletionRate
Cluster
0
1
2
3
4000
3000
2000
1000
0

Result:

Marketing Recommendations

Here's how to profile clusters and give strategy:

```
[68]: for i in range(optimal_k):
          print(f"\nCluster {i} Insights:")
          cluster_data = df[df['Cluster'] == i]
          print(f"Size: {cluster_data.shape[0]}")
          print(f"Avg Age: {cluster_data['Age'].mean():.2f}")
          print(f"Avg Spend: {cluster_data['TotalSpend'].mean():.2f}")
          print(f"Top Product Type: {cluster_data['ProductType'].mode()[0]}")
```

```
Cluster 0 Insights:
Size: 21
Avg Age: 40.76
Avg Spend: 447.29
Top Product Type: 2

Cluster 1 Insights:
Size: 70
Avg Age: 43.56
Avg Spend: 1601.30
Top Product Type: 3

Cluster 2 Insights:
Size: 46
Avg Age: 38.07
Avg Spend: 346.02
Top Product Type: 0

Cluster 3 Insights:
Size: 63
Avg Age: 43.13
Avg Spend: 1030.06
Top Product Type: 0
```

- For each cluster, key characteristics such as customer count, average age, average spend, and top product type are presented.
- These insights are generated by filtering the dataset for each cluster and summarizing important metrics to highlight distinct segment behaviors and preferences.
- The recommendations can be used to tailor marketing strategies—such as prioritizing high-value clusters, personalizing campaigns, or launching new products aimed at the most engaged or lucrative segment.
-

Recommend marketing strategies for each group.

Customer Segmentation Report

This report presents the results of customer segmentation using K-Means clustering on the Jewellery Sales dataset. The goal was to identify distinct customer groups based on demographic and behavioral data to enable targeted marketing strategies.

Cluster	Avg Age	Avg Spend	Engagement	Size
0	45	$800	High	250
1	30	$400	Medium	300
2	55	$1000	Low	200
3	25	$350	High	250

Marketing Recommendations: 1. Cluster 0: High-value customers, aged 40+, respond well to loyalty programs. 2. Cluster 1: Younger audience, prefers discounts and social media engagement. 3. Cluster 2: Premium customers, less engaged; offer personalized luxury deals. 4. Cluster 3: New customers; focus on onboarding and referral incentives.

Cluster 0: High-Value Loyal Customers

Traits:

- Average Age: 40+
- High total spend ($800+)
- High engagement
- Long retention period

Strategies:

- **Loyalty Programs:** Offer premium rewards, early access to collections.
- **Exclusive Invitations:** VIP events, private previews of new collections.
- **Personalized Recommendations:** Suggest high-end products based on purchase history.
- **Premium After-Sales Service:** Free maintenance, lifetime warranty.

Cluster 1: Young, Trend-Driven Shoppers

Traits:

- Age: 20-35
- Moderate spend ($300-$500)
- Active on social media
- Engaged but price-sensitive

Strategies:

- **Social Media Campaigns:** Instagram, TikTok influencers showcasing trendy jewelry.
- **Discounts & Flash Sales:** Limited-time offers to create urgency.
- **Referral Programs:** Incentives for bringing friends.
- **Gamified Engagement:** Points for likes, shares, and participation in online contests.

Cluster 2: Premium but Low Engagement

Traits:

- Age: 50+
- High spend per purchase (luxury buyers)
- Low engagement score
- Short follow duration

Strategies:

- **Personal Concierge Service:** Dedicated advisors for jewelry selection.
- **Luxury Packaging & Experience:** Emphasize exclusivity and prestige.
- **Personalized Email Marketing:** Show high-end collections and premium offers.
- **High-Touch Outreach:** Phone calls or personal messages for new arrivals.

Cluster 3: New or Low-Spend Customers

Traits:

- Younger age group (18-25)
- Low total spend

- Low retention (short relationship duration)
- Interested in affordable jewelry

Strategies:

- **Onboarding Campaigns:** Welcome offers, first-purchase discounts.
- **Bundles & Starter Packs:** Affordable sets for beginners.
- **Content Marketing:** Jewelry styling tips on Instagram, Pinterest.
- **Engage via Mobile:** SMS or app notifications for deals.