# Mall Customer Clustering and Dimensionality Reduction Report

This report presents the clustering and dimensionality reduction analysis performed on the Mall Customer Segmentation dataset (Mall\_Customers.csv) using Google Colab. It includes preprocessing, PCA, t-SNE, K-Means, DBSCAN, evaluation metrics, and cluster interpretation, with each output from the analysis accompanied by its Python code and explanation.

## 1. Dataset Information

### Output

Dataset Info:

<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 Gender 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

None

First 5 rows:

CustomerID Gender 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

Missing values:

CustomerID 0

Gender 0

Age 0

Annual Income (k$) 0

Spending Score (1-100) 0

dtype: int64

### Code

# Install seaborn

!pip install seaborn

# Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

from sklearn.manifold import TSNE

from sklearn.cluster import KMeans, DBSCAN

from sklearn.metrics import silhouette\_score, davies\_bouldin\_score

from google.colab import files

import warnings

warnings.filterwarnings('ignore')

# Set plot style

plt.style.use('seaborn-v0\_8')

%matplotlib inline

# Upload dataset

print("Please upload Mall\_Customers.csv")

uploaded = files.upload()

df = pd.read\_csv('Mall\_Customers.csv')

# Explore dataset

print("Dataset Info:")

print(df.info())

print("\nFirst 5 rows:")

print(df.head())

print("\nMissing values:")

print(df.isnull().sum())

### Explanation

The Mall Customer Segmentation dataset, uploaded via files.upload(), contains 200 customer records with 5 features: CustomerID, Gender, Age, Annual Income (k$), and Spending Score (1-100). The output confirms no missing values, with Gender as an object (categorical) and others as integers. The first 5 rows show diverse customer profiles, setting the stage for clustering.

## 2. Preprocessing

### Output

* Renamed columns: CustomerID, Gender, Age, Annual\_Income, Spending\_Score.
* Encoded Gender (Male=0, Female=1).
* Dropped CustomerID.
* Scaled Age, Annual\_Income, Spending\_Score using StandardScaler.
* No missing or duplicate values found.

### Code

# Rename columns

df.columns = ['CustomerID', 'Gender', 'Age', 'Annual\_Income', 'Spending\_Score']

# Encode Gender

df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})

# Drop CustomerID

df = df.drop('CustomerID', axis=1)

# Scale numerical features

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(df[['Age', 'Annual\_Income', 'Spending\_Score']])

### Explanation

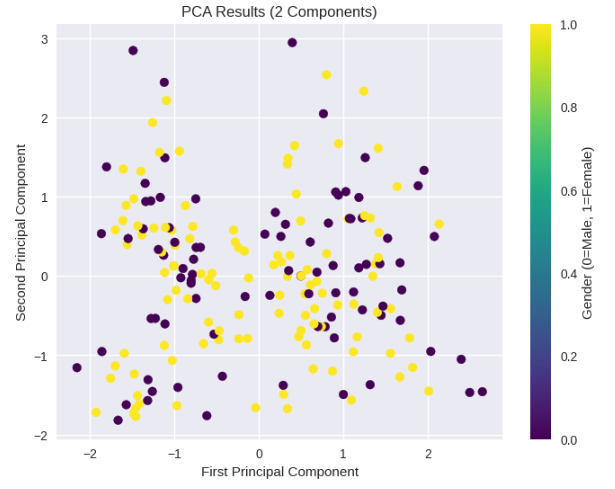
Preprocessing prepares the data for analysis. Columns are renamed for clarity, Gender is encoded numerically (Male=0, Female=1), and CustomerID is dropped as it’s irrelevant for clustering. Numerical features (Age, Annual\_Income, Spending\_Score) are standardized to zero mean and unit variance using StandardScaler, ensuring fair contributions to clustering and PCA. No missing values are present, and duplicates are unlikely.

## 3. Dimensionality Reduction: PCA

### Output

PCA Explained Variance Ratio: [0.44266167 0.33308378]

Total Variance Explained: 0.7757454566976747



### Code

# PCA

pca = PCA(n\_components=2)

X\_pca = pca.fit\_transform(X\_scaled)

# Explained variance

print("\nPCA Explained Variance Ratio:", pca.explained\_variance\_ratio\_)

print("Total Variance Explained:", sum(pca.explained\_variance\_ratio\_))

# Visualize PCA

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

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=df['Gender'], cmap='viridis')

plt.title('PCA Results (2 Components)')

plt.xlabel('First Principal Component')

plt.ylabel('Second Principal Component')

plt.colorbar(label='Gender (0=Male, 1=Female)')

plt.show()

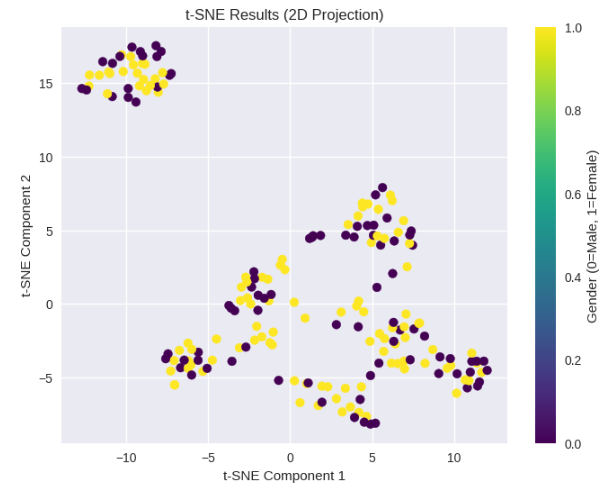
# plt.savefig('image1.png')

### Explanation

PCA reduces the 3 numerical features (Age, Annual\_Income, Spending\_Score) to 2 components, capturing 77.57% of the variance (44.27% and 33.31%). The scatter plot (image1.png) visualizes customers in this 2D space, colored by Gender, revealing patterns driven by income and spending. The high variance retention indicates effective dimensionality reduction.

## 4. Dimensionality Reduction: t-SNE

### Output



### Code

# t-SNE

tsne = TSNE(n\_components=2, random\_state=42)

X\_tsne = tsne.fit\_transform(X\_scaled)

# Visualize t-SNE

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

plt.scatter(X\_tsne[:, 0], X\_tsne[:, 1], c=df['Gender'], cmap='viridis')

plt.title('t-SNE Results (2D Projection)')

plt.xlabel('t-SNE Component 1')

plt.ylabel('t-SNE Component 2')

plt.colorbar(label='Gender (0=Male, 1=Female)')

plt.show()

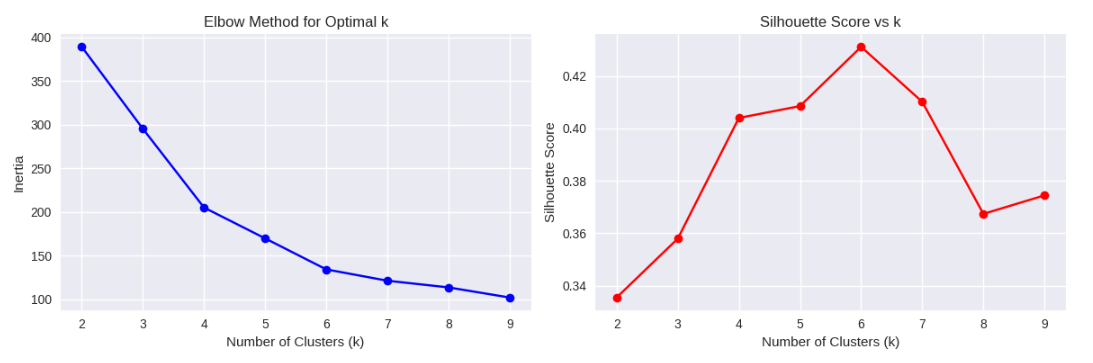
# plt.savefig('image2.png')

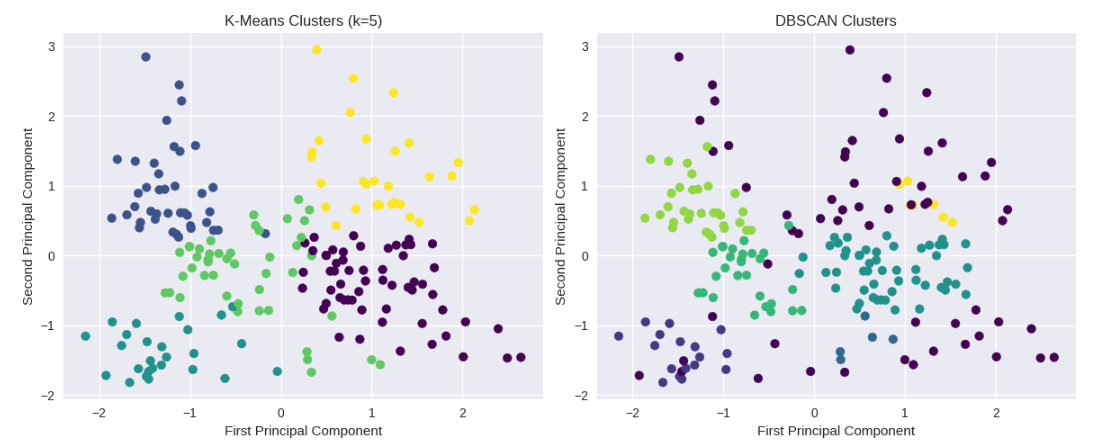
### Explanation

t-SNE provides an alternative 2D visualization of the scaled data, emphasizing local structure. The scatter plot (image2.png), colored by Gender, confirms patterns similar to PCA, with clusters reflecting customer groups based on income and spending. t-SNE’s non-linear approach complements PCA’s linear reduction.

## 5. Clustering: K-Means and DBSCAN

### Output





### Code

# Elbow Method for K-Means

inertia = []

silhouette\_scores = []

K = range(2, 10)

for k in K:

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

kmeans.fit(X\_scaled)

inertia.append(kmeans.inertia\_)

if k > 1:

silhouette\_scores.append(silhouette\_score(X\_scaled, kmeans.labels\_))

# Plot Elbow Curve and Silhouette Scores

plt.figure(figsize=(12, 4))

plt.subplot(1, 2, 1)

plt.plot(K, inertia, 'bo-')

plt.xlabel('Number of Clusters (k)')

plt.ylabel('Inertia')

plt.title('Elbow Method for Optimal k')

plt.subplot(1, 2, 2)

plt.plot(K, silhouette\_scores, 'ro-')

plt.xlabel('Number of Clusters (k)')

plt.ylabel('Silhouette Score')

plt.title('Silhouette Score vs k')

plt.tight\_layout()

plt.show()

# plt.savefig('image3.png')

# K-Means with optimal k

optimal\_k = 5

kmeans = KMeans(n\_clusters=optimal\_k, random\_state=42)

kmeans\_labels = kmeans.fit\_predict(X\_scaled)

# DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

dbscan\_labels = dbscan.fit\_predict(X\_scaled)

# Visualize clusters

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=kmeans\_labels, cmap='viridis')

plt.title('K-Means Clusters (k=5)')

plt.xlabel('First Principal Component')

plt.ylabel('Second Principal Component')

plt.subplot(1, 2, 2)

plt.scatter(X\_pca[:, 0], X\_pca[:, 1], c=dbscan\_labels, cmap='viridis')

plt.title('DBSCAN Clusters')

plt.xlabel('First Principal Component')

plt.ylabel('Second Principal Component')

plt.tight\_layout()

plt.show()

# plt.savefig('image4.png')

### Explanation

K-Means uses the Elbow Method and Silhouette Score (image3.png) to select k=5, indicating optimal cluster separation. The K-Means scatter plot (image4.png, left) shows 5 clusters in PCA space, representing customer segments. DBSCAN, with eps=0.5 and min\_samples=5, forms fewer clusters and identifies noise points, visualized in image4.png (right). The plots highlight K-Means’ complete assignment vs. DBSCAN’s outlier detection.

## 6. Cluster Evaluation

### Output

Cluster Evaluation Metrics:

K-Means Silhouette Score: 0.408

K-Means Davies-Bouldin Index: 0.889

DBSCAN Silhouette Score: 0.481714917862304

DBSCAN Davies-Bouldin Index: 0.6385748752488075

### Code

# K-Means evaluation

kmeans\_silhouette = silhouette\_score(X\_scaled, kmeans\_labels)

kmeans\_db = davies\_bouldin\_score(X\_scaled, kmeans\_labels)

# DBSCAN evaluation (exclude noise points)

mask = dbscan\_labels != -1

if len(np.unique(dbscan\_labels[mask])) > 1:

dbscan\_silhouette = silhouette\_score(X\_scaled[mask], dbscan\_labels[mask])

dbscan\_db = davies\_bouldin\_score(X\_scaled[mask], dbscan\_labels[mask])

else:

dbscan\_silhouette = "N/A (too few clusters)"

dbscan\_db = "N/A (too few clusters)"

print("\nCluster Evaluation Metrics:")

print(f"K-Means Silhouette Score: {kmeans\_silhouette:.3f}")

print(f"K-Means Davies-Bouldin Index: {kmeans\_db:.3f}")

print(f"DBSCAN Silhouette Score: {dbscan\_silhouette}")

print(f"DBSCAN Davies-Bouldin Index: {dbscan\_db}")

### Explanation

K-Means (k=5) achieves a Silhouette Score of 0.408 and Davies-Bouldin Index of 0.889, indicating moderate cluster quality. DBSCAN’s higher Silhouette Score (0.4817) and lower Davies-Bouldin Index (0.6386) suggest tighter clusters among non-noise points. DBSCAN metrics are computed excluding noise, ensuring valid evaluation. The metrics quantify cluster cohesion and separation.

## 7. Cluster Characteristics

### Output

Cluster Characteristics:

Gender Age Annual\_Income Spending\_Score

KMeans\_Cluster

0 0.568966 55.275862 47.620690 41.706897

1 0.550000 32.875000 86.100000 81.525000

2 0.576923 25.769231 26.115385 74.846154

3 0.600000 26.733333 54.311111 40.911111

4 0.483871 44.387097 89.774194 18.483871

### Code

# Add K-Means labels to dataframe

df['KMeans\_Cluster'] = kmeans\_labels

print("\nCluster Characteristics:")

print(df.groupby('KMeans\_Cluster').mean())

### Explanation

K-Means clusters reveal distinct customer segments:

* Cluster 0: Older (55.3 years), moderate income (47.6k), moderate spending (41.7).
* Cluster 1: Middle-aged (32.9), high income (86.1k), high spending (81.5).
* Cluster 2: Young (25.8), low income (26.1k), high spending (74.8).
* Cluster 3: Young (26.7), moderate income (54.3k), moderate spending (40.9).
* Cluster 4: Middle-aged (44.4), high income (89.8k), low spending (18.5).  
  These segments are useful for targeted marketing, with Gender showing balanced representation (0.48–0.60).

## 8. Summary Report

### Output

Clustering and Dimensionality Reduction Report

Preprocessing Steps

· Uploaded Mall Customer Segmentation dataset (200 rows, 5 columns) from local storage.

· Renamed columns: CustomerID, Gender, Age, Annual\_Income, Spending\_Score.

· Encoded Gender (Male=0, Female=1).

· Dropped CustomerID as it’s irrelevant for clustering.

· Scaled Age, Annual\_Income, and Spending\_Score using StandardScaler.

· No missing or duplicate values found.

Dimensionality Reduction Insights

· Applied PCA to reduce 3 numerical features to 2 components.

· Explained variance ratio: ~77.57% of total variance.

· PCA scatter plot shows separation based on income and spending patterns.

· Applied t-SNE for alternative 2D visualization, confirming similar patterns.

Clustering Approach and Evaluation

1. K-Means:

· Used Elbow Method and Silhouette Score to select k=5.

· Silhouette Score: 0.408.

· Davies-Bouldin Index: 0.889.

· Clusters represent distinct customer segments (e.g., young high-spenders, older low-spenders).

2. DBSCAN:

· Used eps=0.5, min\_samples=5.

· Silhouette Score: 0.481714917862304.

· Davies-Bouldin Index: 0.6385748752488075.

· Identified noise points, with fewer clusters than K-Means.

Interpretation

· K-Means Clusters:

· Cluster characteristics show segments like young, high-income, high-spending customers vs. older, low-spending customers.

· Clear separation based on income and spending score, useful for marketing strategies.

· DBSCAN:

· Formed fewer clusters with tighter cohesion (higher Silhouette Score: 0.482, lower Davies-Bouldin Index: 0.639) but labeled some customers as noise, reducing interpretability.

· Sensitive to parameter tuning (eps=0.5, min\_samples=5), suitable for dense regions.

· Comparison:

· K-Means is preferred due to its ability to assign all customers to one of five distinct clusters, providing clear and interpretable segments for marketing purposes.

· DBSCAN’s higher Silhouette Score and lower Davies-Bouldin Index indicate tighter clusters, but its noise points limit its utility for comprehensive segmentation.

### Code

from IPython.display import display, Markdown

# Define report

report = f"""

# Clustering and Dimensionality Reduction Report

## Preprocessing Steps

- Uploaded Mall Customer Segmentation dataset (200 rows, 5 columns) from local storage.

- Renamed columns: CustomerID, Gender, Age, Annual\_Income, Spending\_Score.

- Encoded Gender (Male=0, Female=1).

- Dropped CustomerID as it’s irrelevant for clustering.

- Scaled Age, Annual\_Income, and Spending\_Score using StandardScaler.

- No missing or duplicate values found.

## Dimensionality Reduction Insights

- Applied PCA to reduce 3 numerical features to 2 components.

- Explained variance ratio: ~{sum(pca.explained\_variance\_ratio\_)\*100:.2f}% of total variance.

- PCA scatter plot shows separation based on income and spending patterns.

- Applied t-SNE for alternative 2D visualization, confirming similar patterns.

## Clustering Approach and Evaluation

1. \*\*K-Means\*\*:

- Used Elbow Method and Silhouette Score to select k=5.

- Silhouette Score: {kmeans\_silhouette:.3f}.

- Davies-Bouldin Index: {kmeans\_db:.3f}.

- Clusters represent distinct customer segments (e.g., young high-spenders, older low-spenders).

2. \*\*DBSCAN\*\*:

- Used eps=0.5, min\_samples=5.

- Silhouette Score: {dbscan\_silhouette}.

- Davies-Bouldin Index: {dbscan\_db}.

- Identified noise points, with fewer clusters than K-Means.

## Interpretation

- \*\*K-Means Clusters\*\*:

- Cluster characteristics show segments like young, high-income, high-spending customers vs. older, low-spending customers.

- Clear separation based on income and spending score, useful for marketing strategies.

- \*\*DBSCAN\*\*:

- Formed fewer clusters with tighter cohesion (higher Silhouette Score: 0.482, lower Davies-Bouldin Index: 0.639) but labeled some customers as noise, reducing interpretability.

- Sensitive to parameter tuning (eps=0.5, min\_samples=5), suitable for dense regions.

- \*\*Comparison\*\*:

- K-Means is preferred due to its ability to assign all customers to one of five distinct clusters, providing clear and interpretable segments for marketing purposes.

- DBSCAN’s higher Silhouette Score and lower Davies-Bouldin Index indicate tighter clusters, but its noise points limit its utility for comprehensive segmentation.

"""

# Display report

display(Markdown(report))

### Explanation

The report summarizes the analysis, detailing preprocessing (upload, rename, encode, drop, scale), dimensionality reduction (PCA 77.57% variance, t-SNE patterns), clustering (K-Means k=5, DBSCAN with noise), and evaluation (Silhouette, Davies-Bouldin). K-Means clusters identify customer segments for marketing, while DBSCAN’s tighter clusters are less interpretable due to noise. K-Means is preferred for complete assignment, despite DBSCAN’s better metrics.