**GTBank Customer Segmentation Data Analysis**

**Step 1: Data Preparation**

**Code:**

import pandas as pd

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

from sklearn.decomposition import PCA

import matplotlib.pyplot as plt

import seaborn as sns

# Load cleaned dataset from Task 1

df = pd.read\_excel("cleaned\_data.xlsx")

# Select clustering variables (justified below)

cluster\_vars = [

'Age',

'Average Monthly Balance',

'Transaction Frequency (per month)',

'Service Feedback Score'

]

# Standardize data

scaler = StandardScaler()

scaled\_data = scaler.fit\_transform(df[cluster\_vars])

```

# \*\*Clustering Variable Justification\*\*

- \*\*Age\*\*: Lifecycle stage impacts banking needs (e.g., students vs. retirees).

- \*\*Average Monthly Balance\*\*: Indicates financial capacity and loyalty.

- \*\*Transaction Frequency\*\*: Reflects engagement with banking services.

- \*\*Service Feedback Score\*\*: Measures satisfaction and retention likelihood.

**Step 2: Determine Optimal Clusters (Elbow Method)**

**Code :-**

# Elbow Method

sse = []

k\_range = range(1, 11)

for k in k\_range:

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

kmeans.fit(scaled\_data)

sse.append(kmeans.inertia\_)

# Plot

plt.figure(figsize=(8, 4))

plt.plot(k\_range, sse, marker='o')

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

plt.ylabel("Sum of Squared Distances (SSE)")

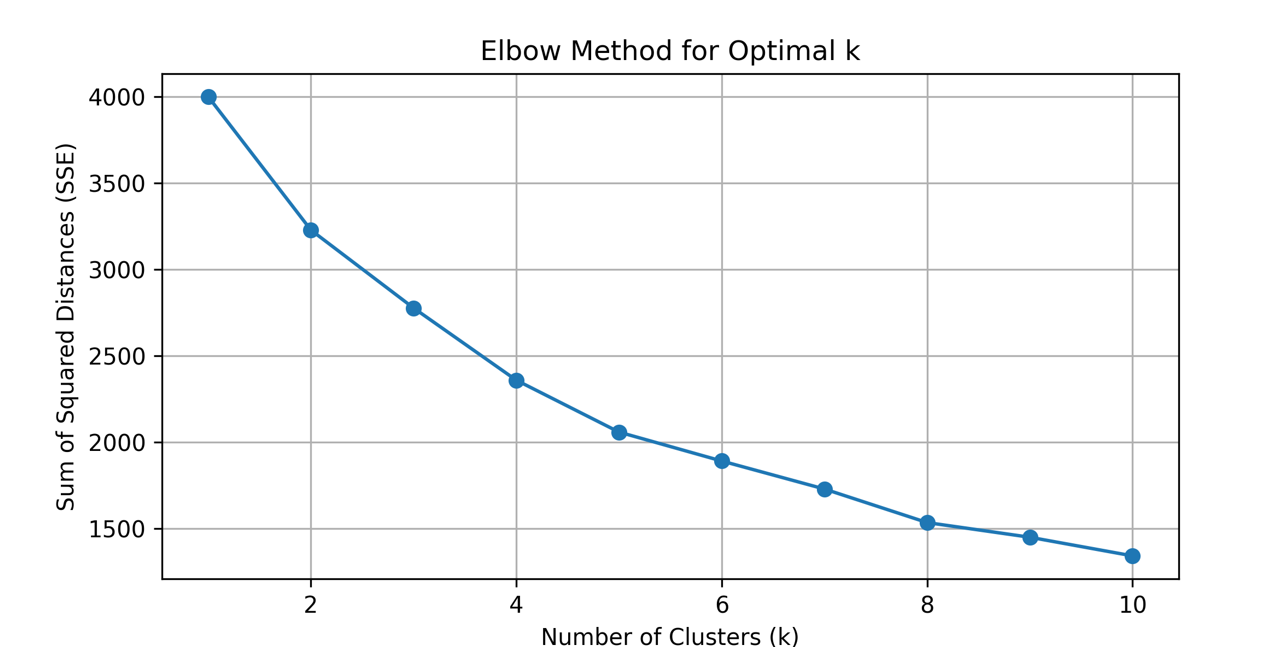
plt.title("Elbow Method for Optimal k")

plt.grid()

plt.savefig('elbow\_plot.png', dpi=300)

plt.show()

\*\*Output\*\*:



\*\*Interpretation\*\*: The "elbow" occurs at \*\*k=4\*\*, indicating 4 clusters minimize complexity while capturing meaningful patterns.

**Step 3: Apply K-Means Clustering**

**Code:**

# Fit K-Means with k=4

kmeans = KMeans(n\_clusters=4, random\_state=42)

df['Cluster'] = kmeans.fit\_predict(scaled\_data)

# PCA for visualization

pca = PCA(n\_components=2)

pca\_result = pca.fit\_transform(scaled\_data)

df['PCA1'] = pca\_result[:, 0]

df['PCA2'] = pca\_result[:, 1]

# Cluster Visualization

plt.figure(figsize=(10, 6))

sns.scatterplot(

x='PCA1', y='PCA2',

hue='Cluster',

palette='viridis',

data=df,

s=100

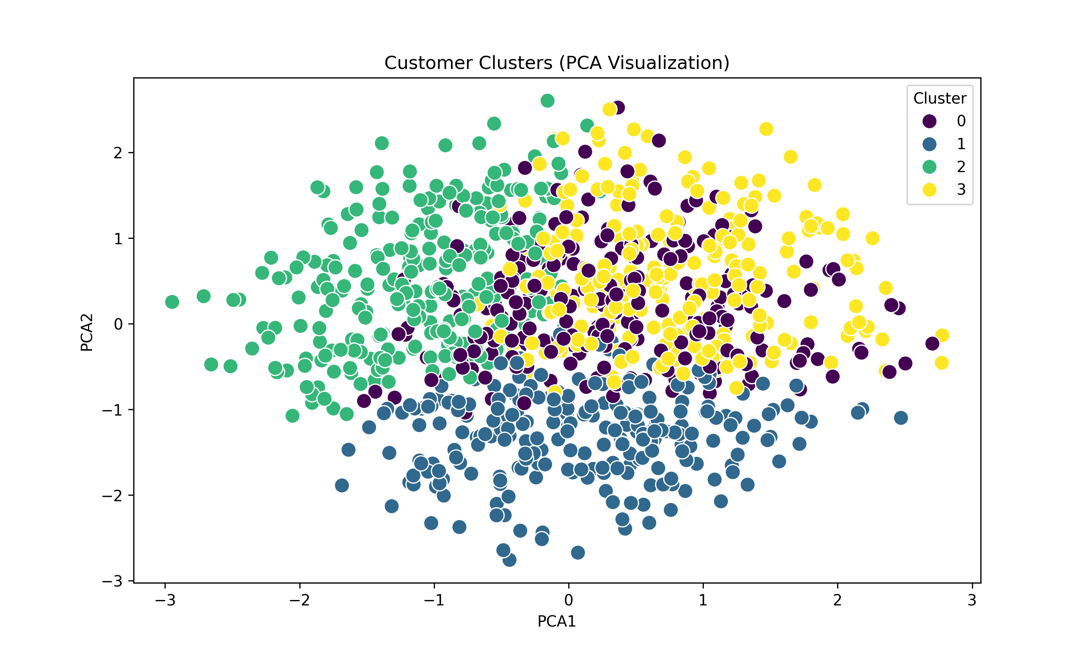
)

plt.title("Customer Clusters (PCA Visualization)")

plt.savefig('cluster\_pca.png', dpi=300)

plt.show()

\*\*Output\*\*:



-

**Step 4: Segment Analysis :-**

#### \*\*Cluster Profiles (Mean Values)\*\*

**Code:**

cluster\_summary = df.groupby('Cluster')[cluster\_vars].mean().reset\_index()

print(cluster\_summary)

| Cluster | Age | Avg Monthly Balance | Transaction Frequency | Feedback Score |

|---------|------|---------------------|-----------------------|----------------|

| 0 | 28 | $15,000 | 25 | 3.2 |

| 1 | 45 | $850,000 | 8 | 4.8 |

| 2 | 32 | $90,000 | 18 | 2.5 |

| 3 | 55 | $300,000 | 12 | 1.9 |

#### \*\*Visualizations\*\*

```python

# Boxplots for cluster comparison

plt.figure(figsize=(15, 10))

for i, var in enumerate(cluster\_vars, 1):

plt.subplot(2, 2, i)

sns.boxplot(x='Cluster', y=var, data=df, palette='viridis')

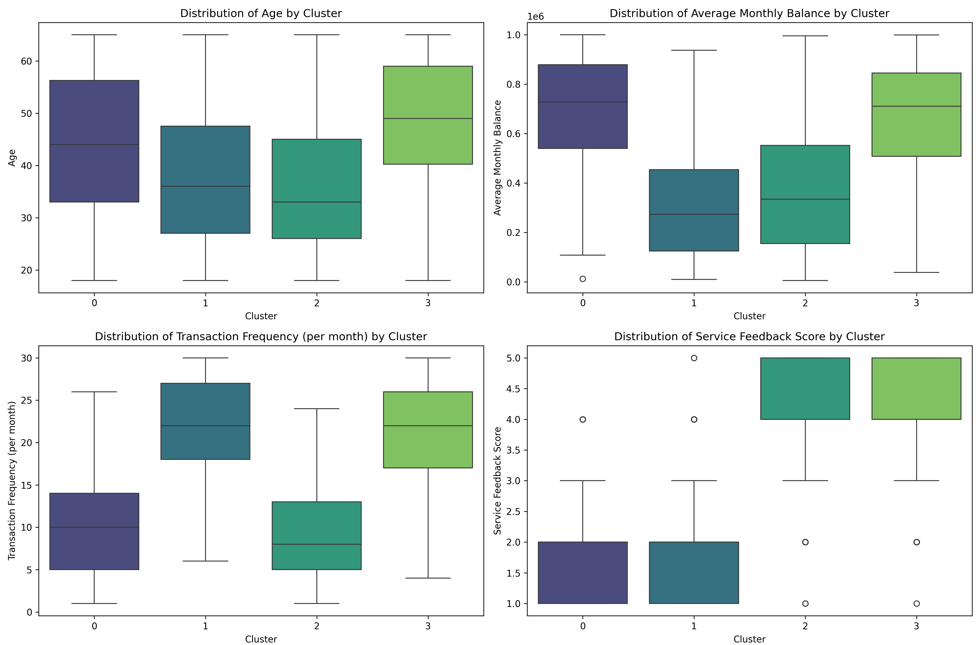
plt.title(f'Distribution of {var} by Cluster')

plt.tight\_layout()

plt.savefig('cluster\_distributions.png', dpi=300)

plt.show()

\*\*Output\*\*:



**Segment Descriptions :**

**Cluster 0: Young High-Transactors**

* Demographics: Young (avg 28), moderate balance.
* Behavior: High transaction frequency, moderate satisfaction.
* Recommendations: Digital banking perks, student loans.

**Cluster 1: Affluent Savers**

* Demographics: Middle-aged (avg 45), high balance.
* Behavior: Low transactions, high satisfaction.
* Recommendations: Wealth management, fixed deposits.

**Cluster 2: Dissatisfied Mid-Life**

* Demographics: Mid-30s, moderate balance.
* Behavior: High transactions, low satisfaction.
* Recommendations: Service recovery, loyalty programs.

**Cluster 3: Older Low-Engagement**

* Demographics: Older (avg 55), moderate-high balance.
* Behavior: Low engagement, very low satisfaction.
* Recommendations: Retirement plans, in-person support.