In [1]:

```
# Import necessecary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
```

In [2]:

```
# Load the dataset
data = pd.read_csv('Wholesale customers data.csv')
data.head()
```

Out[2]:

	Channel	Region	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
0	2	3	12669	9656	7561	214	2674	1338
1	2	3	7057	9810	9568	1762	3293	1776
2	2	3	6353	8808	7684	2405	3516	7844
3	1	3	13265	1196	4221	6404	507	1788
4	2	3	22615	5410	7198	3915	1777	5185

In [3]:

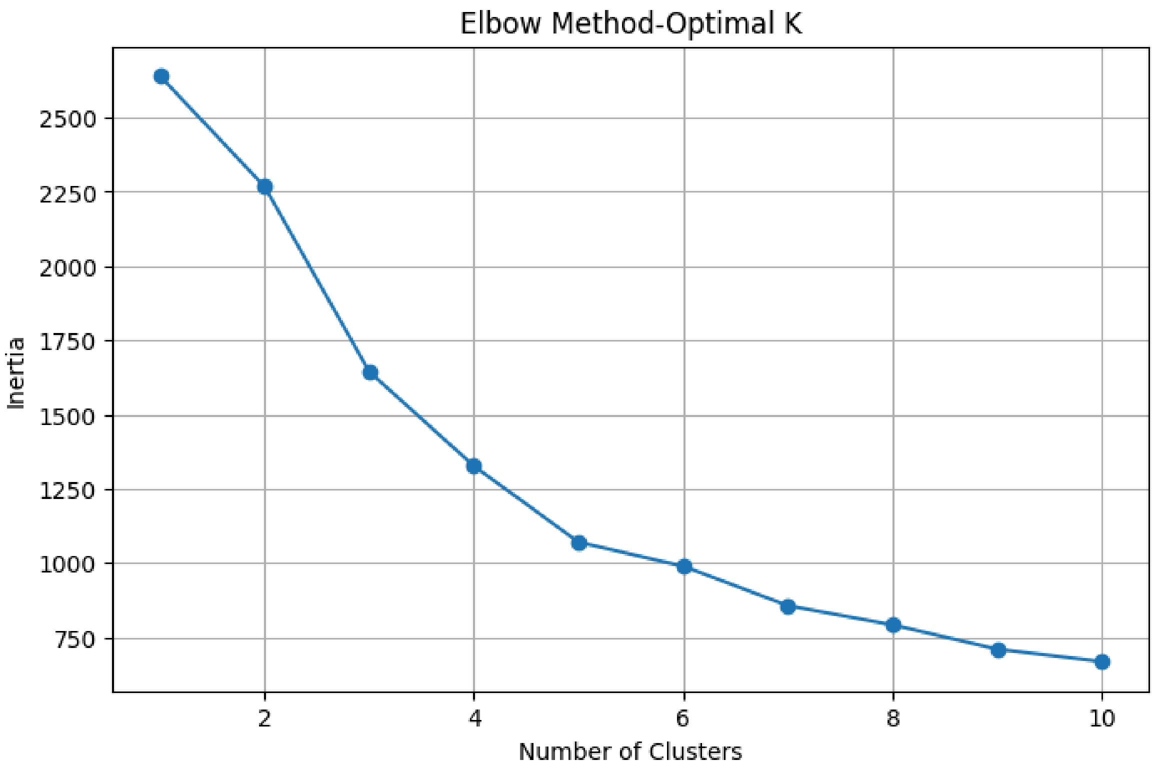
```
# Normalize the data
X = data.drop(['Channel','Region'], axis=1)

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [5]:

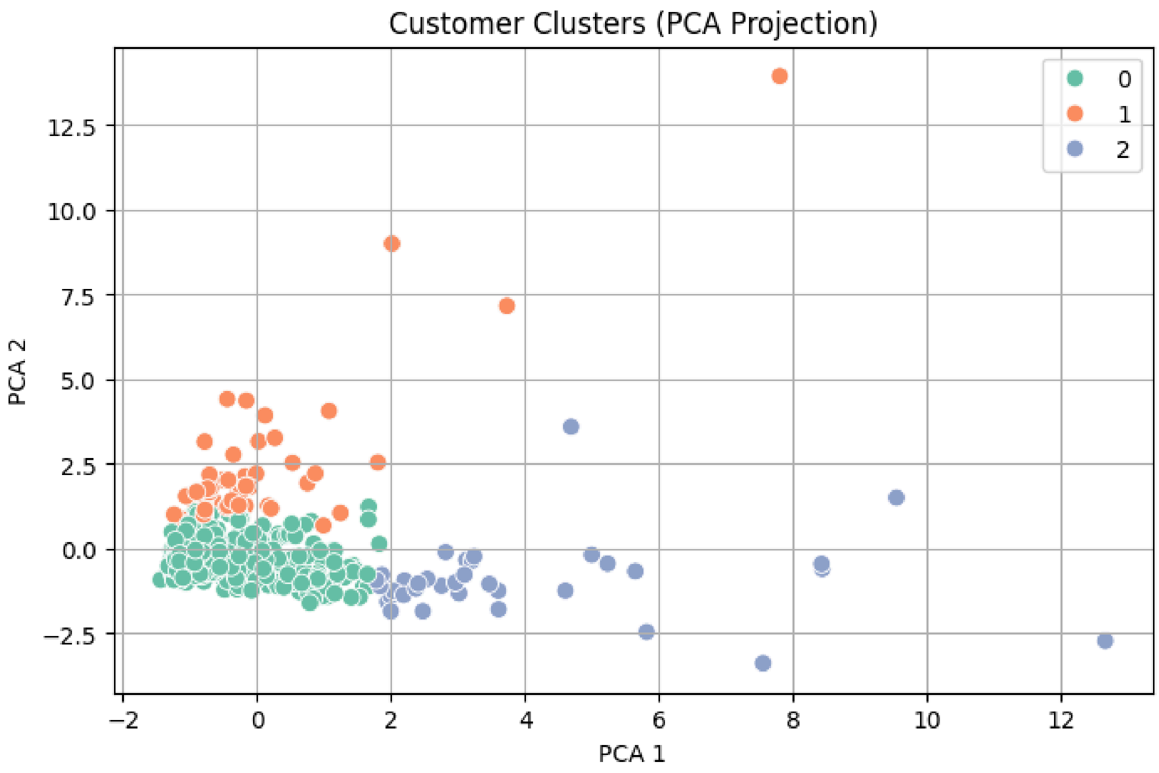
```
# Elbow method to find optimal K
inertia = []
K = range(1,11)

for k in K:
    model = KMeans(n_clusters=k, random_state = 42)
    model.fit(X_scaled)
    inertia.append(model.inertia_)
# plot the elbow curve
plt.figure(figsize=(8,5))
plt.plot(K, inertia, marker='o')
plt.title("Elbow Method-Optimal K")
plt.xlabel("Number of Clusters")
plt.ylabel("Inertia")
plt.grid(True)
plt.show()
```



In [6]:
```
# Fit KMeans
kmeans = KMeans(n_clusters=3, random_state=42)
data['Cluster'] = kmeans.fit_predict(X_scaled)
```

In [7]:
```
# Reduce to 2D for plotting
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
# Add PCA components to dataframe
data['PCA1'] = X_pca[:,0]
data['PCA2'] = X_pca[:,1]
# Plot the clusters
plt.figure(figsize=(8,5))
sns.scatterplot(data=data, x='PCA1', y='PCA2', hue='Cluster', palette='Set2', s=
plt.title("Customer Clusters (PCA Projection)")
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.grid(True)
plt.legend()
plt.show()
```



In [8]:

```
# Evaluate using the silhouette score
score = silhouette_score(X_scaled, data['Cluster'])
print(f"Silhouette Score for K=3: {score:.2f}")
```

```
Silhouette Score for K=3: 0.46
```

In [ ]: