Customer Segmentation using K-Means Clustering for Mall Marketing Insights

By Luis Mireles

This project performs customer segmentation using K-Means clustering on a dataset of mall customers. The dataset includes features such as age, annual income, and spending score. An optimal number of clusters (k=5) was determined using the Elbow Method, which identified the point where the inertia starts to decrease at a slower rate. The resulting clusters provide valuable insights into the different customer segments, enabling targeted marketing strategies and better understanding of customer behavior patterns.

source data: https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python

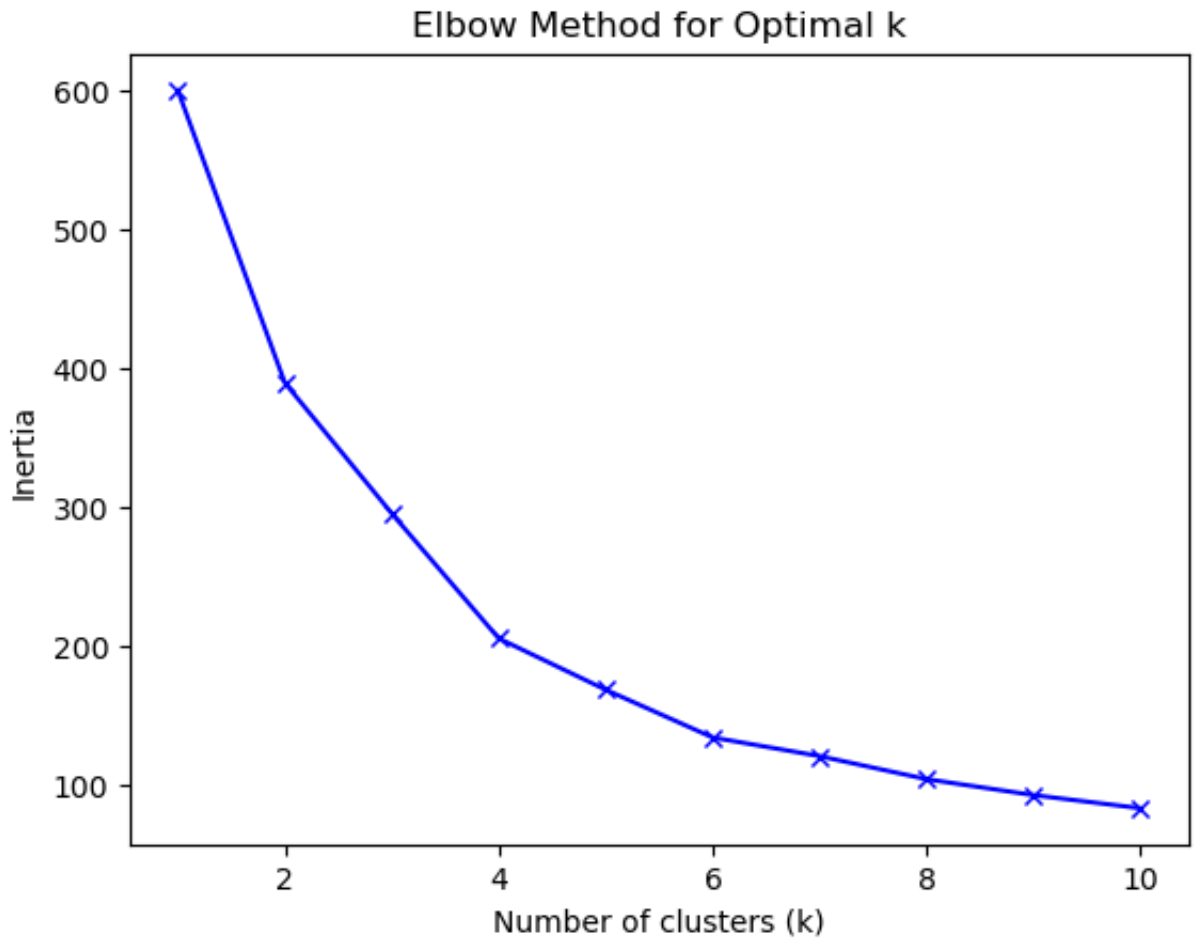
In [19]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

file_path = '/Users/fernando/Desktop/Code/Sources/Mall_Customers.csv'
data = pd.read_csv(file_path)

print('Columns check:\n' ,data.columns)
top_10_spending = data.sort_values(by='Spending Score (1-100)', ascending=False).head(10)
print('\n\n Top 10 Spending Score: \n\n', top_10_spending)
# print(data.head(10))
# print(data.info())

selected_features = ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']
data = data[selected_features].dropna()

scaler = StandardScaler()
```

```
data_scaled = scaler.fit_transform(data)

inertia = []
K = range(1, 11)
for k in K:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)

plt.plot(K, inertia, 'bx-')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal k')
plt.show()

# 5 clusters chosen as minimal slope from 4 to 5 on elbow method for optimal k
n_clusters = 5
kmeans = KMeans(n_clusters=n_clusters, random_state=42)
kmeans.fit(data_scaled)

data['Cluster'] = kmeans.labels_

# print(data.head())

cluster_summary = data.groupby('Cluster').mean()
print("\n\nCluster Summary: \n\n", cluster_summary)
```

```
Columns check:
 Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
        'Spending Score (1-100)'],
       dtype='object')
```

Top 10 Spending Score:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
11	12	Female	35	19	99
19	20	Female	35	23	98
145	146	Male	28	77	97
185	186	Male	30	99	97
127	128	Male	40	71	95
167	168	Female	33	86	95
7	8	Female	23	18	94
141	142	Male	32	75	93
163	164	Female	31	81	93
41	42	Male	24	38	92



Cluster Summary:

Cluster	Age	Annual Income (k$)	Spending Score (1-100)
0	55.638298	54.382979	48.851064
1	32.875000	86.100000	81.525000
2	25.185185	41.092593	62.240741
3	46.250000	26.750000	18.350000
4	39.871795	86.102564	19.358974

CLUSTER REPORT

Cluster 0: Customers in this segment are, on average, 55.6 years old, have an annual income of $54.4k, and a spending score of 48.9. This group represents older customers with moderate income and moderate spending habits.

Cluster 1: Customers in this segment are, on average, 32.9 years old, have an annual income of $86.1k, and a spending score of 81.5. This group represents younger customers with high income and high spending habits.

Cluster 2: Customers in this segment are, on average, 25.2 years old, have an annual income of $41.1k, and a spending score of 62.2. This group represents young customers with lower income but relatively high spending habits.

Cluster 3: Customers in this segment are, on average, 46.3 years old, have an annual income of $26.8k, and a spending score of 18.4. This group represents middle-aged customers with low income and low spending habits.

Cluster 4: Customers in this segment are, on average, 39.9 years old, have an annual income of $86.1k, and a spending score of 19.4. This group represents middle-aged customers with high income but low spending habits.

Comments:

Clusters 1 & 2 have the highest spending score at 81.5 and 62.2 this would be the focus for marketing to make the most out of marketing investment.

Side notes: If wanted to incorporate the gender it would have to be turned numerical since K-Means only works with numeric data. The get_dummies() function will convert the 'Gender' column to binary values (0 and 1)

((Encode the 'Gender' column data = pd.get_dummies(data, columns=['Gender'], drop_first=True) Select the features for clustering X = data[['Age', 'Annual Income (k$)', 'Spending Score (1-100)', 'Gender_Male']]))

Notes: Can use just theGender_Male column as it was created through one-hot encoding of the original 'Gender' column. One-hot encoding is a technique that converts a categorical variable into binary (0 or 1) columns, one for each category. In this case, since there are only two categories (Male and Female), only one binary column is needed to represent the information.