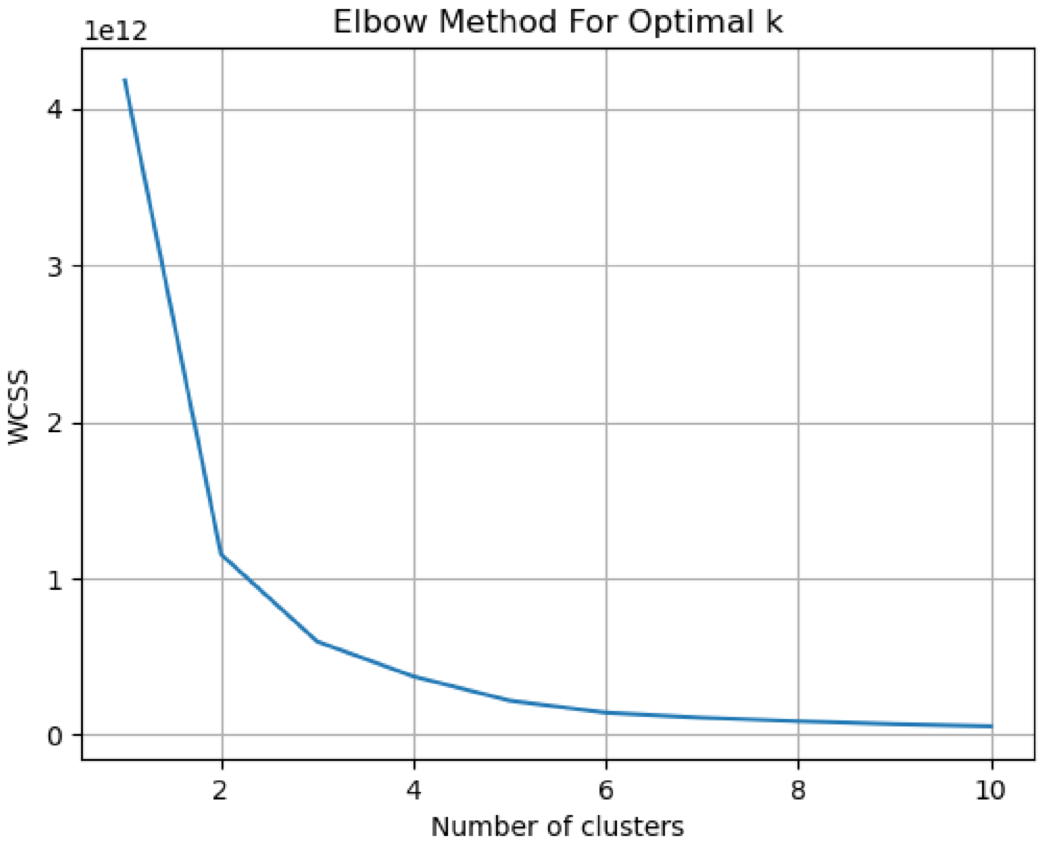
DataScienceProjects

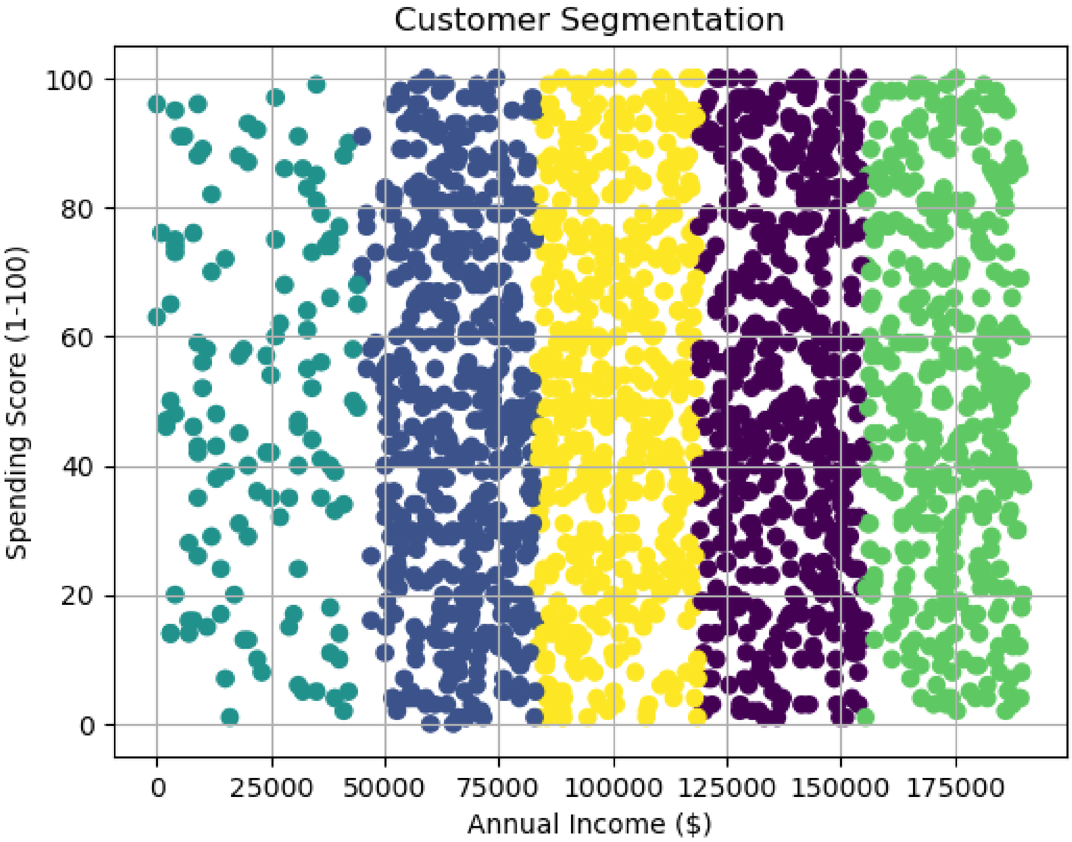
June 14, 2025

[14]:

```
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Load data
data = pd.read_csv("C:/Users/md.saifullah/OneDrive/Desktop/customer/Customers.
 ↪csv")

# Clean whitespaces in column names
data.columns = data.columns.str.strip()

# Print actual column names to identify correct ones
print("Available columns:", data.columns.tolist())

#  Manually check printed names and adjust accordingly.
# Example: You may find them as 'Annual Income (k$)' and 'Spending Score␣
 ↪(1-100)'
# Let's use a flexible approach here:

# Try to find close matches
income_col = [col for col in data.columns if 'income' in col.lower()][0]
score_col = [col for col in data.columns if 'spending' in col.lower()][0]

print("Using columns:", income_col, "and", score_col)

# Select features
features = data[[income_col, score_col]]

# Elbow method to find optimal clusters
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(features)
    wcss.append(kmeans.inertia_)

# Plot Elbow graph
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method For Optimal k')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()

# Apply KMeans with optimal_k = 5
kmeans = KMeans(n_clusters=5, random_state=42)
data['Cluster'] = kmeans.fit_predict(features)

# Visualize clusters
plt.scatter(data[income_col], data[score_col], c=data['Cluster'],␣
 ↪cmap='viridis')
plt.title('Customer Segmentation')
plt.xlabel(income_col)
plt.ylabel(score_col)
plt.grid(True)
plt.show()
```

```
Available columns: ['CustomerID', 'Gender', 'Age', 'Annual Income ($)',
'Spending Score (1-100)', 'Profession', 'Work Experience', 'Family Size']
Using columns: Annual Income ($) and Spending Score (1-100)
```




Customer Segmentation
Spending Score (1-100)
0
20
40
60
80
100
0
25000
50000
75000
100000
125000
150000
175000
Annual Income ($)

```
[ ]:
```