

Customer Segmentation Using Data Science

PHASE 4 : DEVELOPMENT PART-2

Introduction

In this project, we aim to perform customer segmentation using the K-Means clustering algorithm. Customer segmentation is a valuable strategy for businesses, allowing them to group customers with similar characteristics and behaviors to tailor marketing efforts, product offerings, and services.

Dataset

We are using the "Mall_Customers.csv" dataset, which contains information about customers in a shopping mall. The dataset includes the following columns:

- 1) CustomerID: A unique identifier for each customer.
- 2) Genre: Gender of the customer.
- 3) Age: Age of the customer.
- 4) Annual Income (k\$): The annual income of the customer.
- 5) Spending Score (1-100): A score assigned to the customer's spending behavior.

Let's dive into the steps of the project:

Step 1: Data Loading and Exploration

Python

```
import pandas as pd

# Load the dataset
data = pd.read_csv('Mall_Customer.csv')

# Display the first few rows of the dataset
print(data.head())
```

In this step, we load the dataset and display the first few rows to understand its structure.

Step 2: Feature Selection and Preprocessing

Python

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Select relevant features for clustering (Annual Income and Spending Score)
X = data[['Annual Income (k$)', 'Spending Score (1-100)']]

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

We select the "Annual Income" and "Spending Score" as the features for clustering and standardize them for better clustering results.

Step 3: Determining the Optimal Number of Clusters

Python

```
import matplotlib.pyplot as plt

wcss = [] # Within-Cluster Sum of Squares

# Determine the optimal number of clusters using the Elbow Method
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,
random_state=0)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Plot the Elbow Method
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
```

```
plt.show()
```

In this step, we use the Elbow Method to find the optimal number of clusters for K-Means.

Step 4: Clustering

Python

```
# Based on the Elbow Method, choose an appropriate number of clusters (e.g., 5)
num_clusters = 5

# Apply K-Means clustering
kmeans = KMeans(n_clusters=num_clusters, init='k-means++',
max_iter=300, n_init=10, random_state=0)
y_kmeans = kmeans.fit_predict(X_scaled)

# Add the cluster labels to the dataset
data['Cluster'] = y_kmeans
```

We select an appropriate number of clusters (in this case, 5) and apply K-Means clustering to the data.

Step 5: Visualization

python

```
# Visualize the clusters
plt.scatter(X_scaled[y_kmeans == 0, 0], X_scaled[y_kmeans == 0, 1], s=100,
c='red', label='Cluster 1')
# Continue for other clusters...

# Plot centroids
plt.scatter(kmeans.cluster_centers_[0, 0], kmeans.cluster_centers_[0, 1],
s=300, c='yellow', label='Centroids')
plt.title('Customer Segmentation')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
```

```
plt.legend()  
plt.show()
```

In this step, we visualize the clustered data and centroids.

Step 6: Cluster Analysis

Python

```
# Explore and analyze each cluster to understand customer segments  
for cluster_num in range(num_clusters):  
    cluster_data = data[data['Cluster'] == cluster_num]  
    print(f'Cluster {cluster_num} Statistics:')  
    print(cluster_data.describe())
```

We analyze each cluster to understand the characteristics of different customer segments.

Step 7: Save the Clustered Dataset

Python

```
# Save the clustered dataset  
data.to_csv('Preprocessed_Mall_customers.csv', index=False)
```

Finally, we save the clustered dataset for further analysis or marketing strategies.

This project allows businesses to gain insights into their customer base and tailor their strategies to better meet the needs of different customer segments. The resulting clustered dataset can be used for targeted marketing, product recommendations, and more.