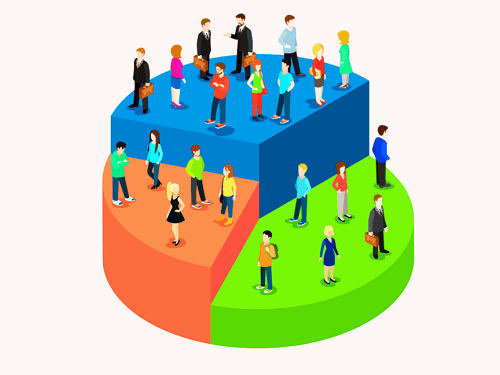
IBM-APPLIED DATA SCIENCE

GROUP-3

Phase 4: Development Part 2

PROJECT: Customer Segmentation Using Data Science



**INTRODUCTION:**

In this part of the project, we will continue building the customer segmentation model by focusing on feature engineering, applying clustering algorithms, visualizing the results, and interpreting the clusters. We will also provide a demo for each step.

**Feature Engineering:**

Feature engineering is crucial for creating meaningful clusters. It involves selecting, transforming, or creating features that will be used for customer segmentation. Let us demonstrate this step with an example:

**Feature Engineering Python Code:**

import pandas as pd

# **Load your dataset**

data = pd.read\_csv('customer\_data.csv')

# **Feature engineering** - **Example**: Creating a new feature 'Total Purchase'

data['Total Purchase'] = data['Order Quantity'] \* data['Price']

# **Select relevant features**

selected\_features = data[['Total Purchase', 'Age', 'Frequency']]

# **Standardize features if necessary**

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaled\_features = scaler.fit\_transform(selected\_features)

**# Continue with clustering algorithms using the scaled features**

**Applying Clustering Algorithms**:

Clustering algorithms will group customers with similar characteristics together. Common algorithms include K-Means, Hierarchical Clustering, and DBSCAN. Here's a demonstration using K-Means:

**Applying K-Means Clustering Python Code:**

from sklearn.cluster import KMeans

# **Determine the optimal number of clusters using the Elbow method**

import matplotlib.pyplot as plt

wcss = []

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(scaled\_features)

wcss.append(kmeans.inertia\_)

plt.plot(range(1, 11), wcss)

plt.title('Elbow Method')

plt.xlabel('Number of clusters')

plt.ylabel('WCSS')

plt.show()

# **Choose the optimal number of clusters (e.g., 3) and fit the K-Means model**

kmeans = KMeans(n\_clusters=3, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

data['Cluster'] = kmeans.fit\_predict(scaled\_features)

# **Continue with visualization and interpretation**

**Visualization:**

Visualizing the clusters helps you understand their characteristics. You can use various techniques like scatter plots, PCA, or t-SNE for dimensionality reduction. Let's create a simple scatter plot for our three clusters:

**Visualization Python Code:**

import seaborn as sns

**# Scatter plot**

sns.scatterplot(data=data, x='Total Purchase', y='Age', hue='Cluster')

plt.show()

**# Continue with interpretation**

**Interpretation:**

Interpreting the clusters is essential to make informed business decisions. You should analyze the cluster characteristics and identify key insights. For example:

**Interpretation Python Code:**

**# Calculate cluster statistics**

cluster\_stats = data.groupby('Cluster').agg({'Total Purchase': 'mean', 'Age': 'mean', 'Frequency': 'mean', 'Cluster': 'count'})

cluster\_stats.columns = ['Avg Total Purchase', 'Avg Age', 'Avg Frequency', 'Count']

# **Describe each cluster**

print(cluster\_stats)

**# Make business recommendations based on the cluster characteristics**

**# For instance, if Cluster 2 has high average purchase and low average age, target marketing efforts towards younger, high-value customers.**

Conclusion:

This completes Phase 4: Development Part 2 of your customer segmentation project. Remember that customer segmentation is an iterative process, and you can fine-tune your model and visualization based on your specific business needs and dataset.