**APPLIED DATA SCIENCE**

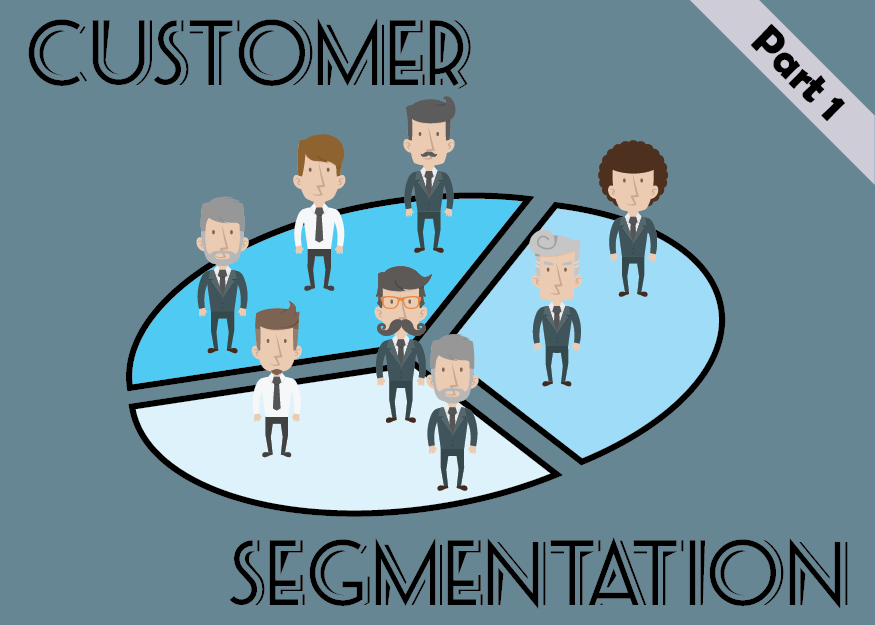
**IBM NAAN MUTHALVAN PHASE 4**

**TEAM MEMBERS:**

* **PRIYADHARSHINI.E**
* **KEERTHANA.S**
* **GAYATHRI.B**
* **DHARSNI RITIKA .KG**

**PROJECT TITLE:**

**CUSTOMER SEGMENTATION USING DATA SCIENCE.**

****

**DATASETLINK:** [**https://www.kaggle.com/datasets/akram24/mall-customers**](https://www.kaggle.com/datasets/akram24/mall-customers)

**Development Part :**

In this part you will continue building your project.

To building the customer segmentation model by:

* Feature engineering
* Applying clustering algorithms
* Visualization
* Interpretation.

**1.FEATURE ENGINEERING:**

1**. Demographic Features**: Include age, gender, location, income, education level, etc. These provide basic information about the customer.

**2. Behavioral Features**: Analyze customer behavior such as purchase history, frequency of purchases, products viewed, time spent on the website/app, etc.

3. **RFM Analysis**: Recency, Frequency, and Monetary analysis assesses how recently a customer made a purchase, how often they make purchases, and how much they spend.

4. **Social Media Activity**: If applicable, consider features like social media interactions, likes, shares, and comments, indicating customer engagement.

5. **Customer Interactions**: Track interactions with customer service, feedback, complaints, or any other communication, which can indicate customer satisfaction and loyalty.

**6. Seasonal Behavior:** Recognize patterns in customer behavior during specific seasons, holidays, or events.

**7. Customer Lifecycle Stage** : Categorize customers based on their stage in the buying cycle (e.g., awareness, consideration, decision, retention).

**8. RFID/NFC Data**:If applicable (for physical stores), utilize RFID/NFC data to understand customer movement within the store and product interaction.

**9. Website/App Usage Data:** Analyze how customers navigate through your website/app, which pages they visit, and how much time they spend on each page.

**10. Sentiment Analysis**: Utilize natural language processing techniques to analyze customer reviews, comments, and feedback to gauge sentiment.

**11. Customer Loyalty Programs**: Include data related to loyalty program participation, points earned, and redeemed.

**12. Geospatial Data**: Utilize location data to understand regional preferences and target specific locations with tailored marketing strategies.

**PROGRAM:**

# Splitting the data into features and target variable

X = data.drop('target\_column', axis=1) # Features

y = data['target\_column'] # Target variable

# Splitting the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Now you can use X\_train, X\_test, y\_train, y\_test for your machine learning model

Training

**2.APPLYING CLUSTERING ALGORITHAM**.

**1.Data Collection**: Gather relevant customer data such as demographics, purchase history, website interactions, and any other relevant features that can help in segmenting customers effectivelty.

**2. Data Preprocessing**: Cleanse the data by handling missing values, outliers, and inconsistencies. Normalize or standardize the data if the variables are on different scales. Preprocessing ensures that the data is ready for analysis.

**3. Feature Selection**: Choose the features that are most relevant to customer segmentation. Too many irrelevant or redundant features can negatively impact the clustering results.

**4. Choosing a Clustering Algorithm**: Select an appropriate clustering algorithm based on the nature of your data and the problem at hand. Common clustering algorithms include K-Means, Hierarchical Clustering, DBSCAN, and Gaussian Mixture Models.

**5. Feature Transformation (if necessary):** For some algorithms, especially distance-based methods like K-Means, it might be beneficial to perform feature transformation techniques such as Principal Component Analysis (PCA) to reduce dimensionality and improve clustering performance.

**6. Choosing the Right Number of Clusters:** If you're using algorithms like K-Means, determining the optimal number of clusters (k) is crucial. Techniques like the Elbow Method, Silhouette Score, or Gap Statistics can help you find the optimal k value.

**7. Model Training**: Apply the chosen clustering algorithm to the preprocessed data. Train the model to group similar customers together based on the selected features.

**8. Interpretation and Evaluation**: Analyze the clusters formed to understand the characteristics of each segment. Evaluate the results using metrics like Silhouette Score or Inertia (for K-Means) to assess the quality of the clustering.

**9. Implementation**: Implement marketing strategies or business decisions tailored to each customer segment. For example, different marketing campaigns or product recommendations can be designed for each cluster.

**10. Monitoring and Iteration**: Continuously monitor the effectiveness of the segmentation strategy. As new data becomes available, re-run the clustering algorithm to adapt to changing customer behaviors and preferences.

**PROGRAM:**

# Importing necessary libraries

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score

# Load the customer data

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

# Data Preprocessing

# Select relevant features for clustering (assuming columns 'feature1' and 'feature2')

features = data[['feature1', 'feature2']]

# Standardize the features (mean=0 and variance=1)

scaler = StandardScaler()

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

# Choosing the number of clusters using the Elbow Method

inertia = []

for k in range(1, 11):

    kmeans = KMeans(n\_clusters=k, random\_state=42)

    kmeans.fit(scaled\_features)

    inertia.append(kmeans.inertia\_)

# Plotting the Elbow Method to choose the optimal number of clusters

import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))

plt.plot(range(1, 11), inertia, marker='o')

plt.xlabel('Number of Clusters')

plt.ylabel('Inertia')

plt.title('Elbow Method')

plt.show()

# Based on the Elbow Method, choose the optimal number of clusters

optimal\_clusters = 3  # Example: assuming 3 clusters based on the elbow in the plot

# Applying K-Means Clustering

kmeans = KMeans(n\_clusters=optimal\_clusters, random\_state=42)

clusters = kmeans.fit\_predict(scaled\_features)

# Evaluating the clustering using Silhouette Score

silhouette\_avg = silhouette\_score(scaled\_features, clusters)

print(f'Silhouette Score: {silhouette\_avg}')

# Adding the cluster labels to the original dataset

data['Cluster'] = clusters

# Analyzing the clusters

for cluster\_num in range(optimal\_clusters):

    cluster\_data = data[data['Cluster'] == cluster\_num]

    print(f'\nCluster {cluster\_num}:')

    print(cluster\_data.describe())

**3.VISUALIZATION:**

**1. Scatter Plots**: Plot clusters using a scatter plot with different colors or markers for each cluster. If you have two prominent features (e.g., 'feature1' and 'feature2'), a scatter plot can provide a clear visual separation of clusters.

import matplotlib.pyplot as plt

# Assuming 'feature1' and 'feature2' are the features used for clustering

plt.figure(figsize=(8, 6))

plt.scatter(data[data['Cluster'] == 0]['feature1'], data[data['Cluster'] == 0]['feature2'], c='red', label='Cluster 0')

plt.scatter(data[data['Cluster'] == 1]['feature1'], data[data['Cluster'] == 1]['feature2'], c='blue', label='Cluster 1')

plt.scatter(data[data['Cluster'] == 2]['feature1'], data[data['Cluster'] == 2]['feature2'], c='green', label='Cluster 2')

plt.xlabel('Feature 1')

plt.ylabel('Feature 2')

plt.title('Customer Segmentation')

plt.legend()

plt.show()

**2. Heatmaps**: Visualize the cluster centers using a heatmap. This is useful when dealing with multiple features, allowing you to see the average values of features for each cluster.

import seaborn as sns

# Assuming 'feature1' and 'feature2' are the features used for clustering

cluster\_centers = pd.DataFrame(kmeans.cluster\_centers\_, columns=['feature1', 'feature2'])

plt.figure(figsize=(8, 6))

sns.heatmap(cluster\_centers, annot=True, cmap='YlGnBu')

plt.title('Cluster Centers')

plt.show()

**3. Radar Charts**: If you have more than two features, radar charts can be used to compare multiple features across clusters. Each axis represents a different feature.

# Assuming 'feature1', 'feature2', 'feature3', and 'feature4' are the features used for clustering

cluster\_means = data.groupby('Cluster').mean()[['feature1', 'feature2', 'feature3', 'feature4']]

categories = list(cluster\_means.columns)

N = len(categories)

values = cluster\_means.loc[cluster\_num].values.tolist()

angles = [n / float(N) \* 2 \* pi for n in range(N)]

values += values[:1]

angles += angles[:1]

plt.polar(angles, values, marker='o', label=f'Cluster {cluster\_num}')

plt.fill(angles, values, color='skyblue', alpha=0.4)

plt.xticks(angles[:-1], categories)

plt.title(f'Cluster {cluster\_num} Radar Chart')

plt.legend()

plt.show()

**4. INTERPRETATION:**

1. **Understand Cluster Characteristics**: Analyze the descriptive statistics (mean, median, standard deviation, etc.) of each cluster. Identify the key features that distinguish each cluster from others. This understanding helps in creating a profile for each segment.
2. **Visual Inspection**: Utilize visualizations (scatter plots, heatmaps, radar charts, etc.) to visually inspect the clusters. Look for clear boundaries and differences between clusters in the visualizations. Visual cues often provide intuitive insights into the nature of the segments.
3. **Business Context**: Relate the clusters back to your business goals. Consider why certain customer segments have formed.
4. **Compare Against Known Data**: If available, compare the segments against external data or existing customer profiles. This could include demographic information, historical data, or customer surveys.
5. **Feature Importance**: If machine learning algorithms were used, analyze feature importance. Identify which features had the most influence on the clustering. This helps in understanding what aspects of customer behavior or attributes are driving the segmentation.
6. **Customer Journey Mapping**: Map out the customer journey for each segment. Understand how customers in different segments interact with your products or services. Identify touchpoints, pain points, and opportunities for engagement for each segment.
7. **Validation and Refinement**: Validate the segmentation results using metrics like Silhouette Score or domain-specific metrics.If the segmentation doesn’t align with expectations, consider refining the features used for clustering or trying different algorithms.
8. **Actionable Insights**: Translate the segment characteristics into actionable strategies. For example, if one segment consists of price-sensitive customers, consider offering discounts or loyalty programs. Tailor marketing campaigns, product recommendations, and customer service approaches to suit the preferences of each segment.

1. **Long-term Analysis**: Continuously monitor customer behavior within segments over time.Customer preferences and behavior may change, requiring periodic reevaluation of segments and strategies.
2. **Stakeholder Collaboration**: Collaborate with marketing, sales, and customer service teams. Their domain expertise and feedback can provide additional insights and enhance the effectiveness of segmentation strategies.