PROJECT:CUSTOMER SEGMENTATION USING DATA SCIENCE

PHASE-4

PROBLEM STATEMENT

The problem is to implement data science techniques to segment customers based on their behavior, preferences, and demographic attributes.

DATASET EXPLANATION

https://www.kaggle.com/datasets/akram24/mall-customers

This dataset we used for this project in phase 4. Below is explanation for the dataset

Customer segmentation is a crucial marketing and business strategy that involves dividing a customer base into distinct groups based on specific characteristics. Here's an explanation of some common variables often used in a customer segmentation dataset:

DEMOGRAPHIC VARIABLES:

Age: Segmenting customers by age can help target products and marketing messages to specificage groups. For example, products for children, teenagers, adults, or seniors may differ significantly.

Gender: Gender-based segmentation can be useful for businesses offering gender-specific products.

Income: Income-based segmentation can help target products and services to customers with different spending power.

Customer id: In customer segmentation, it's common to include a unique customer identifier, often referred to as "Customer ID" or "Customer Number," in the dataset.

Spending score: In customer segmentation, the "spending score" is often an essential feature used for clustering customers based on their spending behaviour

1.FEATURE ENGINEERING

When it comes to customer segmentation through data science, feature engineering is a critical initial step. This process involves carefully selecting and crafting features that provide valuable insights into customer behavior and characteristics.

CODE

```
import pandas as pd

import numpy as np

data = pd.read_csv("D:\IBM project\Mall_Customers.csv")

data['Gender'] = data['Genre'].map({'Male': 0, 'Female': 1})

print('Gender\n',data['Genre'])

data['Income_Spending'] = data['Annual Income (k$)'] * data['Spending Score (1-100)']

print('Income spending\n',data['Income_Spending'])

age_bins = [0, 25, 35, 45, 55, 100]

age_labels = ['18-25', '26-35', '36-45', '46-55', '56+']

data['Age_Group'] = pd.cut(data['Age'], bins=age_bins, labels=age_labels)

print('Age group\n',data['Age_Group'])
```

OUTPUT

```
Gender
 0        Male
1        Male
2      Female
3      Female
4      Female
        ...
195    Female
196    Female
197      Male
198      Male
199      Male
Name: Genre, Length: 200, dtype: object
Income spending
 0        585
1       1215
2         96
3       1232
4        680
       ...
195     9480
196     3528
197     9324
198     2466
199    11371
Name: Income_Spending, Length: 200, dtype: int64
Age group
 0      18-25
1      18-25
2      18-25
3      18-25
4      26-35
       ...
195    26-35
196    36-45
197    26-35
198    26-35
199    26-35
Name: Age_Group, Length: 200, dtype: category
Categories (5, object): ['18-25' < '26-35' < '36-45' < '46-55' < '56+']
```

2. APPLYING CLUSTERING ALGORITHMS

The next phase involves the application of clustering algorithms.

These algorithms group customers into distinct segments based on the features we've engineered.

Popular clustering algorithms for customer segmentation include K-Means, hierarchical clustering, and DBSCAN.

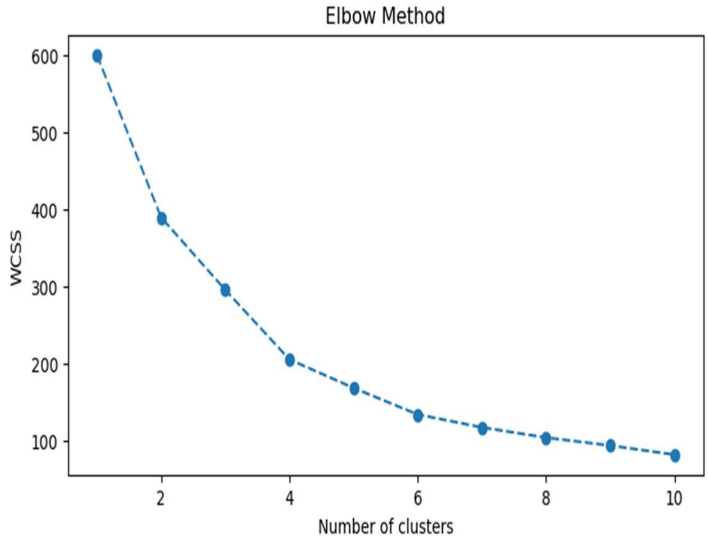
Each cluster generated represents a segment of customers who share common characteristics or behaviors.

CODE:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler
```

```
customer_data = pd.read_csv("D:\IBM project\Mall_Customers.csv")
features = customer_data[['Age', 'Annual Income (k$)','Spending Score (1-100)']]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)
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.figure(figsize=(8, 4))
plt.plot(range(1, 11), wcss, marker='o', linestyle='--')
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
optimal_clusters = 3  # Adjust as needed
kmeans = KMeans(n_clusters=optimal_clusters, init='k-means++', max_iter=300, n_init=10, random_state=0)
customer_data['Cluster'] = kmeans.fit_predict(scaled_features)
plt.figure(figsize=(8, 6))
for cluster in range(optimal_clusters):
    plt.scatter(reduced_features[customer_data['Cluster'] == cluster][:, 0],
                reduced_features[customer_data['Cluster'] == cluster][:, 1],
                label=f'Cluster {cluster}')
```

OUTPUTUT



3. VISUALIZATION

To gain a deeper understanding of the customer segments, visualization is indispensable.

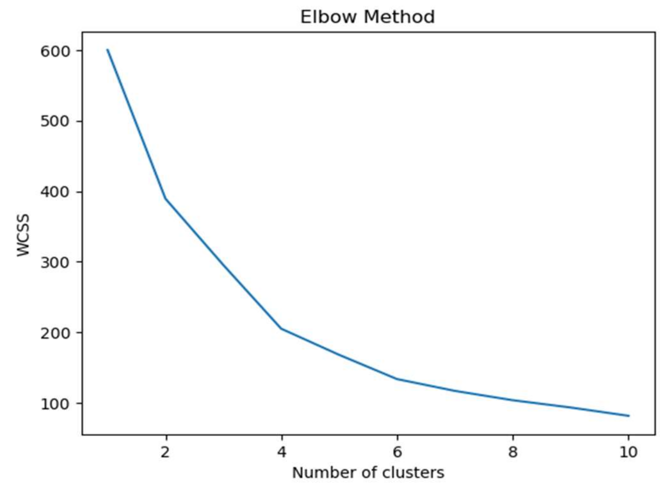
Visual representations help us comprehend and convey the results effectively.

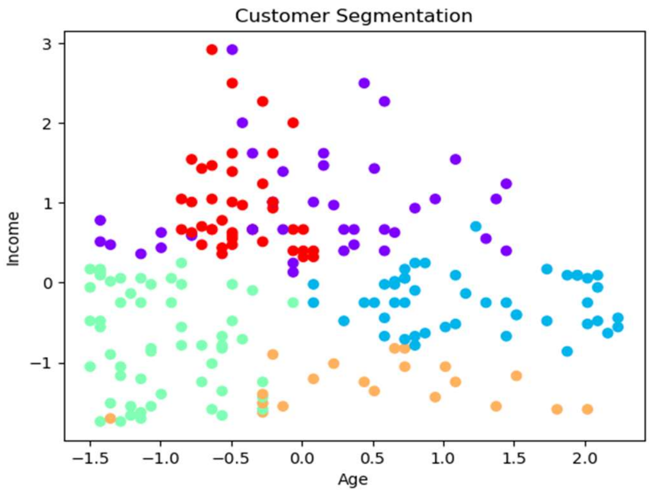
Visualization techniques may include scatter plots to visualize how customers cluster, heatmaps to show the similarity between customers in different clusters, bar charts to display the sizes of each segment, and dimensionality reduction plots like T-SNE or PCA, as well as distribution plots like box plots or histograms to understand feature distributions within clusters.

CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
data = pd.read_csv("D:\IBM project\Mall_Customers.csv")
X = data[['Age', 'Annual Income (k$)','Spending Score (1-100)']]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
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(X_scaled)
   wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')  # Within-cluster sum of squares
plt.show()
k = 5
kmeans = KMeans(n_clusters=k, init='k-means++', max_iter=300, n_init=10, random_state=0)
kmeans.fit(X_scaled)
data['Cluster'] = kmeans.labels_
plt.scatter(X_scaled[:, 0], X_scaled[:, 1], c=kmeans.labels_, cmap='rainbow')
plt.xlabel('Age')
plt.ylabel('Income')
plt.title('Customer Segmentation')
plt.show()
```

OUTPUT





4. INTERPRETATION:

The final step is interpretation, where we derive meaningful insights from the clusters created. This involves:

Assigning labels or names to each cluster based on common customer characteristics.

Distinguishing the unique traits of each segment, such as high spenders, infrequent shoppers, tech-savvy individuals, or price-sensitive customers.

Leveraging domain knowledge to explain the patterns uncovered.

Evaluating the quality of clustering results by using relevant metrics and aligning these insights with the business objectives.

Already interpretation is done in the previous code