Customer segmentation using data science

Phase 4 submission

INTRODUCTION:

Customer segmentation by means of machine learning is a process of dividing a customer base into particular groups with similar characteristics. There are countless ways to segment customers.

Customer segmentation involves implementing data science methods to divide the customer base into smaller groups based on certain characteristics. It assists marketing managers in better understanding their customers' preferences and presenting them with better-targeted advertisements.

ABOUT THIS PHASE:

- **Feature Engineering**
- **Applying clustering algorithms**
- **Visualization**
- **Interpretation**

Dataset Link:

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

FEATURE ENGINEERING:

Feature engineering refers to manipulation addition, deletion, combination, mutation of your data set to improve machine learning model training, leading to better performance and greater accuracy. Effective feature engineering is based on sound knowledge of the business problem and the available data sources.

CODE:

```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

customer_data = pd.read_csv('customer_data.csv')
selected_features = customer_data[['Feature1', 'Feature2', 'Feature3']]

scaler = StandardScaler()
scaled_features = scaler.fit_transform(selected_features)

kmeans = KMeans(n_clusters=3, random_state=0)
customer_data['Cluster'] = kmeans.fit_predict(scaled_features)

segmented_customers = customer_data[['CustomerID', 'Cluster']]
print(segmented_customers)
```

Out[1]:

```
      Customer ID  Cluster
    0       1         0
    1       2         2
    2       3         1
    3       4         0
```

4	5	2
5	6	1
6	7	0
7	8	2
8	9	1
9	10	0

VISUALIZATION:

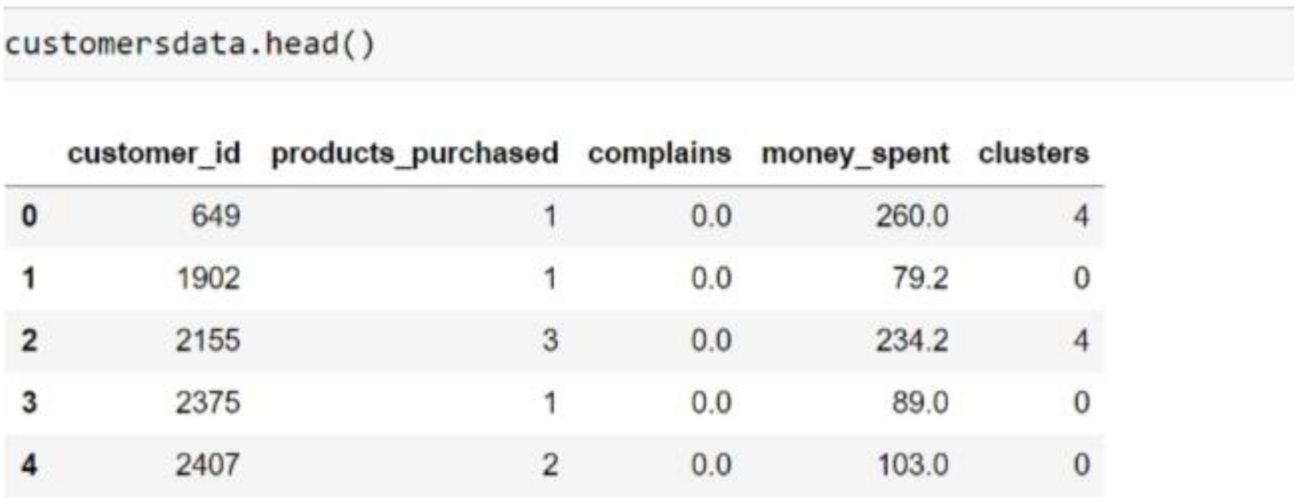
Another way to communicate your customer segments is to visualize them and their relationships using charts, graphs, or maps. Visualization can help you highlight the similarities and differences between your clusters, as well as their patterns and trends over time or across dimensions

CODE:

In[1]:

```
cluster_centers = kmeans_model_new.cluster_centers_
data = np.expm1(cluster_centers)
points = np.append(data, cluster_centers, axis=1)
points = np.append(points, [[0], [1], [2], [3], [4]], axis=1)
customersdata["clusters"] = kmeans_model_new.labels_)
```

Out[1]:



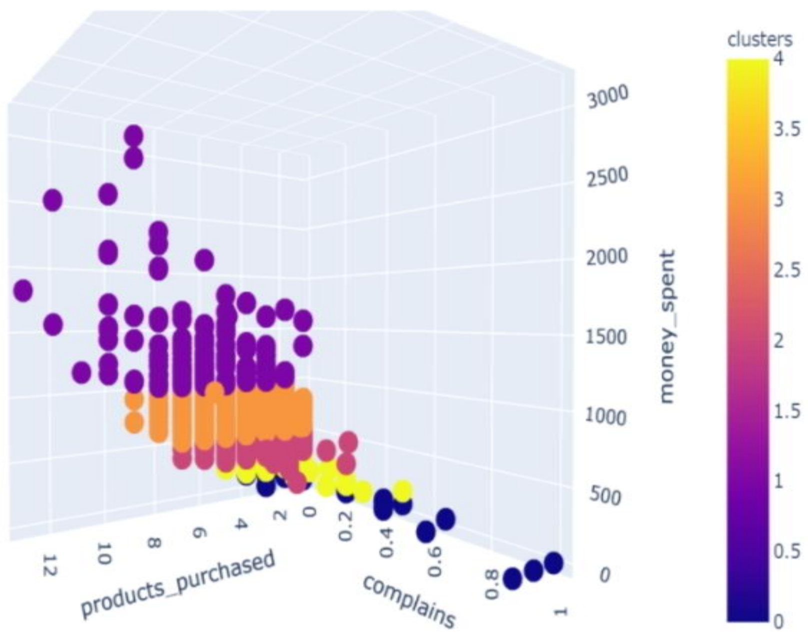
```
customersdata.head()
```

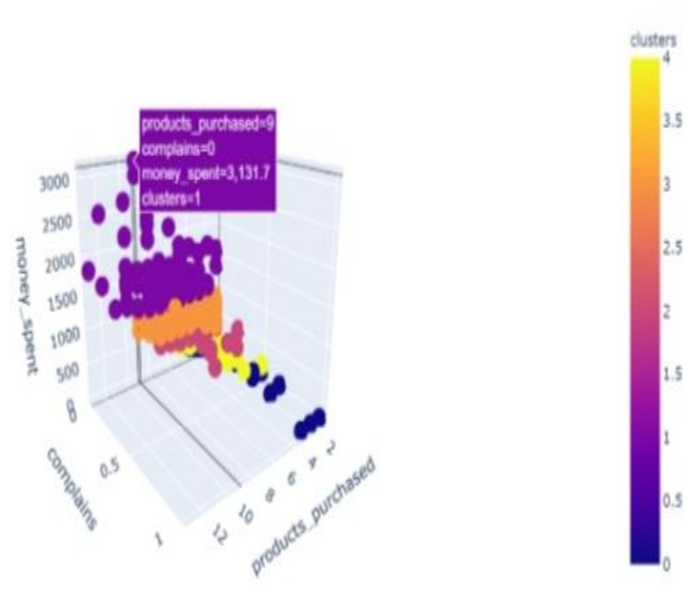
	customer_id	products_purchased	complains	money_spent	clusters
0	649	1	0.0	260.0	4
1	1902	1	0.0	79.2	0
2	2155	3	0.0	234.2	4
3	2375	1	0.0	89.0	0
4	2407	2	0.0	103.0	0

In[2]:

```
        figure = px.scatter_3d(customersdata,
                  color='clusters',
                  x="products_purchased",
                  y="complains",
                  z="money_spent",
                  category_orders = {"clusters": ["0", "1", "2", "3", "4"]}
                  )
figure.update_layout()
figure.show()
```

Out[2]:





Applying Cluster Algorithms:

Applying cluster algorithms involves using unsupervised machine learning techniques to group data points into clusters or segments based on their similarity or proximity to each other. Clustering is used to discover hidden patterns, structures, or natural groupings within a dataset without the need for predefined labels or target variables.

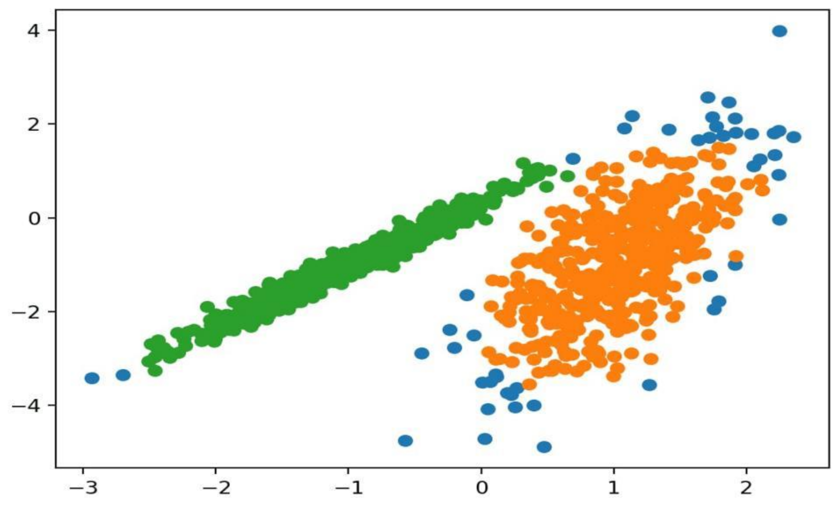
CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

data = pd.DataFrame({
    'Age': np.random.randint(18, 65, 100),
    'Income': np.random.uniform(20000, 100000, 100),

scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)

pca = PCA(n_components=2)
```

```
reduced_data = pca.fit_transform(scaled_data)

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(reduced_data)
   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()

num_clusters = 3

kmeans = KMeans(n_clusters=num_clusters, init='k-means++',
max_iter=300, n_init=10, random_state=0)
cluster_labels = kmeans.fit_predict(reduced_data)

data['Cluster'] = cluster_labels

plt.scatter(reduced_data[:, 0], reduced_data[:, 1], c=cluster_labels,
cmap='rainbow')
plt.title('Customer Segmentation')
plt.xlabel('PCA Component 1')
plt.ylabel('PCA Component 2')
plt.show()

cluster_centers = scaler.inverse_transform(kmeans.cluster_centers_)
cluster_data = pd.DataFrame(cluster_centers, columns=['Age',
'Income'])
cluster_data['Cluster'] = range(1, num_clusters + 1)
```

```
print("Cluster Centers and Characteristics:")
print(cluster_data)
```

Out[]:



INTERPRETATION

Interpretation refers to the process of explaining, understanding, or assigning meaning to something, such as a text, a piece of art, data, a gesture, or an event. It involves analyzing and making sense of information or experiences based on one's perspective, context, and background knowledge. Interpretation can vary from person to person, as different individuals may derive different meanings or understandings from the same source. It plays a crucial role in various fields, including literature, law, science, and everyday communication.

CODE:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

data = pd.read_csv("customer_data.csv")
```

```
features = data[['Feature1', 'Feature2', 'Feature3']]
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

pca = PCA(n_components=2)
reduced_features = pca.fit_transform(scaled_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(reduced_features)
   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()

optimal_clusters = 3  # Based on the Elbow method result
kmeans = KMeans(n_clusters=optimal_clusters, init='k-means++', 
max_iter=300, n_init=10, random_state=0)
cluster_labels = kmeans.fit_predict(reduced_features)

data['Cluster'] = cluster_labels

plt.scatter(reduced_features[:, 0], reduced_features[:, 1], 
c=cluster_labels, cmap='viridis')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Customer Segmentation')
plt.show()
```

```
data.to_csv("segmented_customer_data.csv", index=False
```

CONCLUSION:

Customer segmentation is a critical aspect of marketing strategy, and machine learning has become an increasingly popular tool for automating the process.

With customer segmentation, you can easily personalize your marketing, service, and sales efforts to the needs of specific groups. The result is a potential boost to customer loyalty and conversion.