CUSTOMER SEGEMENTATION
PHASE 5

PROBLEM STATEMENT

The problem is to implement data science techniques to segment customers based on their behavior, preferences, and demographic attributes. The goal is to enable businesses to personalize marketing strategies and enhance customer satisfaction. This project involves data collection, data preprocessing, feature engineering, clustering algorithms, visualization, and interpretation of results.

Customer Segmentation is the process of dividing a company's customers into groups based on common characteristics so companies can market to each group effectively and appropriately.

INTRODUCTION

Customer segmentation is the process of grouping customers based on similar characteristics such as geography, demography, behavior, purchasing power, and more, with the aim of enhancing customer acquisition, retention, profitability, satisfaction, and resource allocation through marketing strategies.

Clustering is a method for customer segmentation that involves placing homogenous data points in a dataset, forming groups called clusters.

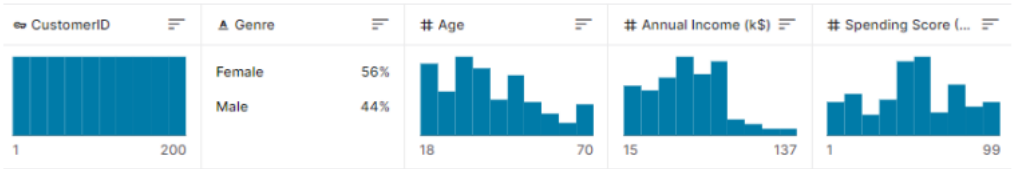
Segmentation allows marketers to better tailor their marketing efforts to various audience subsets. Those efforts can relate to both communications and product development. Specifically, segmentation helps a company to improve customer service.

Some customer data can be gathered from purchasing information—job title, geography, or products purchased, for example. Some of it might be gleaned from how the customer entered your system.

DATASET USED

Dataset Link: https://www.Kaggle.com/data

There exist 5*5 columns in the data set that is using in this project.



0035	Female	49	33	14
0036	Female	21	33	81
0037	Female	42	34	17
0038	Female	30	34	73
0039	Female	36	37	26
0040	Female	20	37	75
0041	Female	65	38	35
0042	Male	24	38	92
0043	Male	48	39	36
0044	Female	31	39	61
0045	Female	49	39	28
0046	Female	24	39	65
0047	Female	50	40	55
0048	Female	27	40	47
0049	Female	29	40	42
0050	Female	31	40	42

TOOLS & SOFTWARE USED

1. Programming language
2. Machine Learning Libraries
3. Integrated Development Environment
4. Data Visualization tools
5. Data Preprocessing tools
6. Data Collection & Storage
7. Notebooks & Documentation
8. External Data Source
9. Data Labelling tools
10. Version tools

PROCESS OVERVIEW

(1) Design thinking
(2) Design into innovation
(3) Build Load and Preprocessing of dataset
(4) Additional Evaluation

(1) DESIGN THINKING AND PRESENT IN FORMOF DOCUMENT

Empathize:

- Understand the annual income and spending score of the customers.
- Conduct interviews and surveys to gather insights on customers and what information is used for critical thinking.

Define:

- Clearly articulate the problem statement and identify the key goals and success criteria for the project and improve customer's trust in the evaluation process

Ideate:

- Creative solutions and data sources that can enhance the transparency of segmentation.
- Encourage interdisciplinary collaboration to generate a wide range of ideas, including the use of alternative data, new algorithms, or improved visualization techniques.

Prototype:

- Create prototype machine learning models based on the ideas generated during the ideation phase.
- Test and iterate on these prototypes to determine which approaches are most promising in terms of accuracy and usability.

Test & Train:

- There are many different machine learning algorithms that can be used for house price prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.

Implement:

- Implement transparency measures, such as model interpretability tools, to ensure customers understand how segments are generated.

Evaluate:

- Continuously monitor the performance of the machine learning model after implementation to ensure it remains accurate and relevant.
- Gather feedback and insights from customers to identify improvement.

Iterate:

- Apply an iterative approach to refine the machine learning model based on ongoing feedback and changing user needs.

(2) DESIGN INTO INNOVATION

Data Collection:

- The process involves gathering customer data, which includes information about their purchase history, demographics, and interaction patterns.

Data Preprocessing:

- The task involves preparing and cleaning data, handling missing values, and converting categorical features into numerical representations.

Feature Engineering:

- Data preparation and cleaning, handling missing values, and the transformation of categorical features into numerical representations are all part of the task.

Algorithms Used:

- Apply clustering algorithms like K-Means, DBSCAN, or hierarchical clustering to segment customers.

Visualization:

- Visualize the customer segments using techniques like scatter plots, bar charts, and heatmaps.

Interpretation:

- Analyze and interpret the characteristics of each customer segment to derive actionable insights for marketing strategies.

Features Selection:

- Use techniques like feature importance scores or recursive feature elimination to identify the most relevant features.

Monitoring:

- Regularly monitor the model's performance in the real world and update it as needed.

Ethical Consideration:

- Be mindful of potential biases in the data and model. Ensure fairness and transparency in your segmentation process.

Innovation:

- Consider innovative approaches to natural language processing for descriptions.

(3) BUILD LOADING AND PREPROCESSING THE DATASET

Dataset Loading:

- Import relevant libraries, such as pandas for data manipulation and numpy for numerical operations. Load the dataset into a pandas Data Frame for easy data handling. You can use pd.read_csv() for CSV files or other appropriate functions for different file formats.
- df=pd.read_csv('/kaggle/input/mall-customers/Mall_Customers.csv')

Data Exploration:

- Explore the dataset to understand its structure and contents. Check for the presence of missing values, outliers, and data types of each feature.

Data Cleaning:

- Handle missing values by either removing rows with missing data or imputing values based on the nature of the data.

Data Encoding:

- Convert categorical variables into numerical format using techniques like one-hot encoding.

Features Selection:

- Use techniques like feature importance scores or recursive feature elimination to identify the most relevant features.

Data Preprocessing:

- The raw data we downloaded is complex and in a format that cannot be easily ingested by customer segmentation models. We need to do some preliminary data preparation to make this data interpretable.

Model Building:

- We are going to create a K-Means clustering algorithm to perform customer segmentation. The goal of a K-Means clustering model is to segment all the data available into non-overlapping sub-groups that are distinct from each other.
- These includes various aspects like Cluster of customers, Kmeans clusters, dropping, counterplotting, scatterplotting, describing the values.

Missing Values:

- By using isnull() or notnull() we can identify the missing values in the data.

(4) ADDITIONAL EVALUATION

Evaluation Process:

- This includes various aspects starting from the preparation to deployment.

Data Preparation:

- This includes cleaning the data, removing outliers, and handling missing values.

Feature selection:

- This can be done using a variety of methods, such as correlation analysis, information gain, and recursive feature elimination.

Model Training:

- There are many different machine learning algorithms that can be used for house price prediction. Some popular choices include linear regression, random forests, and gradient boosting machines.
- Model Training is the process of teaching a machine learning model to do customer segmentation. It involves feeding the model historical data and features. The model then learns the relationships between these customers and spending scores.

Model Evaluation:

- This can be done by calculating the mean squared error (MSE) or the root mean squared error (RMSE) of the model's predictions on the held-out test set.
- Model Evaluation is the process of assessing the performance of a machine learning model on unseen data. This is important to ensure that the model will generalize to the new data.

Model Deployment:

- Once the model has been evaluated and found to be performing well, it can be deployed to production.

Model Comparison:

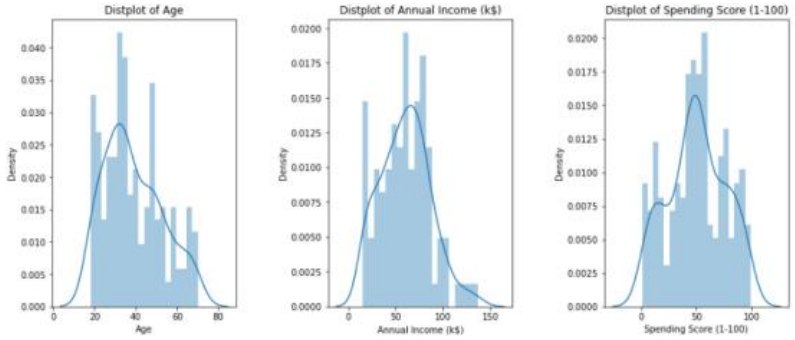
- Model Comparison is the tool enhanced to provide systematic representation on the relationship between annual income and spending score.

Model Analysis:

- Once evaluation of model gets completes, the analysis of model’s predictions can be started to identify any patterns or biases. This will help us to understand the strengths and weaknesses of the model and to improve it.

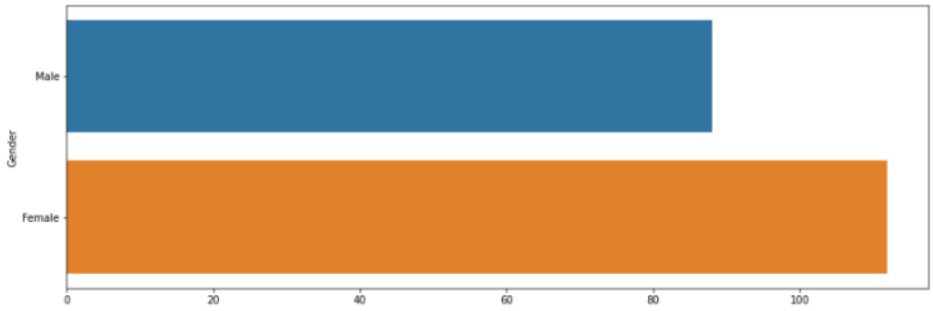
```
df.drop(['CustomerID'],axis=1,inplace=True)
```

```
plt.figure(1,figsize=(15,6))
n = 0
for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n +=1
    plt.subplot(1,3,n)
    plt.subplots_adjust(hspace=0.5,wspace=0.5)
    sns.distplot(df[x],bins=20)
    plt.title('Distplot of {}'.format(x))
plt.show()
```



Counter Plotting:

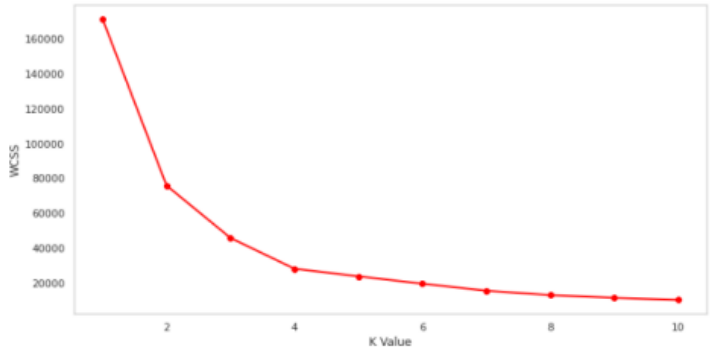
```
plt.figure(figsize=(15,5))
sns.countplot(y='Gender',data=df)
plt.show()
```



K-Means:

```
X1 = df.loc[:,["Age","Spending Score (1-100)"]].values

from sklearn.cluster import KMeans
wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = "k-means++")
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize =( 12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")
plt.xlabel("K Value")
plt.ylabel("WCSS")
plt.show()
```



K-Means Clustering:

```
kmeans = KMeans(n_clusters=4)

label = kmeans.fit_predict(X1)

print(label)
```

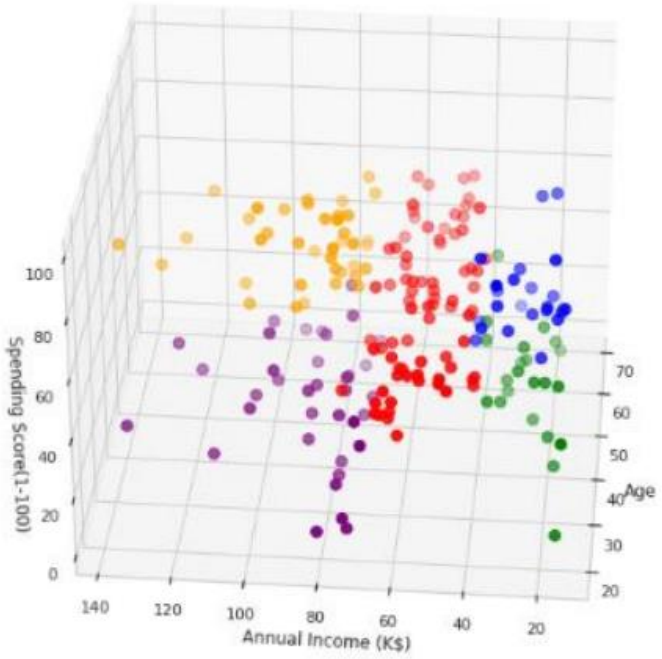
```
[1 2 0 2 1 2 0 2 0 2 0 2 0 2 0 2 1 1 0 2 1 2 0 2 0 2 0 1 0 2 0 2 0 2 0 2 0
 2 0 2 3 2 3 1 0 1 3 1 1 1 3 1 1 3 3 3 3 3 1 3 3 1 3 3 3 1 3 3 1 1 3 3 3 3
 3 1 3 1 1 3 3 1 3 3 1 3 3 1 1 3 3 1 3 1 1 1 3 1 3 1 1 3 3 1 3 1 3 3 3 3 3
 1 1 1 1 1 3 3 3 3 1 1 1 2 1 2 3 2 0 2 0 2 1 2 0 2 0 2 0 2 0 2 1 2 0 2 3 2
 0 2 0 2 0 2 0 2 0 2 0 2 3 2 0 2 0 2 0 2 0 1 0 2 0 2 0 2 0 2 0 2 0 2 0 2 1
 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2]
```

Cluster of Customers:

```
plt.scatter(X1[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
plt.title('Clusters of Customers')
plt.xlabel('Age')
plt.ylabel('Spending Score(1-100)')
plt.show
```

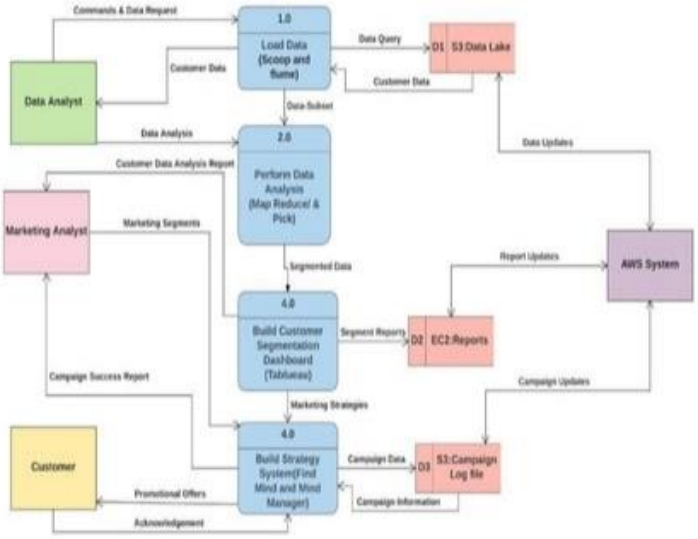
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

3-D Plot:

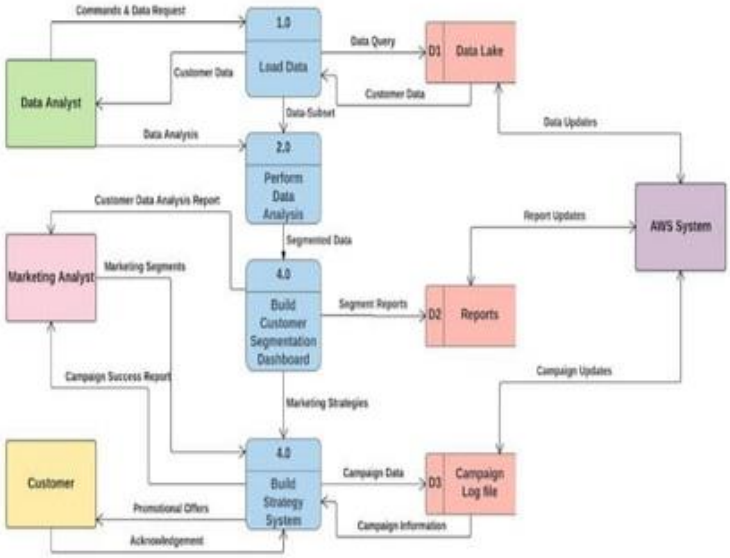


(5) DATA FLOW OF CUSTOMER MODEL

Physical Flow:



Logical Flow:



(6) CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df=pd.read_csv('/kaggle/input/mall-customers/Mall_Customers.csv')

df.rename(columns={'Genre':'Gender'},inplace=True)
df.head()
df.describe()

df.isnull().sum()

df.drop(['CustomerID'],axis=1,inplace=True)

plt.figure(1,figsize=(15,6))
n = 0
for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n +=1
    plt.subplot(1,3,n)
    plt.subplots_adjust(hspace=0.5,wspace=0.5)
    sns.distplot(df[x],bins=20)
    plt.title('Distplot of {}'.format(x))
plt.show()
plt.figure(figsize=(15,5))
sns.countplot(y='Gender',data=df)
plt.show()

plt.figure(1,figsize=(15,6))
n = 0
for cols in ['Age','Annual Income (k$)','Spending Score (1-100)']:
    n +=1
    plt.subplot(1,3,n)
    sns.set(style="whitegrid")
    plt.subplots_adjust(hspace=0.5,wspace=0.5)
    sns.violinplot(x = cols,y = 'Gender',data=df)
    plt.ylabel('Gender' if n== 1 else '')
    plt.title('Violin Plot')
plt.show()

age_18_25 = df.Age[(df.Age >=18) & (df.Age <= 25)]
age_26_35 = df.Age[(df.Age >=26) & (df.Age <= 35)]
```

```
age_36_45 = df.Age[(df.Age >=36) & (df.Age <= 45)]
age_46_55 = df.Age[(df.Age >=46) & (df.Age <= 55)]
age_55_above = df.Age[(df.Age >= 56)]

age_x =["18-25","26-35","36-45","46-55","55+"]
age_y =
[len(age_18_25.values),len(age_26_35.values),len(age_36_45),len(age_46_55),len(age_55_abov
e)]

plt.figure(figsize = (15,6))
sns.barplot(x=age_x, y=age_y,palette = "mako")
plt.title("Number of Customer and Ages")
plt.xlabel("Age")
plt.ylabel("Number of Customer")
plt.show()

sns.relplot(x="Annual Income (k$)",y = "Spending Score (1-100)",data=df)

ss_1_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Spending
Score (1-100)"] <= 20)]
ss_21_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) &
(df["Spending Score (1-100)"] <= 40)]
ss_41_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) &
(df["Spending Score (1-100)"] <= 60)]
ss_61_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) &
(df["Spending Score (1-100)"] <= 80)]
ss_81_100 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 81) &
(df["Spending Score (1-100)"] <= 100)]

ssx= ["1-20","21-40","41-60","61-80","81-100"]
ssy=[len(ss_1_20.values),len(ss_21_40.values),len(ss_41_60.values),len(ss_61_80.values),len(ss
_81_100.values)]

plt.figure(figsize=(15,6))
sns.barplot(x=ssx,y=ssy, palette="rocket")
plt.title("Spending Scores")
plt.xlabel("Score")
plt.ylabel("Number of Customer having the Score")
plt.show()
        ai_0_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual
Income (k$)"] <= 30)]
ai_31_60= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=31)& (df["Annual Income
(k$)"] <=60)]
ai_61_90= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=61)& (df["Annual Income
```

```
(k$)"] <=90)]
ai_61_90=df["Annual Income (k$)"][(df["Annual Income (k$)"] >=91)& (df["Annual Income
(k$)"] <=120)]
ai_121_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=121) & (df["Annual
Income (k$)"] <=150)]

aix = ["$ 0 - 30,000","$ 30,001 - 60,000","$ 60,001 - 90,000","$ 90,001 - 120,000","$ 120,001 -
150,000"]
aiy =
[len(ai_0_30.values),len(ai_31_60.values),len(ai_61_90.values),len(ai_61_90.values),len(ai_121
_150.values)]

plt.figure(figsize=(15,6))
sns.barplot(x=aix,y=aiy,palette="Spectral")
plt.title("Annual Incomes")
plt.xlabel("Income")
plt.ylabel("Numer of Customer")
plt.show()

X1 = df.loc[:,["Age","Spending Score (1-100)"]].values

from sklearn.cluster import KMeans
wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = "k-means++")
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize =( 12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")
plt.xlabel("K Value")
plt.ylabel("WCSS")
plt.show()

kmeans = KMeans(n_clusters=4)
label = kmeans.fit_predict(X1)
print(label)

print(kmeans.cluster_centers_)

plt.scatter(X1[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
plt.title('Clusters of Customers')
plt.xlabel('Age')
plt.ylabel('Spending Score(1-100)')
```

```
plt.show()

X2 = df.loc[:,["Annual Income (k$)","Spending Score (1-100)"]].values

from sklearn.cluster import KMeans
wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = "k-means++")
    kmeans.fit(X2)
    wcss.append(kmeans.inertia_)
plt.figure(figsize =( 12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")
plt.xlabel("K Value")
plt.ylabel("WCSS")
plt.show()

plt.scatter(X2[:,0],X1[:,1],c=kmeans.labels_,cmap='rainbow')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],color='black')
plt.title('Clusters of Customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score(1-100)')
plt.show()

X3 = df.iloc[:,1:]


wcss=[]
for k in range(1,11):
    kmeans = KMeans(n_clusters = k, init = "k-means++")
    kmeans.fit(X3)
    wcss.append(kmeans.inertia_)
plt.figure(figsize =( 12,6))
plt.grid()
plt.plot(range(1,11),wcss,linewidth=2,color="red",marker="8")
plt.xlabel("K Value")
plt.ylabel("WCSS")
plt.show()

cluster = kmeans.fit_predict(X3)
df["label"] = cluster


from mpl_toolkits.mplot3d import Axes3D
```

```
fig = plt.figure(figsize=(20,10))
ax = fig.add_subplot(111,projection = '3d')

ax.scatter(df.Age[df.label == 0 ],df["Annual Income (k$)"][df.label == 0],df["Spending Score (1-100)"][df.label == 0], c = 'blue',s=60)
ax.scatter(df.Age[df.label == 1 ],df["Annual Income (k$)"][df.label == 1],df["Spending Score (1-100)"][df.label == 1], c = 'red',s=60)
ax.scatter(df.Age[df.label == 2 ],df["Annual Income (k$)"][df.label == 2],df["Spending Score (1-100)"][df.label == 2], c = 'green',s=60)
ax.scatter(df.Age[df.label == 3 ],df["Annual Income (k$)"][df.label == 3],df["Spending Score (1-100)"][df.label == 3], c = 'orange',s=60)
ax.scatter(df.Age[df.label == 4 ],df["Annual Income (k$)"][df.label == 4],df["Spending Score (1-100)"][df.label == 4], c = 'purple',s=60)

ax.view_init(30,185)

plt.xlabel("Age")
plt.ylabel("Annual Income (K$)")
ax.set_zlabel('Spending Score(1-100)')
plt.show()
```

(7) CONCLUSION

The process of customer segmentation ensures that your brand is customer-centric and helps you serve them better. It boosts conversions, brings your marketing efforts to fruition, and also helps build everlasting customer relationships. The strategies discussed here will help you organize your segments, but after you have them in place, continue to monitor and make sure your product is still valuable to the groups. The key to successful customer segmentation is the constant research it entails to ensure your brand and product stay relevant and indispensable.

The customer segmentation project using machine learning has yielded valuable insights that can significantly benefit our business. Through the application of advanced algorithms and data analysis techniques, we have successfully divided our customer base into distinct segments based on various attributes such as behaviour, demographics, and preferences.