Customer segmentation using applied data science

Project Introduction:

The problem at hand is to implement data science techniques to segment customers based on their behavior, preferences, and demographic attributes. The primary objective is to enable businesses to personalize marketing strategies and enhance overall customer satisfaction. This project encompasses several key stages, including data collection, data preprocessing, feature engineering, clustering algorithms, visualization, and interpretation of results.

1.Data Collection

• Identify and collect relevant data sources: This may include customer transaction data, demographic information, website interactions, and any other relevant sources.

• Ensure data quality and integrity: Clean and preprocess data to remove missing values, outliers, and inconsistencies

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

A list of tools and software commonly used in the process:

1.Programming language:

Python is arguably the most popular language for data science due to its simplicity, extensive libraries, and a strong community of data scientists. Libraries like NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, and TensorFlow make useful for data manipulation, analysis, and machine learning.

2. An Integrated Development Environment (IDE):

Choose an IDE for coding and running applied data science experiments.Some popular options are Jupyter Notebook,Google Colab.

3.Machine Learning Libraries:

You will need various libraries ,including: Libraries like NumPy, Pandas, Matplotlib, Seaborn

4.Data Visualization Tools:

Tools like Matplotlib, Seaborn

5.Data Preprocessing Tools:

Libraries like Pandas help with data cleaning, manipulation and preprocessing.

```
import numpy as np
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(context="notebook", palette="Spectral", style = 'darkgrid' ,font_scale =
1.5, color_codes=True)
print(os.listdir("../input"))
```

```
['mall-customers']
```

In [2]:

```
# Importing the dataset
dataset = pd.read_csv('../input/mall-
customers/Mall_Customers.csv',index_col='CustomerID')
```

In [3]:

```
dataset.head()
```

Out[3]:

CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

In [4]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 200 entries, 1 to 200
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Genre                   200 non-null    object
 1   Age                     200 non-null    int64
 2   Annual Income (k$)      200 non-null    int64
 3   Spending Score (1-100)  200 non-null    int64
dtypes: int64(3), object(1)
memory usage: 7.8+ KB
```

In [5]:

```
dataset.describe()
```

Out[5]:

	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000
mean	38.850000	60.560000	50.200000
std	13.969007	26.264721	25.823522
min	18.000000	15.000000	1.000000
25%	28.750000	41.500000	34.750000
50%	36.000000	61.500000	50.000000
75%	49.000000	78.000000	73.000000
max	70.000000	137.000000	99.000000

In [6]:

```
dataset.isnull().sum()
```

Out[6]:

```
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [7]:

```
dataset.drop_duplicates(inplace=True)
```

2.Feature Engineering

• Create relevant features: Derive meaningful features from the raw data that can be used for segmentation. This might involve aggregating transaction history, calculating customer lifetime value, or encoding categorical variables.

```
import numpy as np import pandas as pd
```

In [19]:

```
import os for dirname, _, filenames in os.walk('/kaggle/input'):
for filename in filenames: print(os.path.join(dirname, filename))
```

In [20]:

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

In [21]:

```
df=pd.read_csv('/content/Mall_Customers.csv')
 df.rename(columns={'Genre':'Gender'},inplace=True)
df.head()
```

Out[21]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [22]:

```
df.describe()
```

Out[22]:

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

In [23]:

```
df.shape
```

Out[23]:

```
(200, 5)
```

In [24]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

In [25]:

```
df.isnull().sum()
```

Out[25]:

```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [26]:

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

In [27]:

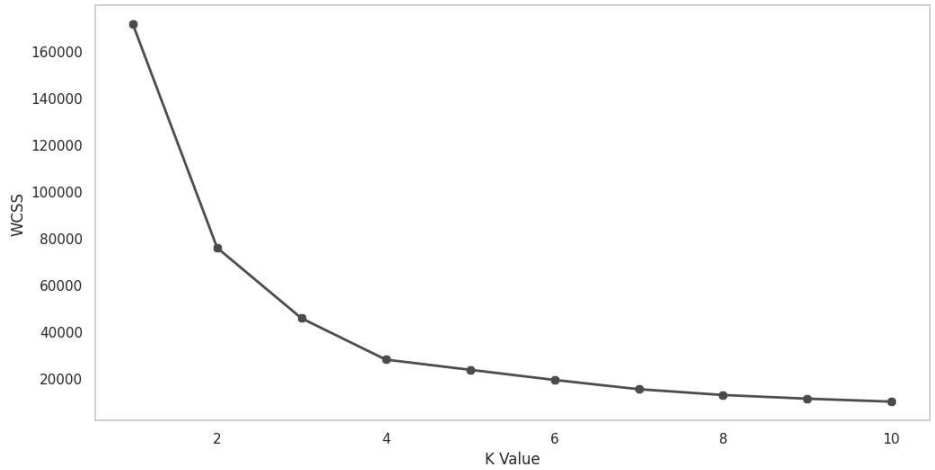
```
plt.figure(1,figsize=(15,6))
 n = 0 for x in ['Age','Annual Income (k$)','Spending Score (1-100)']:
n +=1 plt.subplot(1,3,n)
plt.subplots_adjust(hspace=0.5,wspace=0.5)
 sns.distplot(df[x],bins=20)
plt.title('Distplot of {}'.format(x))
plt.show()
```

3.Clustering Algorithms

• Choose appropriate clustering algorithms: Select algorithms suitable for customer segmentation, such as K-means, hierarchical clustering, or DBSCAN.
• Determine the number of clusters: Experiment with different cluster numbers to identify the optimal segmentation.
• Train the clustering model: Apply selected algorithms to the preprocessed data.

```
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)
```

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

```
print(kmeans.cluster_centers_)
```

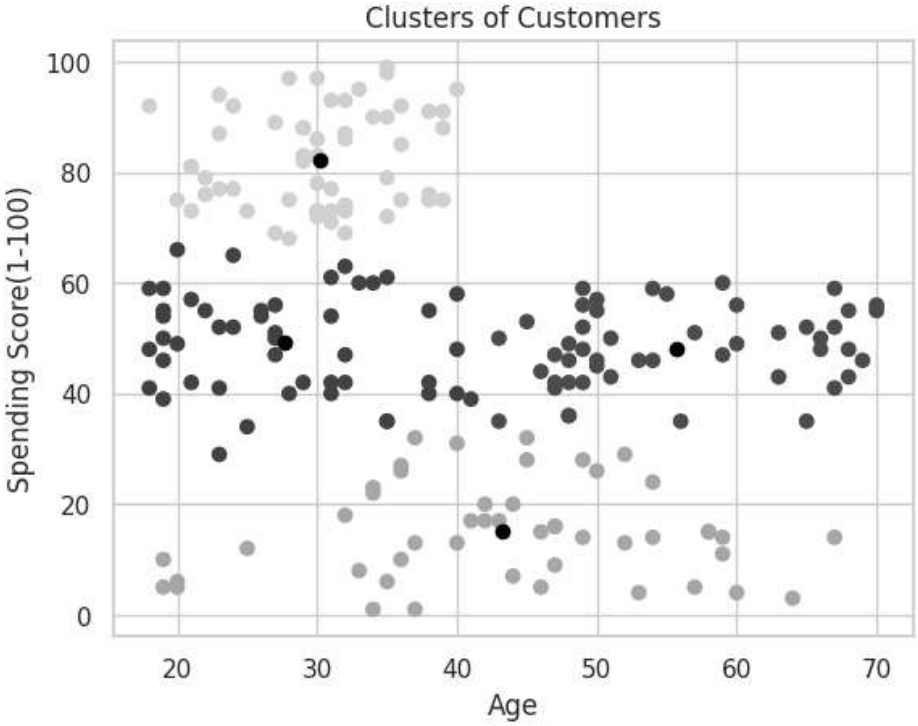
```
[[55.70833333 48.22916667]
 [30.1754386  82.35087719]
 [27.61702128 49.14893617]
 [43.29166667 15.02083333]]
```

In [37]:

```
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
```

Out[37]:

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

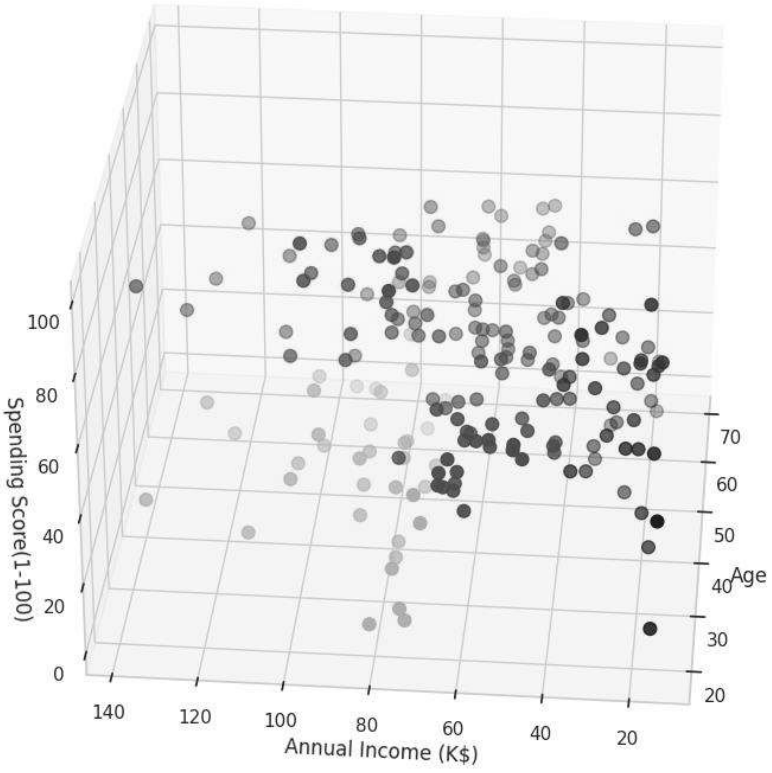


4.Visualization

• Visualize the clusters: Create visualizations (e.g., scatter plots, dendrograms) to illustrate the identified customer segments. • Interpretation of clusters: Understand the characteristics that define each cluster.

```
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()
```



5.Implementation

• Implement customer segmentation: Assign each customer to a segment based on the clustering results.

• Utilize segments for marketing strategies: Develop personalized marketing strategies for each segment

. 6.Evaluation

• Evaluate the segmentation:

Assess the effectiveness of the segmentation in improving marketing strategies and customer satisfaction.
• Refine and iterate: Make necessary adjustments to the segmentation approach based on evaluation results.

Project Execution Plan In Brief:

To proceed with solving the problem of customer segmentation, we will follow a structured approach:

Phase 1: Data Collection and Preprocessing

• Task 1.1: Identify and gather relevant data sources, including transaction history, demographic data, and online interactions.
• Task 1.2: Perform data cleaning and preprocessing to handle missing values, outliers, and data quality issues.
• Task 1.3: Combine and integrate data sources into a unified dataset.

Phase 2: Feature Engineering

• Task 2.1: Explore the dataset and identify potential features for segmentation.
• Task 2.2: Create new features or transform existing ones to capture customer behavior and preferences.
• Task 2.3: Normalize or scale features as necessary.

Phase 3: Clustering

• Task 3.1: Select appropriate clustering algorithms (e.g., K-means, hierarchical clustering).
• Task 3.2: Determine the optimal number of clusters using techniques like the elbow method or silhouette analysis.
• Task 3.3: Train the selected clustering model on the preprocessed data.

Phase 4: Visualization and Interpretation

• Task 4.1: Visualize the customer segments using appropriate plots and charts.
• Task 4.2: Interpret the characteristics and behaviors associated with each segment.

Phase 5: Implementation

• Task 5.1: Implement the customer segmentation by assigning each customer to a cluster.
• Task 5.2: Develop personalized marketing strategies for each customer segment.

Phase 6: Evaluation and Refinement

• Task 6.1: Evaluate the impact of customer segmentation on marketing strategies and customer satisfaction.
• Task 6.2: Refine the segmentation approach based on evaluation results.

- Task 6.3: Iterate on the entire process if necessary.

Advantages:

1.Personalized Marketing:

Customer segmentation allows businesses to create highly targeted marketing campaigns. They can tailor messages and promotions to specific segments, making the content more relevant and engaging for each group. This can lead to higher conversion rates and improved customer satisfaction.

2.Improved Customer Retention:

Understanding the different needs and preferences of customer segments enables businesses to provide personalized experiences and offers to keep customers coming back. This can lead to increased customer loyalty and reduced churn rates.

3.Efficient Resource Allocation:

By focusing marketing efforts and resources on the most promising customer segments, businesses can optimize their marketing budget and efforts. This can result in higher returns on investment (ROI) and cost savings.

4.Product and Service Customization:

Segmentation can help businesses identify which products or services are most popular among specific customer groups. This information can guide product development and customization efforts, leading to better alignment with customer expectations.

5.Market Expansion Opportunities:

Data-driven segmentation can reveal underserved or unexplored market segments, presenting growth opportunities for businesses. Identifying new customer segments can help diversify revenue streams.

Disadvantages:

1.Over-Simplification:

Segmentation can sometimes lead to oversimplification of customer behavior and preferences. It may not capture the full complexity of individual customer needs and motivations, potentially overlooking valuable nuances.

2.Data Quality Issues:

Effective segmentation relies on the quality of the data used for analysis. Inaccurate or incomplete data can lead to erroneous segmentation, resulting in suboptimal marketing strategies and customer experiences.

3.Privacy Concerns:

Collecting and analyzing customer data for segmentation purposes can raise privacy concerns. Businesses must handle customer data ethically and in compliance with data protection regulations like GDPR and CCPA to avoid legal and reputational risks.

4.Segment Churn:

Customer segments can change over time due to shifts in market trends, customer preferences, or other factors. Maintaining accurate and up-to-date segments can be challenging, requiring ongoing analysis and adjustment.

Benefits:

1.Reduced Marketing Costs:

Targeted marketing can lead to cost savings by avoiding wasteful spending on marketing efforts that don't resonate with specific segments. Businesses can focus their budget on the most promising segments.

2.Competitive Advantage:

Companies that effectively use customer segmentation gain a competitive advantage by delivering better, more personalized experiences than competitors who use a one-size-fits-all approach.

3.Data-Driven Decision-Making:

Customer segmentation encourages data-driven decision-making. It allows businesses to continuously refine their segmentation and marketing strategies based on real-time data and customer behavior.

Conclusion

Solving the problem of customer segmentation involves a systematic approach that covers data collection, preprocessing, feature engineering, clustering, visualization, implementation, and evaluation. By following this structured plan, we aim to provide businesses with valuable insights to personalize their marketing strategies and enhance customer satisfaction. Continuous refinement and iteration are crucial to ensure the segmentation remains effective over time.

.