**CUSTOMER SEGMENTATION USING DATA SCIENCE**

**PHASE-4**

**Introduction:**

Customer segmentation is a critical task in marketing and business analytics that involves dividing a customer base into distinct groups based on certain characteristics or behaviors. Feature engineering is the process of selecting, transforming, and creating relevant features or variables to improve the performance of a segmentation model. Here is a step-by-step guide to the feature engineering process for customer segmentation:

**Feature Selection:**

Identify the relevant features (variables) that are likely to influence your segmentation goals. This might include demographic information, purchase history, website interactions, location data, etc. Start with a comprehensive set and then narrow it down as needed.

**Code:**

#We take just the Annual Income and Spending score

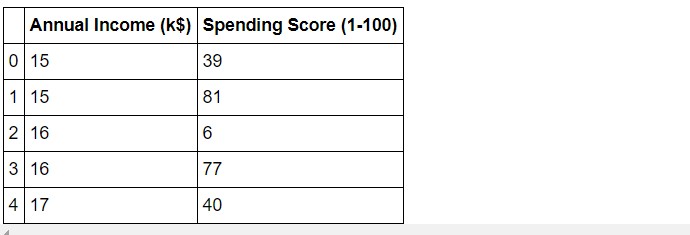
df1=data[["CustomerID","Gender","Age","Annual Income (k$)","Spending Score (1-100)"]]

X=df1[["Annual Income (k$)","Spending Score (1-100)"]]

#The input data

X.head()

**Output:**

****

**Applying K-Means Algorithm:**

Model Building and Evaluation:

Use a suitable segmentation algorithm (e.g., k-means, hierarchical clustering, or machine learning models) to segment your customers based on the engineered features. Evaluate the model's performance using appropriate metrics such as Silhouette Score, Davies-Bouldin Index, or business-specific KPIs.

#Taking 5 clusters

km1=KMeans(n\_clusters=5)

#Fitting the input data

km1.fit(X)

#predicting the labels of the input data

y=km1.predict(X)

#adding the labels to a column named label

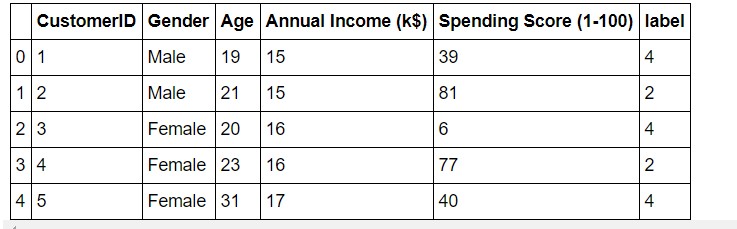
df1["label"] = y

#The new dataframe with the clustering done

df1.head()

#The labels added to the data.

**Output:**



**Visualization:**

Visualizing customer segmentation is essential for understanding the distinct groups and their characteristics. Here are several common visualization techniques you can use to represent customer segmentation results in Python:

**Code:**

#Scatterplot of the input data

plt.figure(figsize=(10,6))

sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)', data = X ,s = 60 )

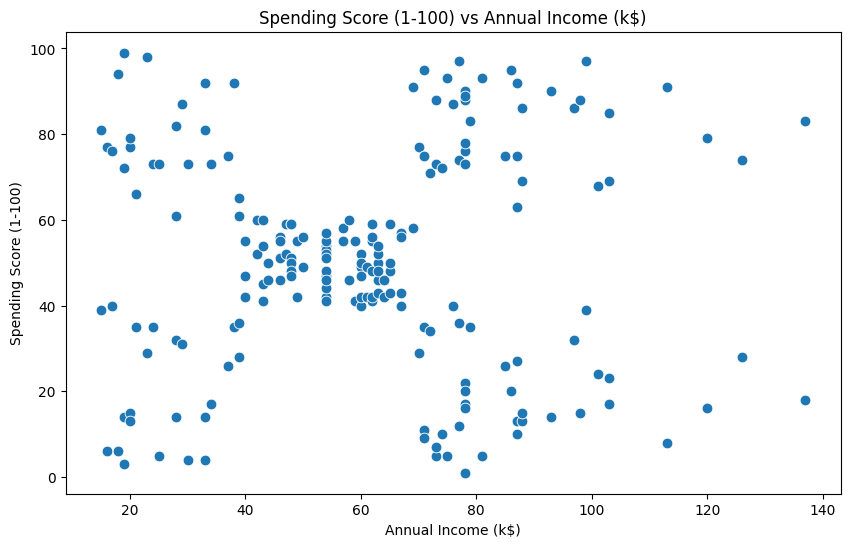
plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Spending Score (1-100) vs Annual Income (k$)')

plt.show()

**Output:**

****

**Code:**

#Scatterplot of the clusters

plt.figure(figsize=(10,6))

sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',hue="label",

palette=['green','orange','brown','dodgerblue','red'], legend='full',data = df1 ,s = 60 )

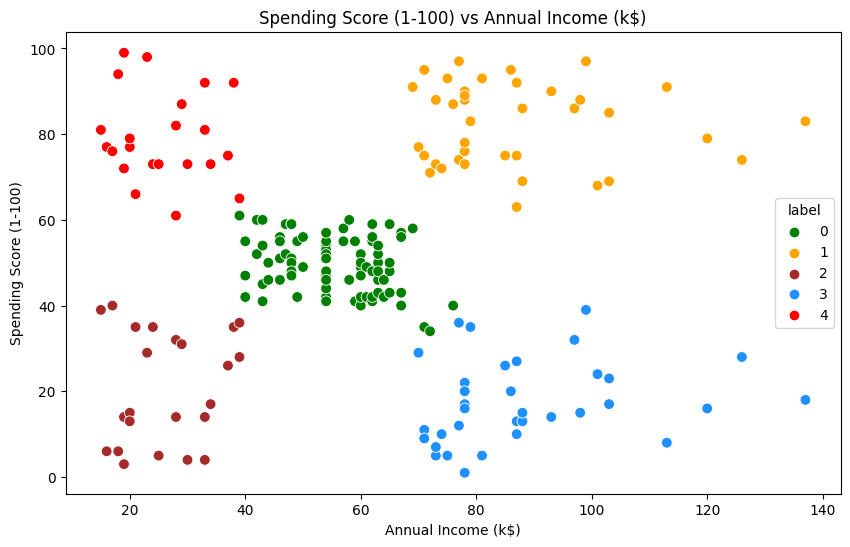
plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Spending Score (1-100) vs Annual Income (k$)')

plt.show()

**Output:**

****

**3D Visualization:**

**Code:**

#3D Plot as we did the clustering on the basis of 3 input features

fig = plt.figure(figsize=(20,10))

ax = fig.add\_subplot(111, projection='3d')

ax.scatter(df2.Age[df2.label == 0], df2["Annual Income (k$)"][df2.label == 0], df2["Spending Score (1-100)"][df2.label == 0], c='purple', s=60)

ax.scatter(df2.Age[df2.label == 1], df2["Annual Income (k$)"][df2.label == 1], df2["Spending Score (1-100)"][df2.label == 1], c='red', s=60)

ax.scatter(df2.Age[df2.label == 2], df2["Annual Income (k$)"][df2.label == 2], df2["Spending Score (1-100)"][df2.label == 2], c='blue', s=60)

ax.scatter(df2.Age[df2.label == 3], df2["Annual Income (k$)"][df2.label == 3], df2["Spending Score (1-100)"][df2.label == 3], c='green', s=60)

ax.scatter(df2.Age[df2.label == 4], df2["Annual Income (k$)"][df2.label == 4], df2["Spending Score (1-100)"][df2.label == 4], c='yellow', s=60)

ax.view\_init(35, 185)

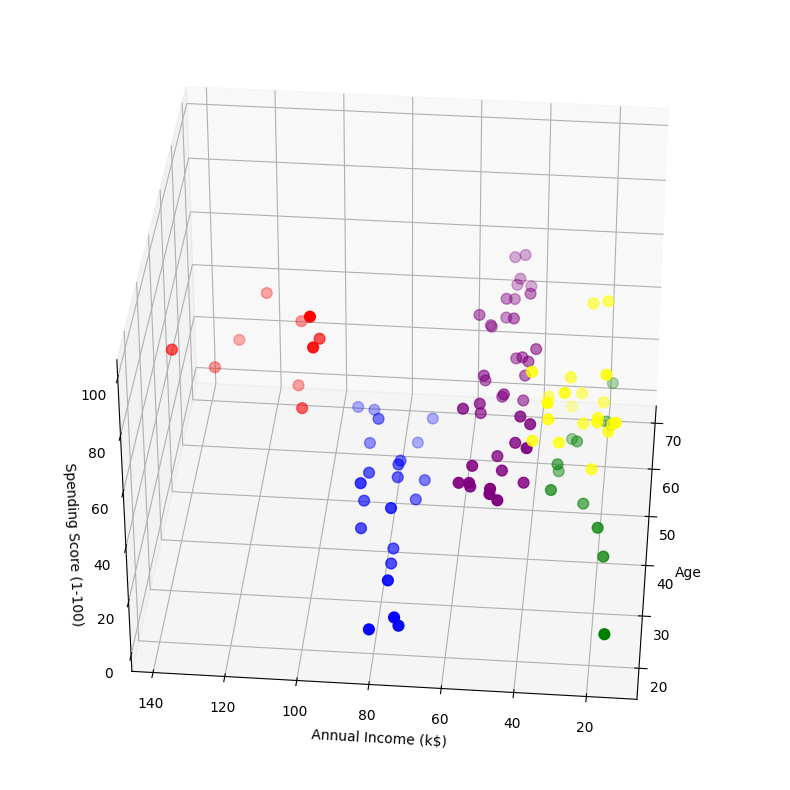
plt.xlabel("Age")

plt.ylabel("Annual Income (k$)")

ax.set\_zlabel('Spending Score (1-100)')

plt.show()

**Output:**

****

**Interpretation:**

Finally, you interpret the clusters by analyzing the statistics and the visualizations, give them meaningful names, and use the insights to drive business actions specific to each segment. Adjust the code to match your specific dataset and clustering results, and be sure to perform a more in-depth analysis based on your domain knowledge and segmentation goals.

**Code:**

cust1=df2[df2["label"]==1]

print('Number of customer in 1st group=', len(cust1))

print('They are -', cust1["CustomerID"].values)

print("--------------------------------------------")

cust2=df2[df2["label"]==2]

print('Number of customer in 2nd group=', len(cust2))

print('They are -', cust2["CustomerID"].values)

print("--------------------------------------------")

cust3=df2[df2["label"]==0]

print('Number of customer in 3rd group=', len(cust3))

print('They are -', cust3["CustomerID"].values)

print("--------------------------------------------")

cust4=df2[df2["label"]==3]

print('Number of customer in 4th group=', len(cust4))

print('They are -', cust4["CustomerID"].values)

print("--------------------------------------------")

cust5=df2[df2["label"]==4]

print('Number of customer in 5th group=', len(cust5))

print('They are -', cust5["CustomerID"].values)

print("--------------------------------------------")

**Output:**

Number of customer in 1st group= 10

They are - [182 184 186 188 190 192 194 196 198 200]

--------------------------------------------

Number of customer in 2nd group= 22

They are - [129 131 135 137 139 141 145 149 151 153 155 157 159 163 165 167 169 171

173 175 177 179]

--------------------------------------------

Number of customer in 3rd group= 40

They are - [44 46 47 48 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70

71 72 73 74 75 76 77 78 79 81 82 85 86 87 88 89]

--------------------------------------------

Number of customer in 4th group= 12

They are - [ 3 7 9 11 13 15 23 25 31 33 35 37]

--------------------------------------------

Number of customer in 5th group= 21

They are - [ 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42] --------------------------------------------