**Title:**

Customer Segmentationwith Machine Learnings

**Introduction:**

Preprocessing of customer segmentation data involves a series of critical tasks aimed at cleaning, organizing, and enhancing the raw data before it is used for analysis and segmentation. This introductory stage is essential to ensure that the segmentation process yields meaningful and actionable insights.

**I) Data Selection and loading:**

Data selection and loading are vital steps in the data preparation process for machine learning. Data selection involves choosing the right dataset that aligns with your objectives, ensuring it contains relevant information. Data loading refers to the process of importing the chosen dataset into your machine learning environment, such as a Python notebook or database. This step includes reading data from files (e.g., CSV, Excel) or databases, making it ready for further analysis. Accurate data selection and loading set the foundation for successful machine learning tasks by providing the right information for model training and evaluation.

**Code:**

#Importing the necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from mpl\_toolkits.mplot3d import Axes3D

%matplotlib inline

#Reading the excel file

data=pd.read\_excel("Mall\_Customers.xlsx")

data.head()

data.describe()

Output:

|index|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|

<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

**II) Data Cleaning:**

- Data often contains missing values, outliers, and inconsistencies that need to be addressed. Missing values can be filled in through imputation or removed if necessary. Outliers can be adjusted or flagged.

- Importance: Cleaning data ensures that the model is not influenced by errors or outliers, leading to more robust and accurate predictions.

**Code:**

# Data cleaning and handling missing values

print(df.drop\_duplicates())

print(df.dropna())

print(df.isna().any())

print(df.isna().sum())

cf=df.copy()

Output:

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

.. ... ... ... ... ...

195 196 Female 35 120 79

196 197 Female 45 126 28

197 198 Male 32 126 74

198 199 Male 32 137 18

199 200 Male 30 137 83

[200 rows x 5 columns]

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

.. ... ... ... ... ...

195 196 Female 35 120 79

196 197 Female 45 126 28

197 198 Male 32 126 74

198 199 Male 32 137 18

199 200 Male 30 137 83

[200 rows x 5 columns]

CustomerID False

Gender False

Age False

Annual Income (k$) False

Spending Score (1-100) False

dtype: bool

CustomerID 0

Gender 0

Age 0

Annual Income (k$) 0

Spending Score (1-100) 0

dtype: int64

**III) Feature Selection:**

In some cases, not all variables or features in the data may be equally relevant for segmentation. Feature selection identifies and retains the most valuable attributes for the segmentation process while discarding less informative ones.

**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:**

|index|Annual Income \(k$\)|Spending Score \(1-100\)|

|---|---|---|

|0|15|39|

|1|15|81|

|2|16|6|

|3|16|77|

|4|17|40|

**IV) Data Exploration:**

Data exploration is a crucial initial step in the process of customer segmentation. It involves gaining a deeper understanding of your customer data, identifying patterns, trends, and characteristics that can inform your segmentation strategy.

**Data correlation:**

Code:

data.corr()

dataplot = sns.heatmap(data.corr(), cmap="YlGnBu", annot=True)

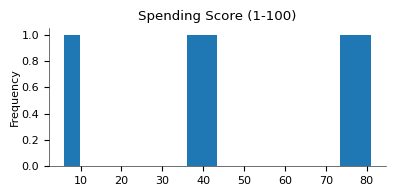
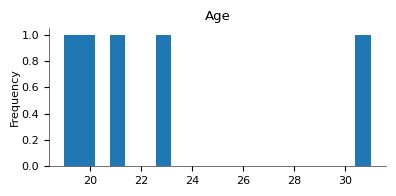
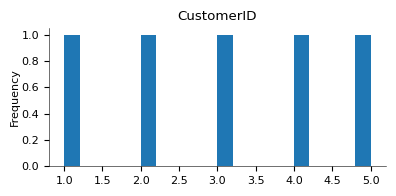
# displaying heatmap

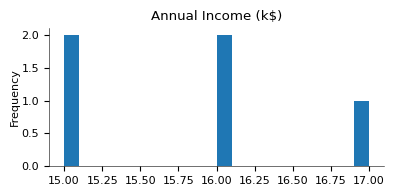
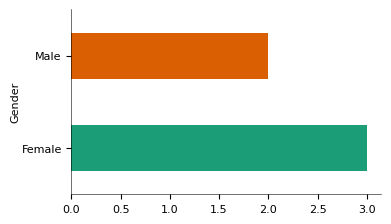
plt.show()

Output:



**Feature distributions:**

** **

****

**V)Data visualization:**

Visualize your data to reveal patterns and relationships. Create histograms, bar charts, scatter plots, and other relevant charts to examine the distribution of variables and correlations between them. Tools like Matplotlib, Seaborn, or ggplot2 are useful for data visualization.

**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 )

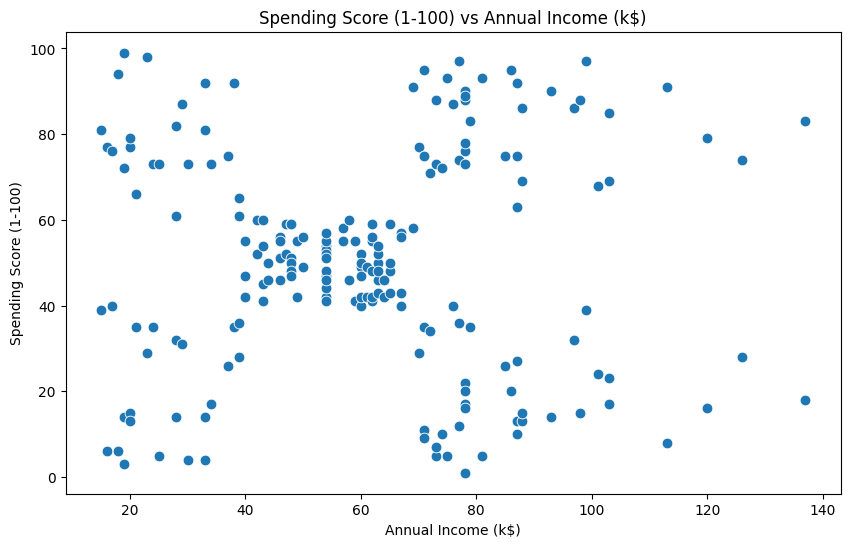
plt.xlabel('Annual Income (k$)')

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

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

plt.show()

**Output:**

****

**Conclusion:**

In summary, preprocessing is a critical phase in Customer segmentation with machine learning. Proper data preparation ensures that the model is trained on high-quality, relevant data, which ultimately results in more accurate predictions. Each preprocessing step plays a crucial role in addressing different data challenges and improving the model's ability to forecast product demand.