CUSTOMER SEGEMENTATION

INTRODUCTION

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.

In this phase the building and loading of data flow of customer segmentation is going to be done.

PREREQUISITES FOR BUILDING A CUSTOMER SEGMENTATION MODEL

- The data is obtained from https://www.Kaggle.com/data
- Have the following libraries installed—Numpy, Pandas, Matplotlib, Seaborn, Scikit-Learn, Kneed, and Scipy.
- Columns Required from dataset
    1. CustomerID
    2. Gender
    3. Age
    4. Annual Income
    5. Spending Score

UNDERSTAND THE SEGMENTATION DATA

Before starting any data science project, it is vital to explore the dataset and understand each variable.

- Libraries Imported :

    1. Numpy
    2. Pandas
    3. Matplotlib
    4. Seaborn

- Loading the Data

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

- let’s look at the head of the dataframe:

df.head()

	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

PREPROCESSING DATA FOR SEGMENTATION

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.

- Description

df.describe()

Out[4]:

	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

- Null Values

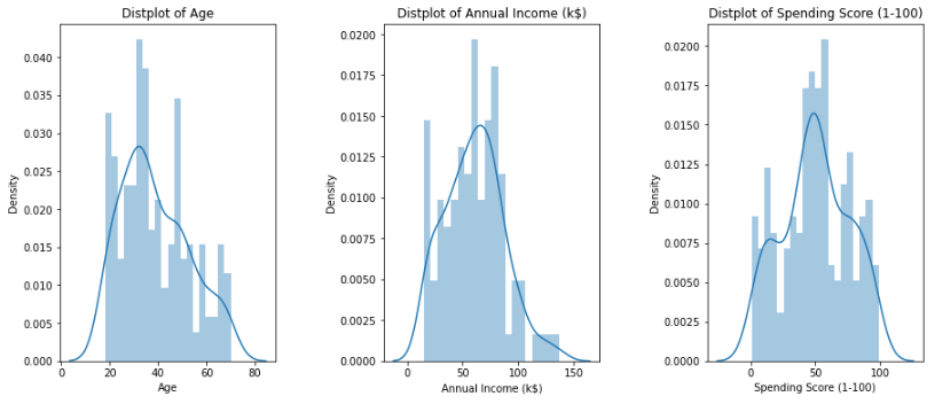
```
df.isnull().sum()
```

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

- Dropping

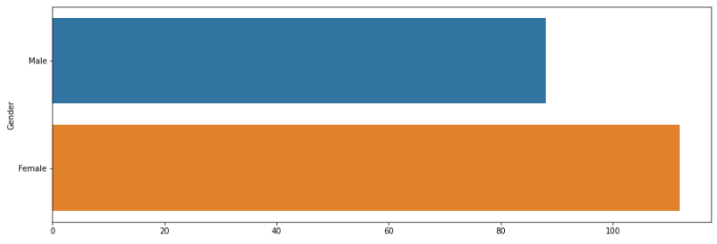
```
df.drop(['CustomerID'],axis=1,inplace=True)
```

```
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()
```



- CounterPlot

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



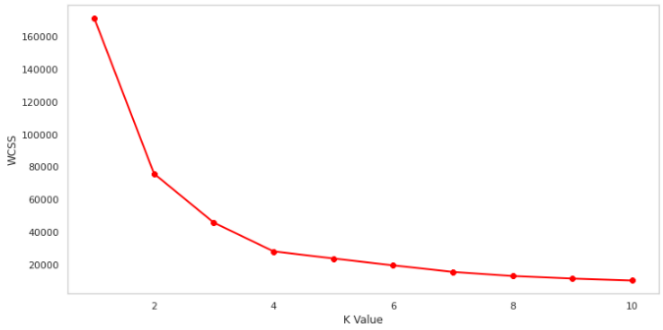
BUILDING THE CUSTOMER SEGMENTATION MODEL

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.

- 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 Cluster

```
kmeans = KMeans(n_clusters=4)

label = kmeans.fit_predict(X1)

print(label)
```

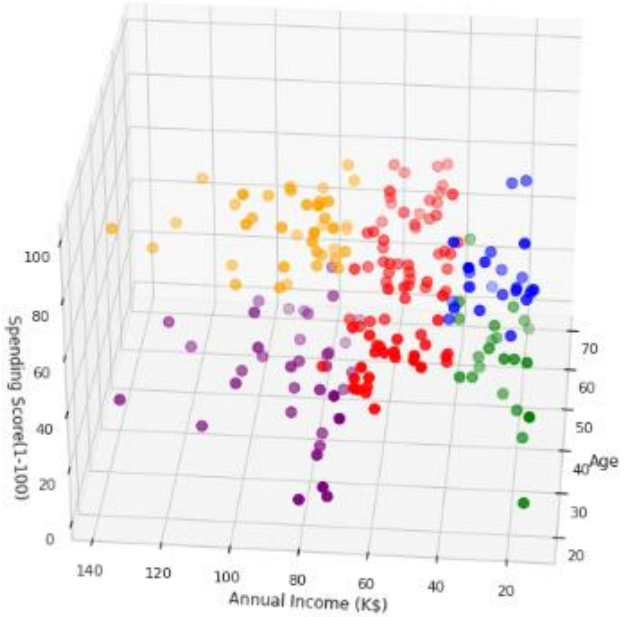
```
[1 2 0 2 1 2 0 2 0 2 0 2 0 2 0 2 1 1 0 2 1 2 0 2 0 2 0 1 0 2 0 2 0 2 0 2 0
 2 0 2 3 2 3 1 0 1 3 1 1 1 3 1 1 3 3 3 3 3 1 3 3 1 3 3 3 1 3 3 1 1 3 3 3 3
 3 1 3 1 1 3 3 1 3 3 1 3 3 1 1 3 3 1 3 1 1 1 3 1 3 1 1 3 3 1 3 1 3 3 3 3 3
 1 1 1 1 1 3 3 3 3 1 1 1 2 1 2 3 2 0 2 0 2 1 2 0 2 0 2 0 2 0 2 1 2 0 2 3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

- Clusters 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
```

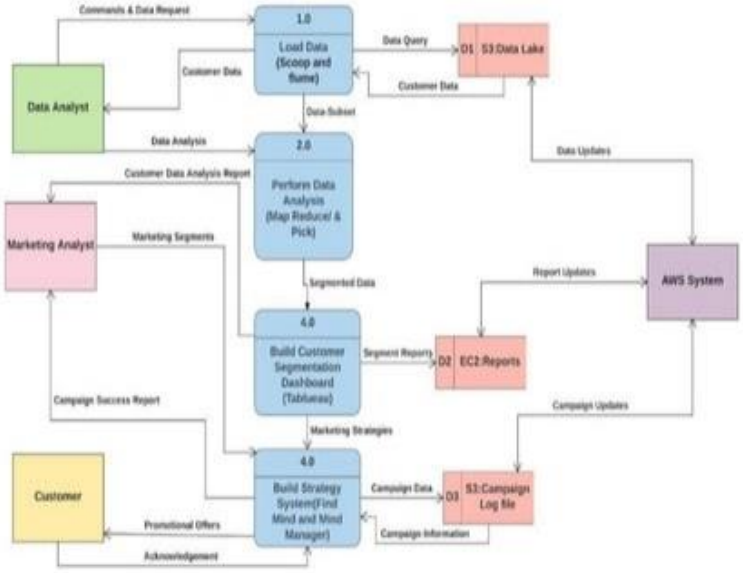
```
<function matplotlib.pyplot.show(close=None, block=None)>
```

- 3D Model



DATA FLOW OF CUSTOMER MODEL

1.Physical Flow



2.Logical Flow