MALL CUSTOMER SEGMENTATION USING K-MEANS CLUSTERING

CH.RISHITHA
AP19110010441

OVERVIEW:

Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics. Customer Segmentation can be a powerful means to identify unsatisfied customer needs. I want to increase customer lifetime value by segmenting the customers into several groups with similar characteristics and form growth strategies for each group.

Clustering algorithms try to find natural clusters in data, the various aspects of how the algorithms to cluster data can be tuned and modified. Clustering is based on the principle that items within the same cluster must be similar to each other. The data is grouped in such a way that related elements are close to each other.

In this project we have taken a dataset of mall customers and cluster them using k-means method. K-Means clustering is an algorithm that divides the given data into the given number of clusters. Here, the "K" is the given number of predefined clusters, that need to be created.Finding the optimal number of clusters is an important part of this algorithm. A commonly used method for finding optimal K value is Elbow Method. It is a centroid based algorithm in which each cluster is associated with centroid. The main idea is to reduce the distance between the data points and their respective cluster centroid.

Steps involved:

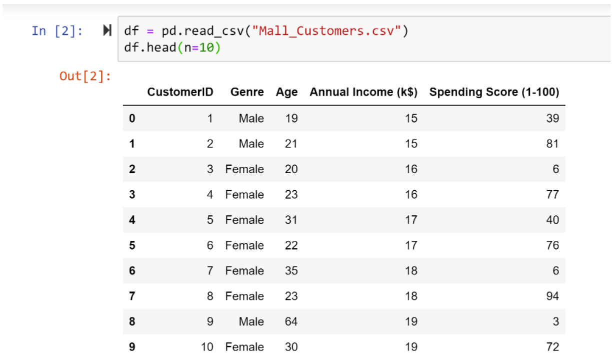
1. DataSet selection

2. Data Preprocessing

3. Applying K mean

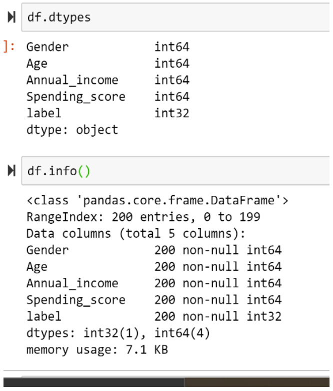
4.Clustering

1.DataSet selection and dependencies: We have selected a dataset of customers in a mall which includes the following features: Customer Id,Customer Gender,Customer Age, Annual Income,Spendingscore



```
In [2]:  df = pd.read_csv("Mall_Customers.csv")
         df.head(n=10)
Out[2]:
```

	CustomerID	Genre	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
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72



```
df.dtypes

]:  Gender            int64
    Age               int64
    Annual_income     int64
    Spending_score    int64
    label             int32
    dtype: object

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
Gender            200 non-null int64
Age               200 non-null int64
Annual_income     200 non-null int64
Spending_score    200 non-null int64
label             200 non-null int32
dtypes: int32(1), int64(4)
memory usage: 7.1 KB
```

* Dependencies: Here we can find the libraries we will use in order to develop a solution for this problem.

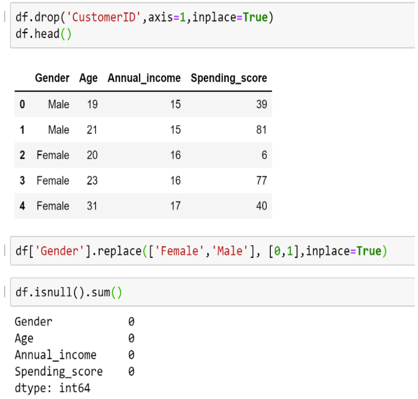
numpy|pandas: Will help us treat and explore the data, and execute vector and matrix operations.

matplotlib|seaborn: Will help us plot the information so we can visualize it in different ways and have a better understanding of it.

plotly: Will also help us plotting data in a fancy way.

sklearn: Will provide all necessary tools to train our models and test them afterwards

2.Data Preprocessing: we need to clean the dataset and apply a machine learning model for better results. First, we checked for any duplicates are there in the dataset and then we went for Data Cleaning which includes removing null values, renaming errors in spellings, dropping unwanted columns,combine and replacing two columns into one.



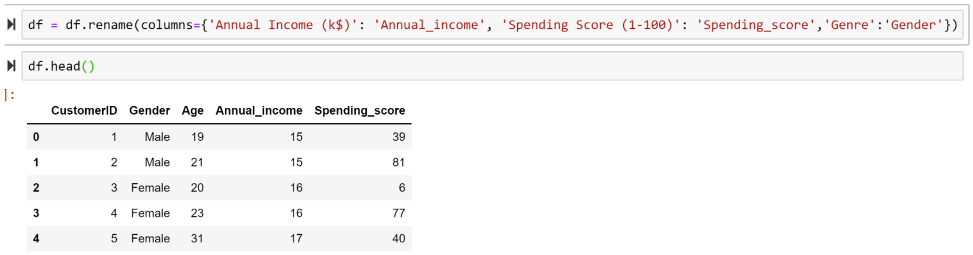
```
df.drop('CustomerID',axis=1,inplace=True)
df.head()
```

	Gender	Age	Annual_income	Spending_score
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

```
df['Gender'].replace(['Female','Male'], [0,1],inplace=True)
```

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

```
Gender            0
Age               0
Annual_income     0
Spending_score    0
dtype: int64
```



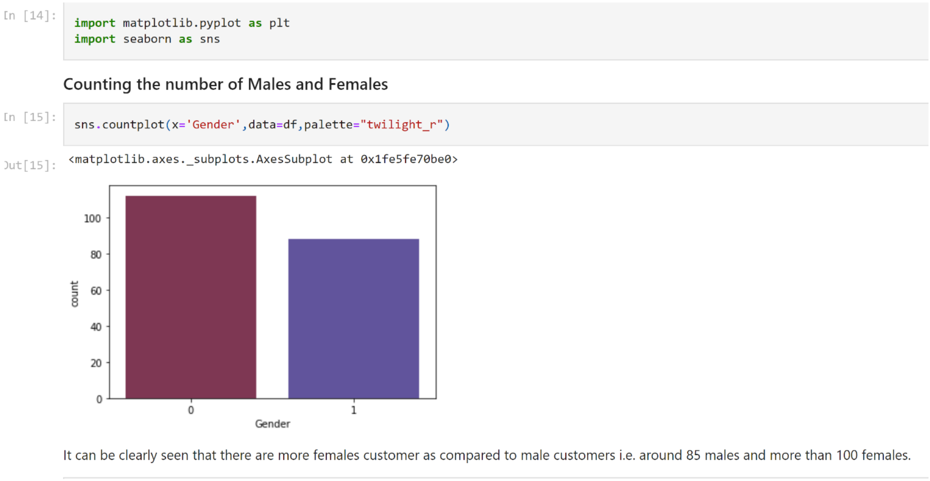
```
df = df.rename(columns={'Annual Income (k$)': 'Annual_income', 'Spending Score (1-100)': 'Spending_score','Genre':'Gender'})
```

```
df.head()
```

]:

	CustomerID	Gender	Age	Annual_income	Spending_score
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

After preprocessing we analyse the data visually .Data visualization is the graphical representaion of data which helps us to get better insight by plotting all the data collectively using matplot and seaborn library.

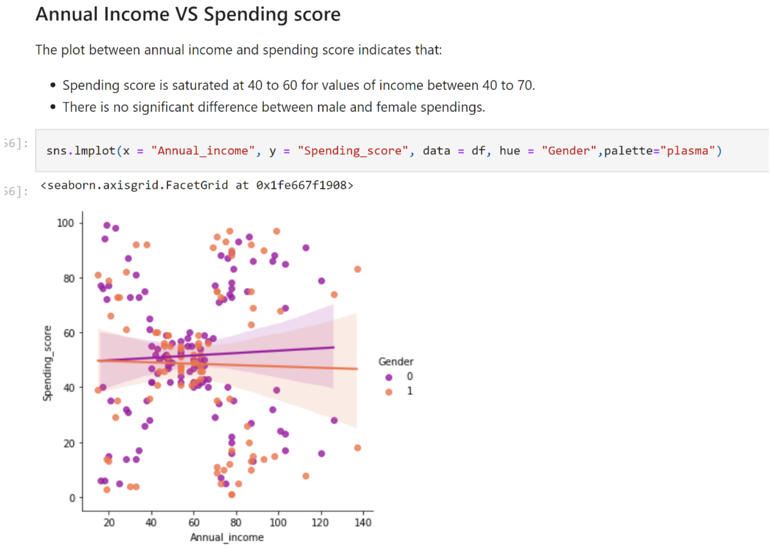


```
[n [14]:  import matplotlib.pyplot as plt
          import seaborn as sns
```

Counting the number of Males and Females

```
[n [15]:  sns.countplot(x='Gender',data=df,palette="twilight_r")

Out[15]:  <matplotlib.axes._subplots.AxesSubplot at 0x1fe5fe70be0>
```

It can be clearly seen that there are more females customer as compared to male customers i.e. around 85 males and more than 100 females.



Annual Income VS Spending score

The plot between annual income and spending score indicates that:

- Spending score is saturated at 40 to 60 for values of income between 40 to 70.
- There is no significant difference between male and female spendings.

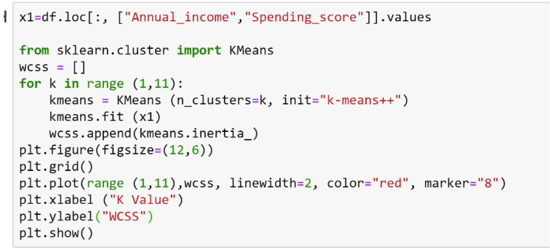
```
56]:  sns.lmplot(x = "Annual_income", y = "Spending_score", data = df, hue = "Gender",palette="plasma")

56]:  <seaborn.axisgrid.FacetGrid at 0x1fe667f1908>
```

3. Applying clustering method :

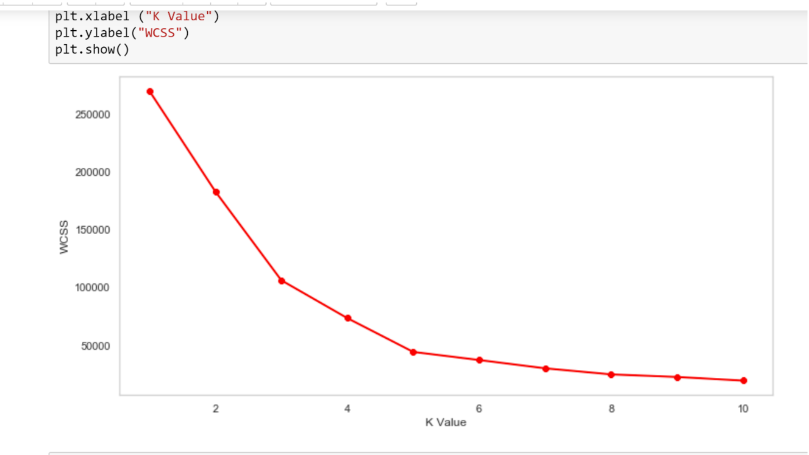
K-Means clustering is an algorithm that divides the given data into the given number of clusters. Here, the “K” is the given number of predefined clusters, that need to be created.Finding the optimal number of clusters is an important part of this algorithm. A commonly used method for finding optimal K value is Elbow Method.

In the Elbow method, we are actually varying the number of clusters (K) from 1 – 10. For each value of K, we are calculating WCSS (Within-Cluster Sum of Square). WCSS is the sum of squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow. As the number of clusters increases, the WCSS value will start to decrease. WCSS value is largest when K = 1. When we analyze the graph we can see that the graph will rapidly change at a point and thus creating an elbow shape. From this point, the graph starts to move almost parallel to the X-axis. The K value corresponding to this point is the optimal K value or an optimal number of clusters.

It plots the value of the cost function produced by different values of k and one should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. In this problem, we are using the **inertia** as cost function in order to identify the sum of squared distances of samples to the nearest cluster centre.



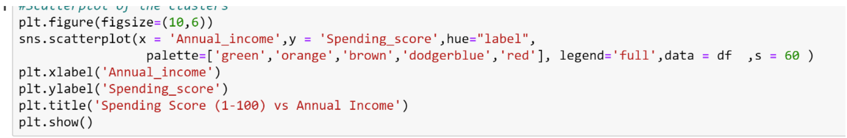
```
x1=df.loc[:, ["Annual_income","Spending_score"]].values

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()
```



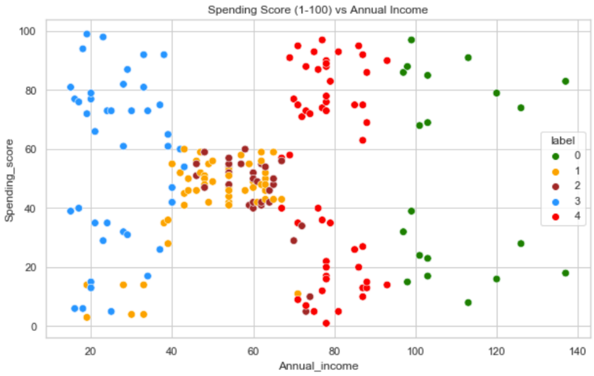
```
plt.xlabel ("K Value")
plt.ylabel("WCSS")
plt.show()
```

As we can observe, the K-means algorithm has already finished its work and now it's time to plot the results we obtained by it so we can visualize the different clusters and analyze them.

4.CLUSTERING



```
#Scatterplot of the clusters
plt.figure(figsize=(10,6))
sns.scatterplot(x = 'Annual_income',y = 'Spending_score',hue="label",
                palette=['green','orange','brown','dodgerblue','red'], legend='full',data = df  ,s = 60 )
plt.xlabel('Annual_income')
plt.ylabel('Spending_score')
plt.title('Spending Score (1-100) vs Annual Income')
plt.show()
```



From the above result we can say that
Label 0(green)-people with high income and equal amount of high and low spending score
Label 1(orange)-people with moderate income and moderate spending
Label 2(brown)-people with moderate income and moderate to low spending
Label 3(dodgerblue)-people with low income and high to low spending score
Label 4(red)-people with moderate income and equal amount of high and low spending score

Conclusion:

- KMeans Clustering is a powerful technique in order to achieve a decent customer segmentation.
- Customer segmentation is a good way to understand the behaviour of different customers and plan a good marketing strategy accordingly.
- There isn't much difference between the spending score of LABEL 4 AND LABEL 3, which leads us to think that their behaviour when it comes to shopping is pretty similar.

- There are more people we have to consider, like people who belong to the orange cluster-label-1, they are what we would commonly name after "middle class" and it seems to be the biggest cluster.