COMSATS UNIVERSITY ISLAMABAD

SUBMITTED BY

NAME: NOOR UL AIN

REG. NO.: FA19-BSE-168

SECTION: BSE-7C

SUBMITTED TO

TEACHER: Ms. MAMUNA FATIMA

SUBJECT: DATA WAREHOUSING & DATA MINING

LAB TASK # 07

Lab Tasks:

1. Write purpose of numpy, pandas and sklearn library in your own words.

Numpy

NumPy stands for 'Numerical Python' or 'Numeric Python'. It is an open-source module of Python which provides fast mathematical computation on arrays and matrices.

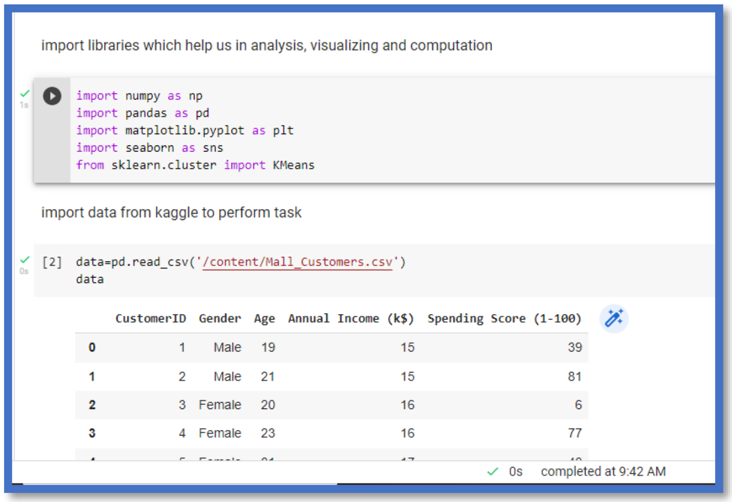
Pandas

Pandas is a Python library for data analysis. Pandas is built on top of two core Python libraries—**matplotlib** for data visualization and **NumPy** for mathematical operations.

Sklearn

Scikit-learn (formerly scikits. learn and also known as sklearn) is **a free software machine learning library for the Python programming language**.

2. Understand the code above and write the purpose of each code statement in your own words.
3. Execute the above code with provided data.



import libraries which help us in analysis, visualizing and computation

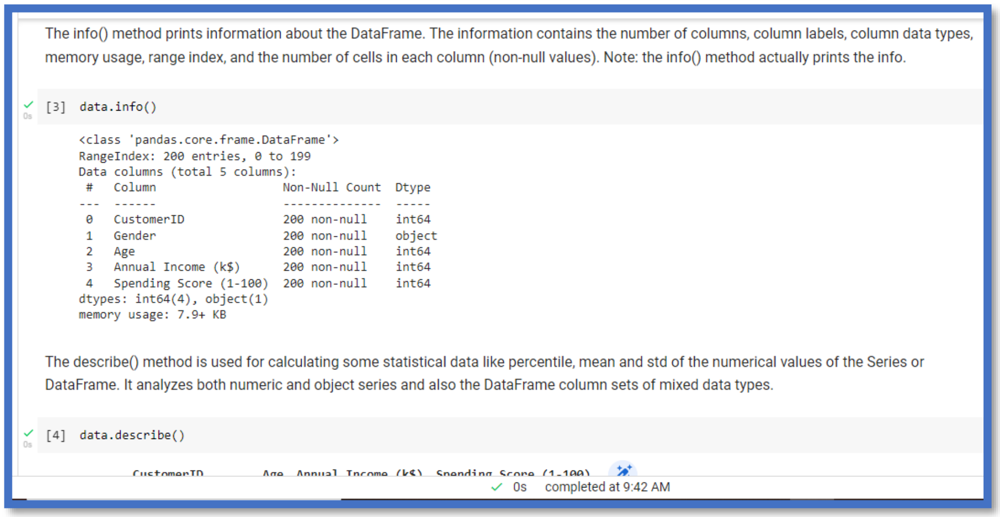
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

import data from kaggle to perform task

```python
[2] data=pd.read_csv('/content/Mall_Customers.csv')
    data
```

	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

✓ 0s completed at 9:42 AM



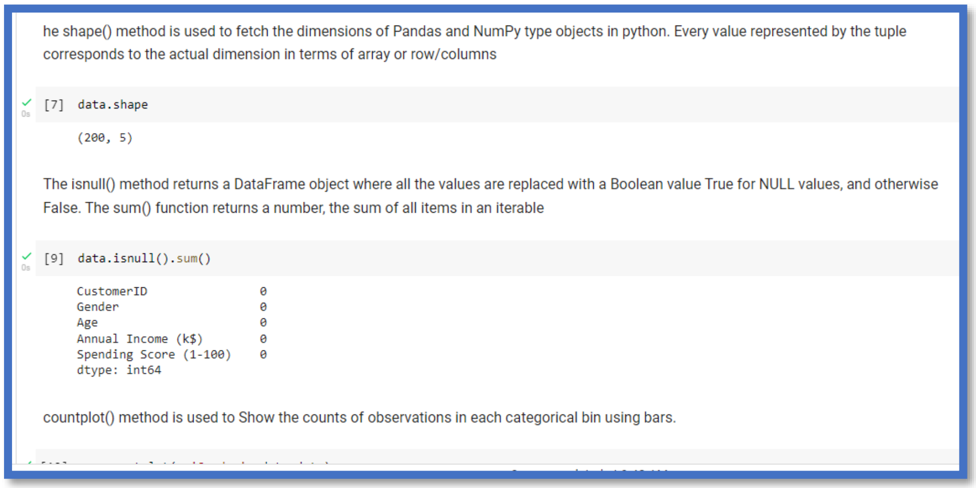
The info() method prints information about the DataFrame. The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values). Note: the info() method actually prints the info.

```
[3] data.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
```

The describe() method is used for calculating some statistical data like percentile, mean and std of the numerical values of the Series or DataFrame. It analyzes both numeric and object series and also the DataFrame column sets of mixed data types.

```
[4] data.describe()
```

✓ 0s completed at 9:42 AM



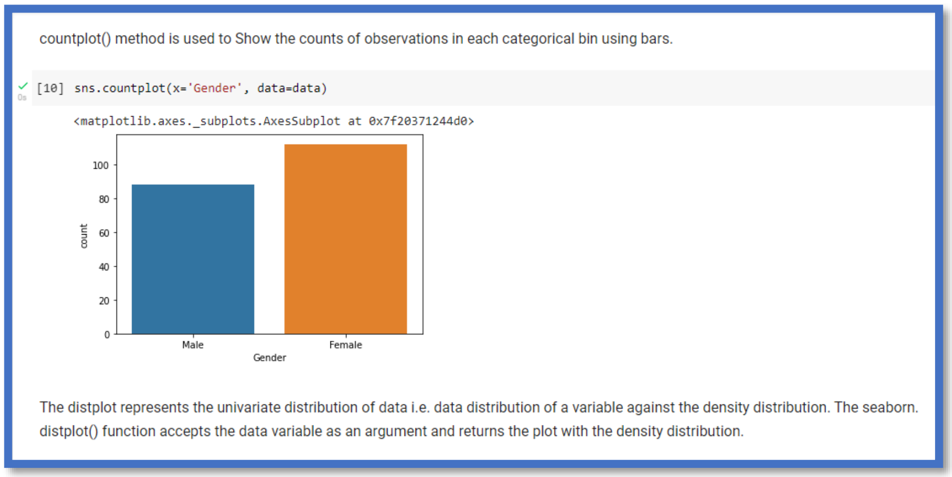
he shape() method is used to fetch the dimensions of Pandas and NumPy type objects in python. Every value represented by the tuple corresponds to the actual dimension in terms of array or row/columns

```
[7]  data.shape

     (200, 5)
```

The isnull() method returns a DataFrame object where all the values are replaced with a Boolean value True for NULL values, and otherwise False. The sum() function returns a number, the sum of all items in an iterable

```
[9]  data.isnull().sum()

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

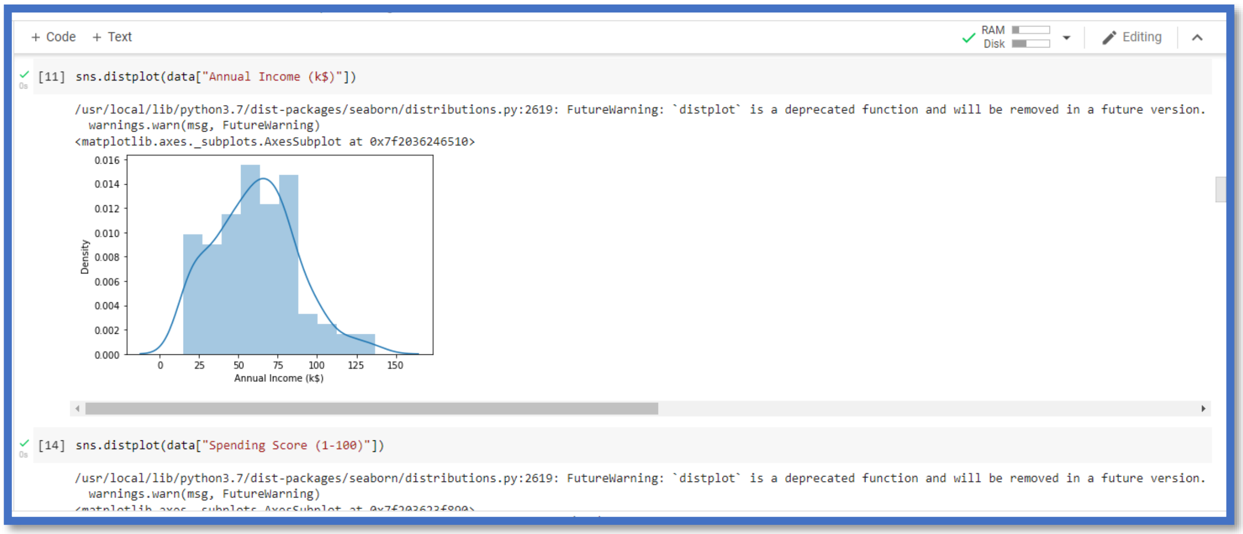
countplot() method is used to Show the counts of observations in each categorical bin using bars.



countplot() method is used to Show the counts of observations in each categorical bin using bars.

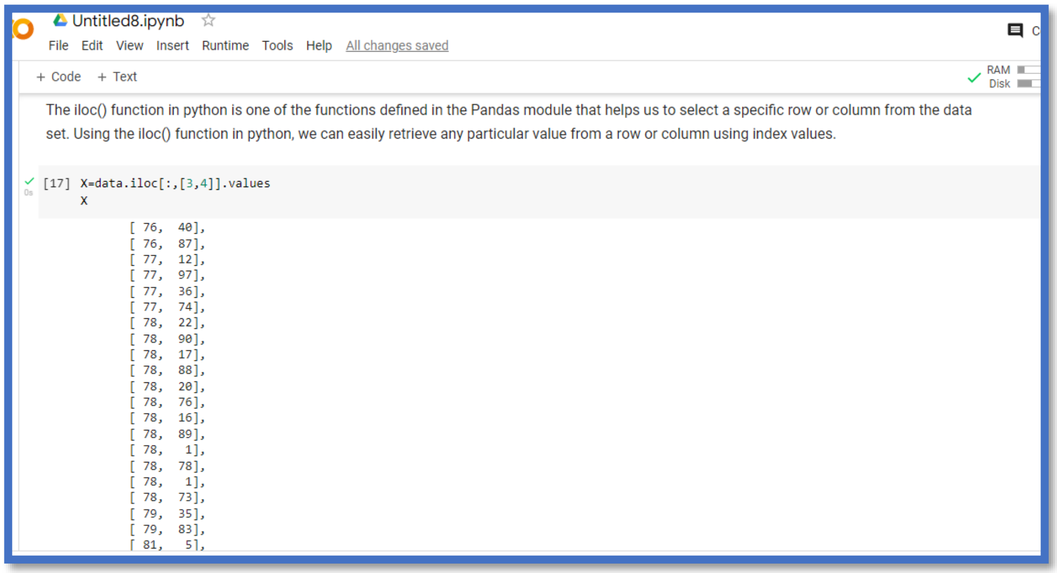
```
[10] sns.countplot(x='Gender', data=data)

     <matplotlib.axes._subplots.AxesSubplot at 0x7f20371244d0>
```

The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution. The seaborn. distplot() function accepts the data variable as an argument and returns the plot with the density distribution.



```
[11] sns.distplot(data["Annual Income (k$)"])

     /usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version.
       warnings.warn(msg, FutureWarning)
     <matplotlib.axes._subplots.AxesSubplot at 0x7f2036246510>
```

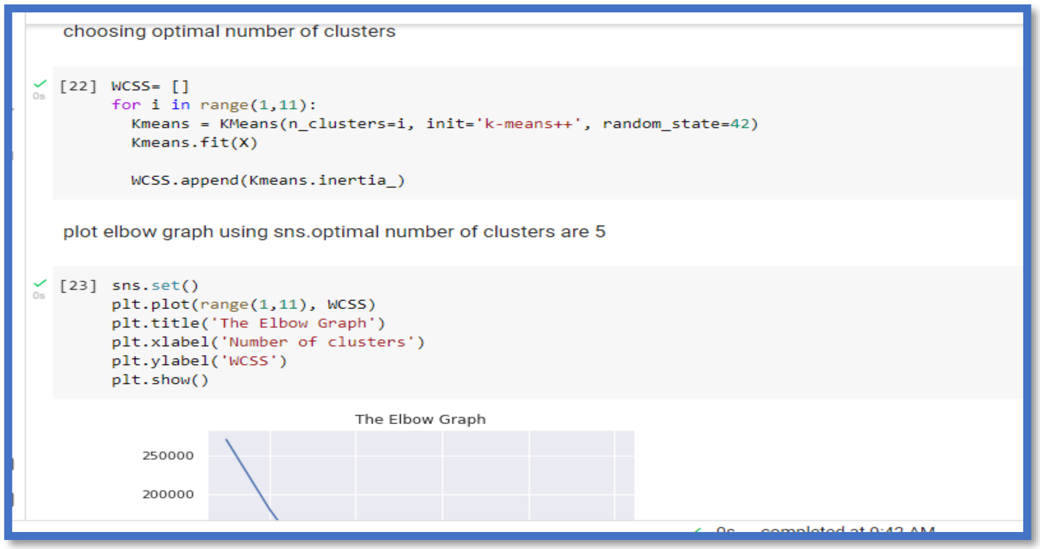
```
[14] sns.distplot(data["Spending Score (1-100)"])

     /usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version.
       warnings.warn(msg, FutureWarning)
```



The iloc() function in python is one of the functions defined in the Pandas module that helps us to select a specific row or column from the data set. Using the iloc() function in python, we can easily retrieve any particular value from a row or column using index values.

```
[17] X=data.iloc[:,[3,4]].values
     X
```

```
        [ 76,  40],
        [ 76,  87],
        [ 77,  12],
        [ 77,  97],
        [ 77,  36],
        [ 77,  74],
        [ 78,  22],
        [ 78,  90],
        [ 78,  17],
        [ 78,  88],
        [ 78,  20],
        [ 78,  76],
        [ 78,  16],
        [ 78,  89],
        [ 78,   1],
        [ 78,  78],
        [ 78,   1],
        [ 78,  73],
        [ 79,  35],
        [ 79,  83],
        [ 81,   5],
```

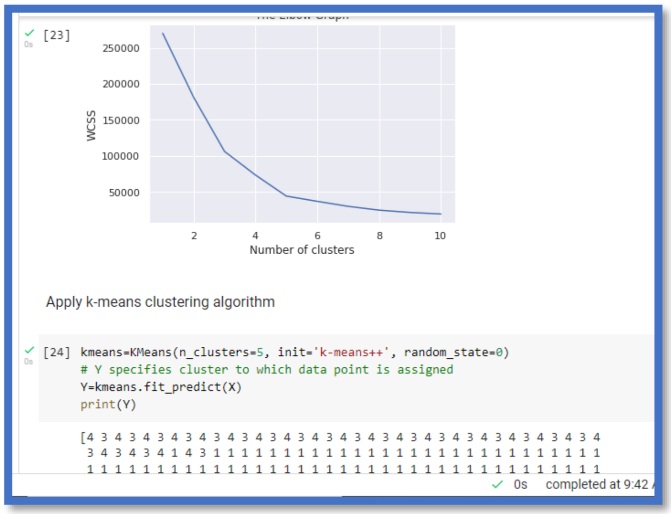


choosing optimal number of clusters

```
[22] WCSS= []
     for i in range(1,11):
       Kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
       Kmeans.fit(X)

       WCSS.append(Kmeans.inertia_)
```

plot elbow graph using sns.optimal number of clusters are 5

```
[23] sns.set()
     plt.plot(range(1,11), WCSS)
     plt.title('The Elbow Graph')
     plt.xlabel('Number of clusters')
     plt.ylabel('WCSS')
     plt.show()
```

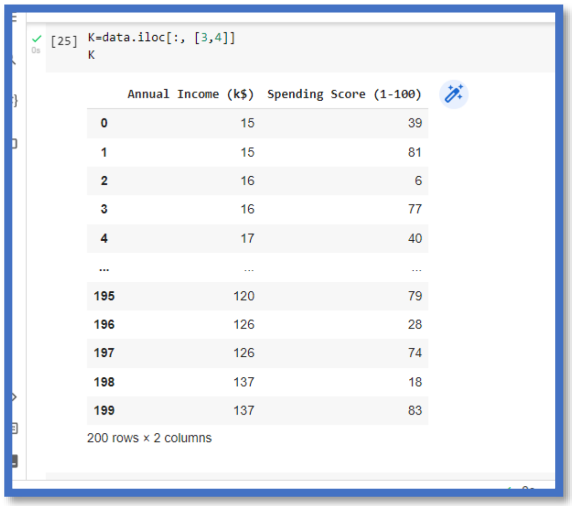


```
[23]
```

Apply k-means clustering algorithm

```
[24] kmeans=KMeans(n_clusters=5, init='k-means++', random_state=0)
     # Y specifies cluster to which data point is assigned
     Y=kmeans.fit_predict(X)
     print(Y)
```

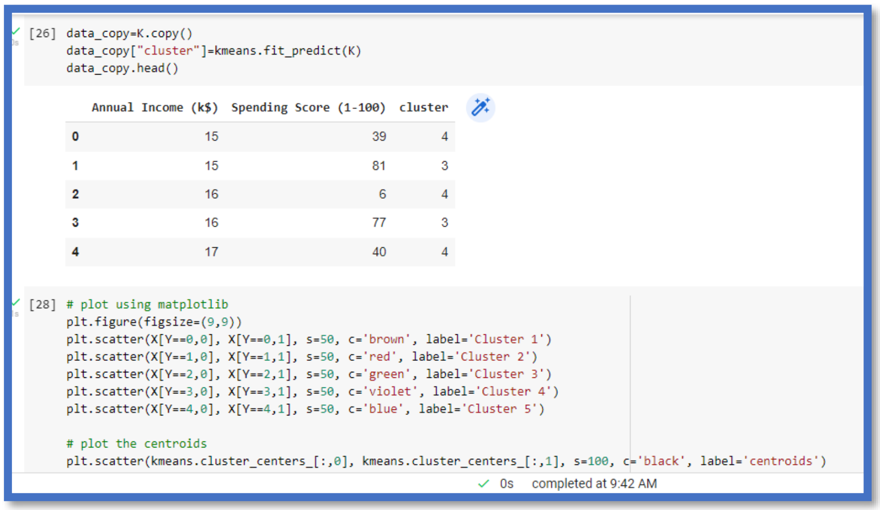
```
[4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4
 3 4 3 4 3 4 1 4 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```



```
[25] K=data.iloc[:, [3,4]]
     K
```

	Annual Income (k$)	Spending Score (1-100)
0	15	39
1	15	81
2	16	6
3	16	77
4	17	40
...	...	...
195	120	79
196	126	28
197	126	74
198	137	18
199	137	83

200 rows × 2 columns

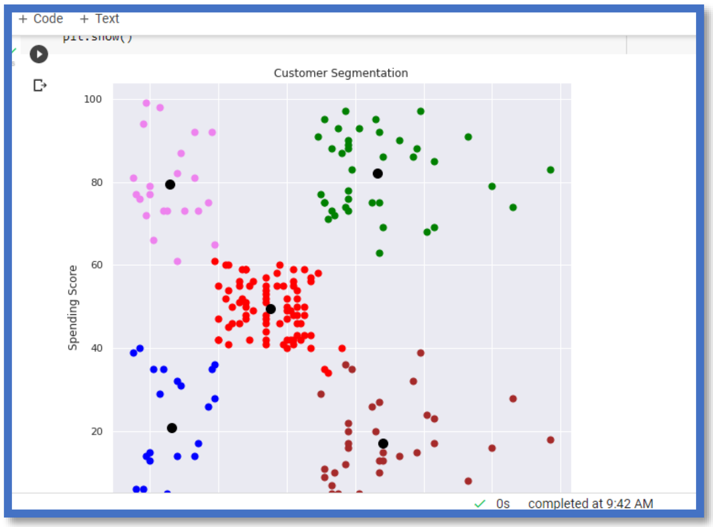


```
[26] data_copy=K.copy()
     data_copy["cluster"]=kmeans.fit_predict(K)
     data_copy.head()
```

	Annual Income (k$)	Spending Score (1-100)	cluster
0	15	39	4
1	15	81	3
2	16	6	4
3	16	77	3
4	17	40	4

```
[28] # plot using matplotlib
     plt.figure(figsize=(9,9))
     plt.scatter(X[Y==0,0], X[Y==0,1], s=50, c='brown', label='Cluster 1')
     plt.scatter(X[Y==1,0], X[Y==1,1], s=50, c='red', label='Cluster 2')
     plt.scatter(X[Y==2,0], X[Y==2,1], s=50, c='green', label='Cluster 3')
     plt.scatter(X[Y==3,0], X[Y==3,1], s=50, c='violet', label='Cluster 4')
     plt.scatter(X[Y==4,0], X[Y==4,1], s=50, c='blue', label='Cluster 5')

     # plot the centroids
     plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], s=100, c='black', label='centroids')
```

✓ 0s completed at 9:42 AM


+ Code
+ Text
plt.show()
Customer Segmentation
Spending Score
✓ 0s completed at 9:42 AM