Adebusola Adewole c00448223

Exam 1

1. Write a paragraph that describes YOUR scientific paper (i.e., topic, objectives, ML models, etc.)

   Topic : Customer Segmentation

The goal of my scientific paper is to understand and classify customers using behavior and spending habits to make strategic decisions concerning products to increase company profit. Customer segmentation is the process of categorizing customers into groups that reflect similarities in their purchase history. I will be using an unsupervised/supervised machine learning approach to classify customers from an ecommerce store. Based on this analysis, this will allow me anticipate what purchase will be made by a new customer the following year.

Models- K-mean clustering, Random forest

2. Indicate the dataset that you are using for your scientific paper (or, in the case of a survey paper by undergraduate students, a dataset that relates to your survey topic).

I will be using data from an ecommerce store that sells household items. It contains a purchased made by 4000 customers over a period of 1 year 2010/12/01 to 2011/12/09). This dataset has been gotten from Kaggle. It contains

3. Identify a dataset that pertains to your scientific paper, which is DIFFERENT FROM the dataset you are using (or plan to use) for your paper.

The different data set I will be using for my analysis is data set from a supermarket which shows the different spending habits of their customers

4. Demonstrate your use and application of the Machine Learning Project Checklist (Appendix B), applying the 8 steps indicated in the Checklist, using at least one or more ML models from Chapters 1 through 4, inclusive. Be sure to use the dataset you identify in #3 above -- that is, do NOT use the dataset that you are using (or will use) for your scientific paper.

Frame the problem

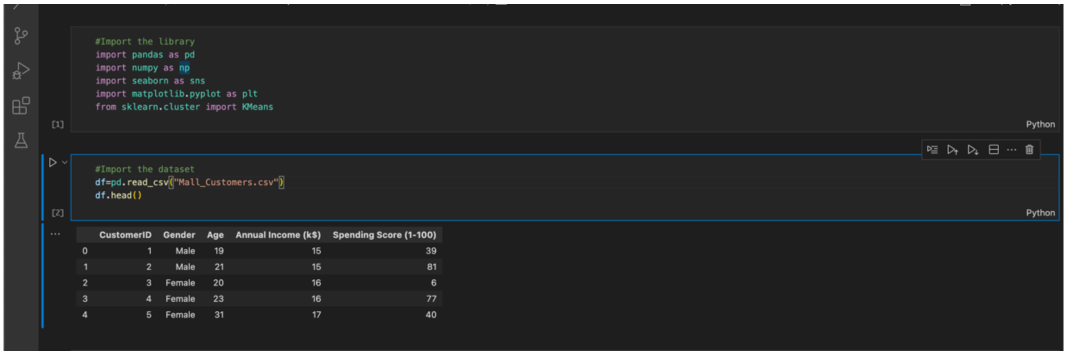
Separation of customers into groups is important for organizations because it helps you better understand them and meet their needs It also helps the marketing team make strategic decisions concerning product growth. The business task is that this company hasn't grouped their customers and are having a hard time understanding their customers needs. I have been tasked with understanding their customers based on their spending habits and classifying them into groups based on certain traits that they share. Spending score has been assigned to the customers based on customer behavior and purchasing power. The goal of this project is to segment customers a machine learning algorithm so that the marketing team can identify the target customers and start a marketing strategy.

This problem is an unsupervised machine learning problem. The performance of my model will be evaluated using the sum of squared distance between the data points and all centroids.

If I were to solve this problem manually it would be time consuming. Having to manually separate customers based on their spending habits would be a difficult task.

Get the data

I'll be needing data about customer spending habits from the ecommerce website. My data will be gotten from Kaggle. The size of the data is about 4KB. There are no legal obligation or authorizations necessary to get this data since I'll be downloading it from Kaggle. If I were in data scientist at the company I would need to obtain authorization from the IT department. The data would also need to be anonymized to prevent bias and protect the customers information. I have downloaded the data as a csv file, which is a sufficient format for my project. Also, I have confirmed that sensitive information has been anonymized.



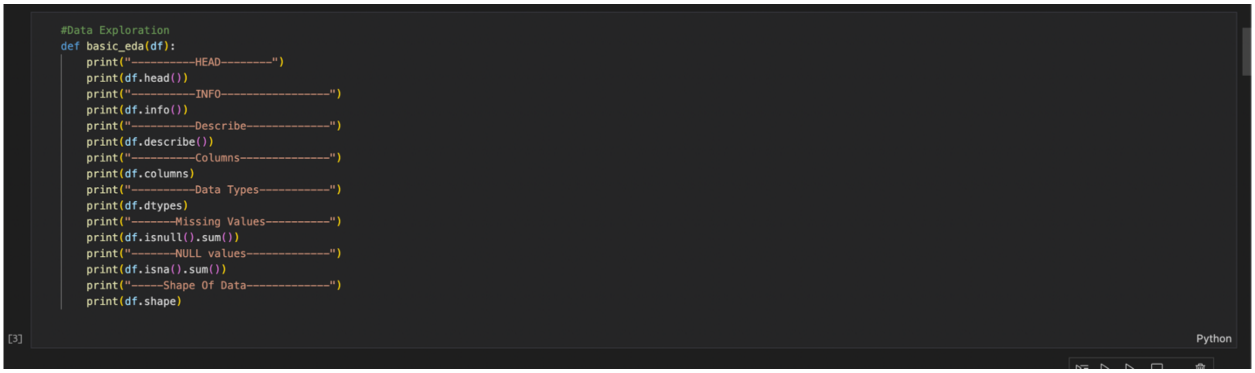
```python
#Import the library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
```

```python
#Import the dataset
df=pd.read_csv("Mall_Customers.csv")
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
```

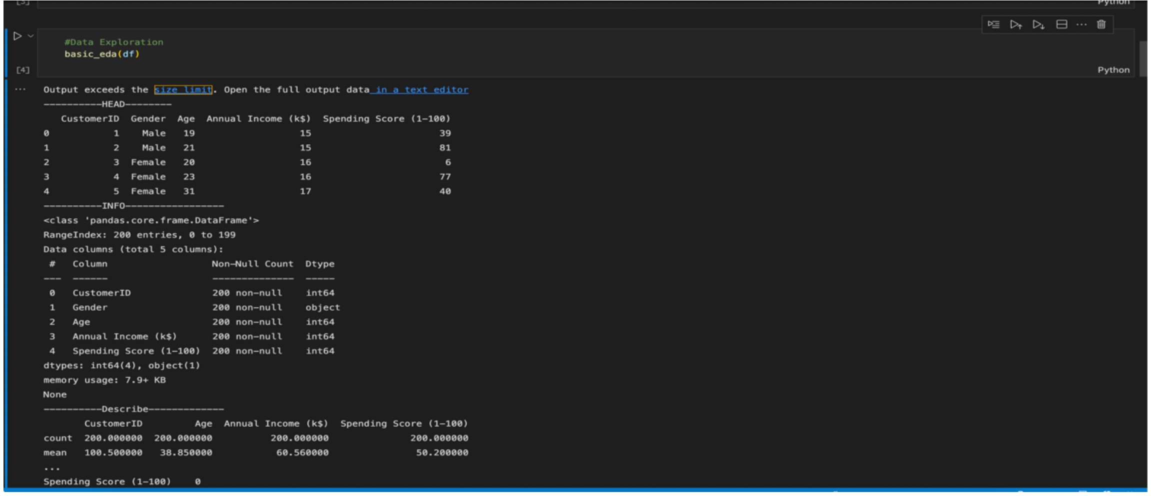
S1: Necessary libraries have been downloaded and my data has been loaded into my workspace (Jupyter notebook)

Explore and Prepare the Data For Machine learning



```python
#Data Exploration
def basic_eda(df):
    print("----------HEAD--------")
    print(df.head())
    print("----------INFO-----------------")
    print(df.info())
    print("----------Describe-------------")
    print(df.describe())
    print("----------Columns--------------")
    print(df.columns)
    print("----------Data Types-----------")
    print(df.dtypes)
    print("-------Missing Values----------")
    print(df.isnull().sum())
    print("-------NULL values-------------")
    print(df.isna().sum())
    print("-----Shape Of Data-------------")
    print(df.shape)
```

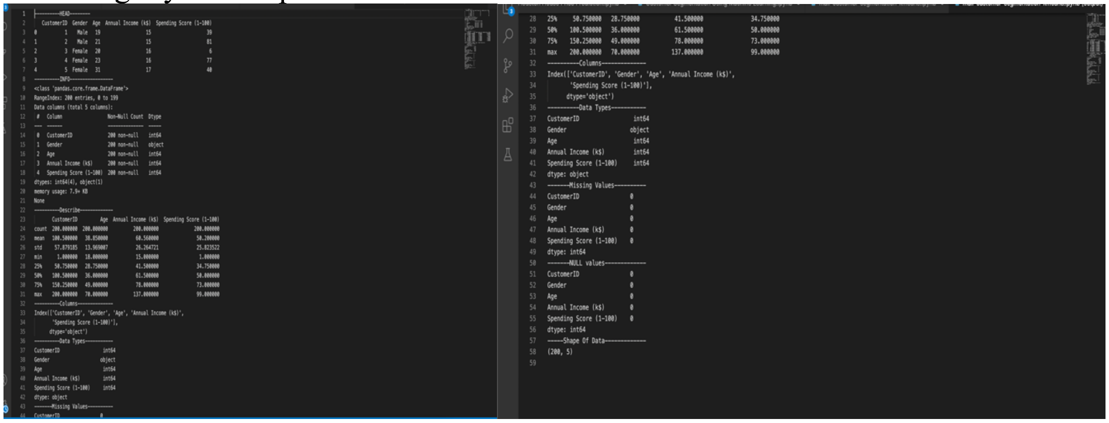
S2: I created a function for basic data exploration of my data set to save time. The function displays the first 5 rows of the data, data type of the columns, 5 number summary, the number of missing and values and the shape of my data set



```python
#Data Exploration
basic_eda(df)
```

```
Output exceeds the size limit. Open the full output data in a text editor
----------HEAD--------
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
----------INFO-----------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
None
----------Describe-------------
       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
...
Spending Score (1-100)    0
```

S3: Using my data exploration function on the data



```
----------HEAD--------
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
----------INFO-----------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
None
----------Describe-------------
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
----------Columns--------------
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
----------Data Types-----------
CustomerID                 int64
Gender                    object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
-------Missing Values----------
CustomerID                0
```

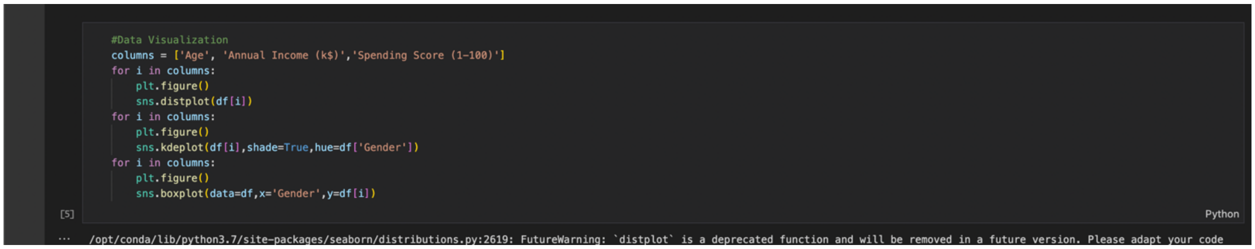
```
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
----------Columns--------------
Index(['CustomerID', 'Gender', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
----------Data Types-----------
CustomerID                 int64
Gender                    object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
-------Missing Values----------
CustomerID                0
```

```
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
-------NULL values-------------
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
-----Shape Of Data-------------
(200, 5)
```

S4: Results

Observation

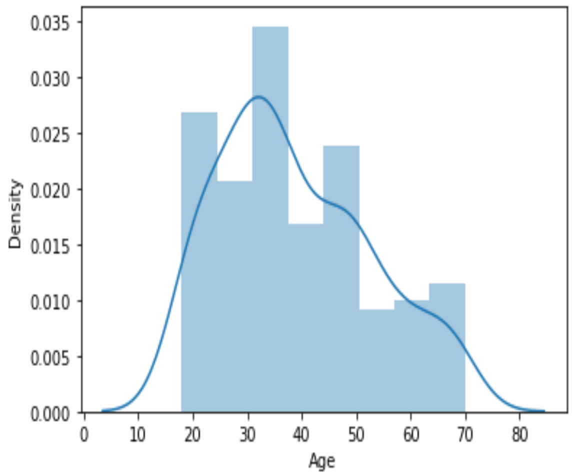
- The number of rows in the dataset is 200 while the number of columns is 5
- The data is clean and there no duplicates or null values
- There are 4 numerical columns and 1 categorical column

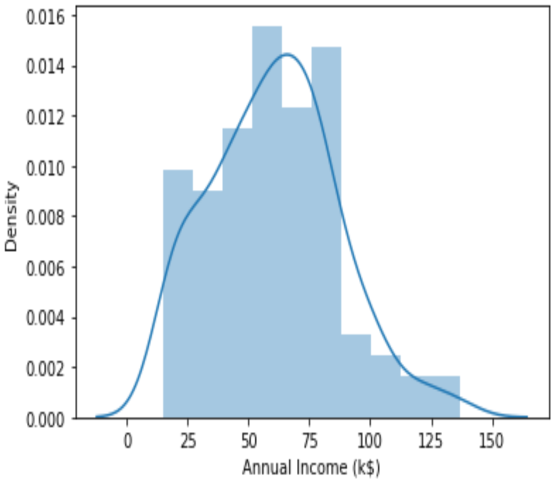


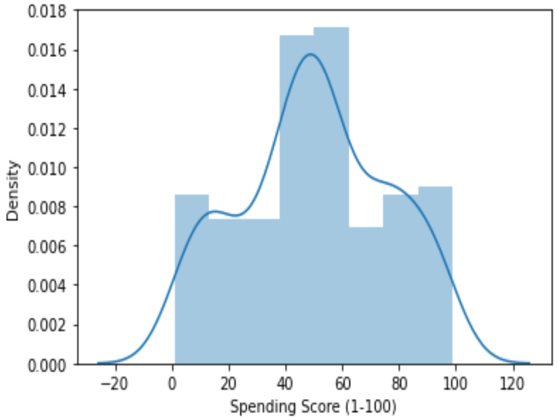
```
#Data Visualization
columns = ['Age', 'Annual Income (k$)','Spending Score (1-100)']
for i in columns:
    plt.figure()
    sns.distplot(df[i])
for i in columns:
    plt.figure()
    sns.kdeplot(df[i],shade=True,hue=df['Gender'])
for i in columns:
    plt.figure()
    sns.boxplot(data=df,x='Gender',y=df[i])
```

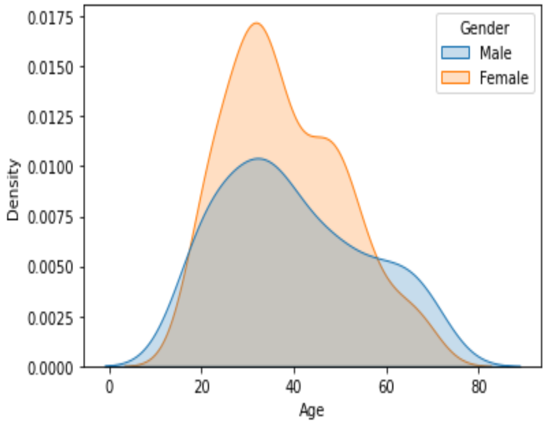
```
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code
```

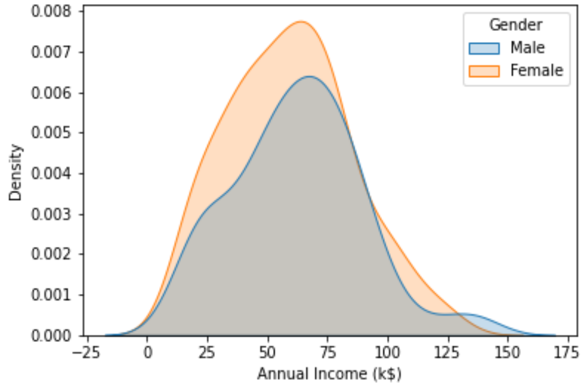
S5: Visualizations to better understand the distributions, correlations between attributes

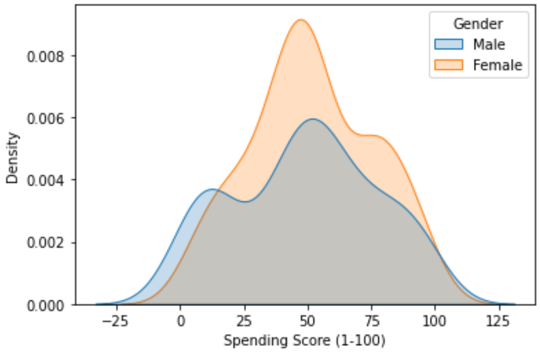

Density
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
10
20
30
40
50
60
70
80
Age

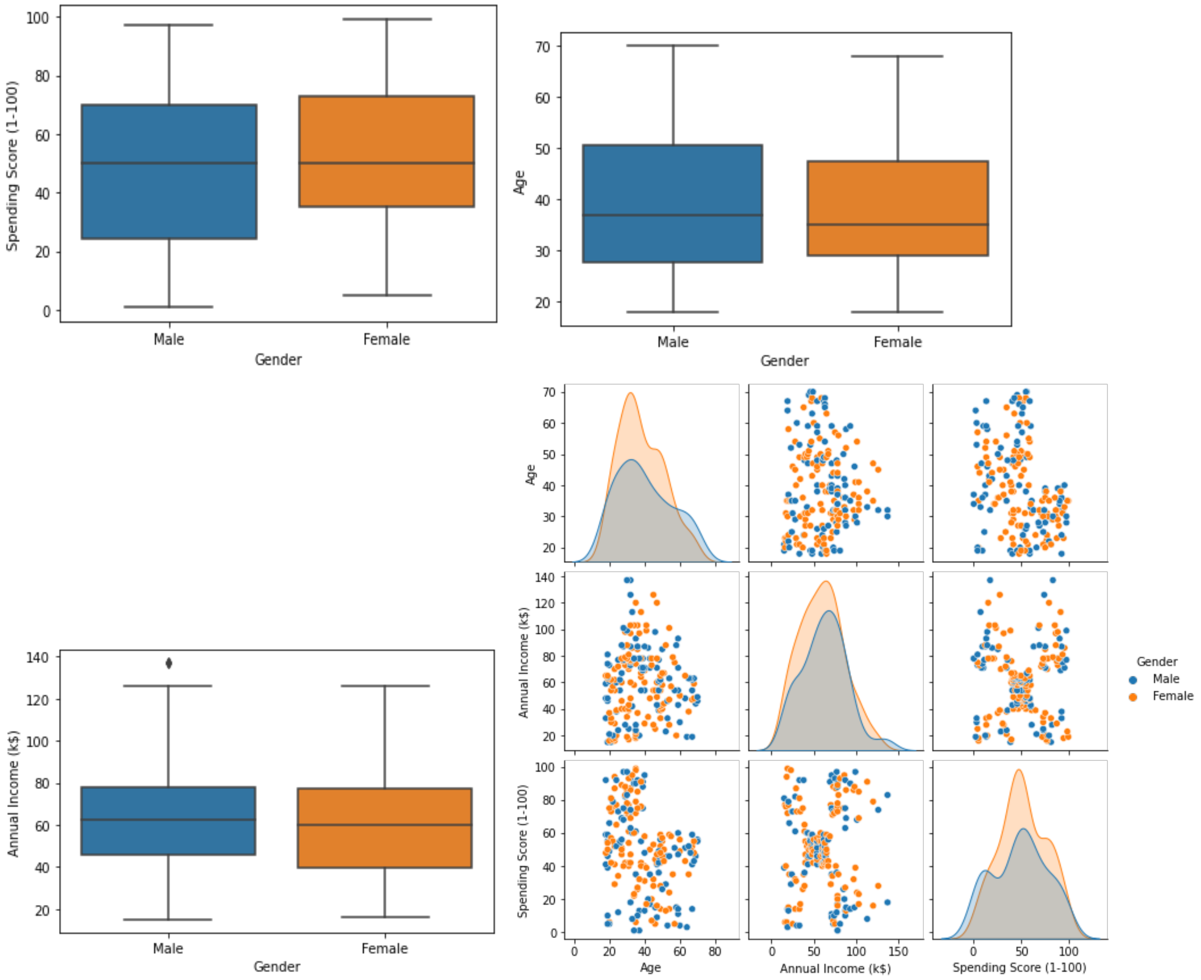

Density
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
25
50
75
100
125
150
Annual Income (k$)


Density
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
−20
0
20
40
60
80
100
120
Spending Score (1-100)


Gender
Male
Female
Density
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
20
40
60
80
Age


Gender
Male
Female
Density
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
−25
0
25
50
75
100
125
150
175
Annual Income (k$)


Gender
Male
Female
Density
0.008
0.006
0.004
0.002
0.000
−25
0
25
50
75
100
125
Spending Score (1-100)



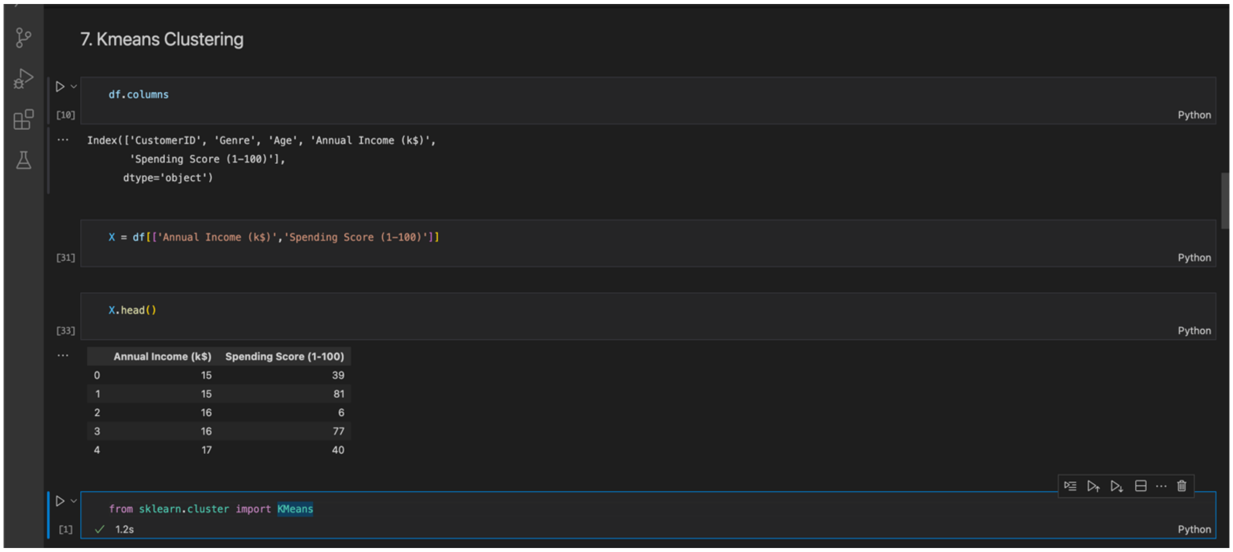
S6: Data visualizations showing how the different variables are related

Observation

- Customers who are 32 years have the most visits to the store and mean age of customers is 38.2
- the annual income in mall per visitor is 54 K
- The average spending score is 50
- From the correlation plot we can conclude there is no correlation in the data set. Although, there is a negative relationship between age and spending score
- The histograms show that the data is normally distributed
- 56% of the customers are female
- The average age of female customers is 39 and 38 years for male customers
- The average income per year for female customers is 59k while for male customer it is 62k
- Total annual income for female customers (6636K) is higher than male customers (5476K)
- Female customers tend to have a higher spending score than male customers

Select a model and train it

I'll be using an unsupervised machine learning technique; K-means Clustering. It is the process of dividing the dataset into groups in which the members in the same group possess similarities. For this, we randomly initialize the K numbers of centroids. This model works by calculating the Euclidean distance or Manhattan distance and assigns the points to the nearest centroid, thus creating K groups. Afterwards, it finds the original centroid in each group and reassigns the whole data points based on this new centroid then this step is repeated until the position of the centroid doesn't change.



7. Kmeans Clustering

```
df.columns
```

[10] Python

```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

```
X = df[['Annual Income (k$)','Spending Score (1-100)']]
```

[31] Python

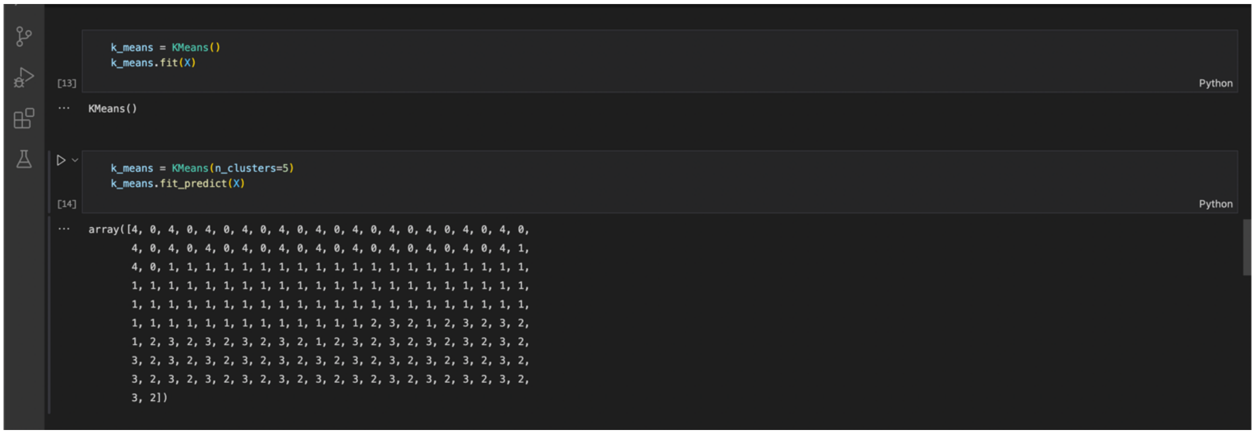
```
X.head()
```

[33] Python

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

```
from sklearn.cluster import KMeans
```

[1] ✓ 1.2s Python



```
k_means = KMeans()
k_means.fit(X)
```

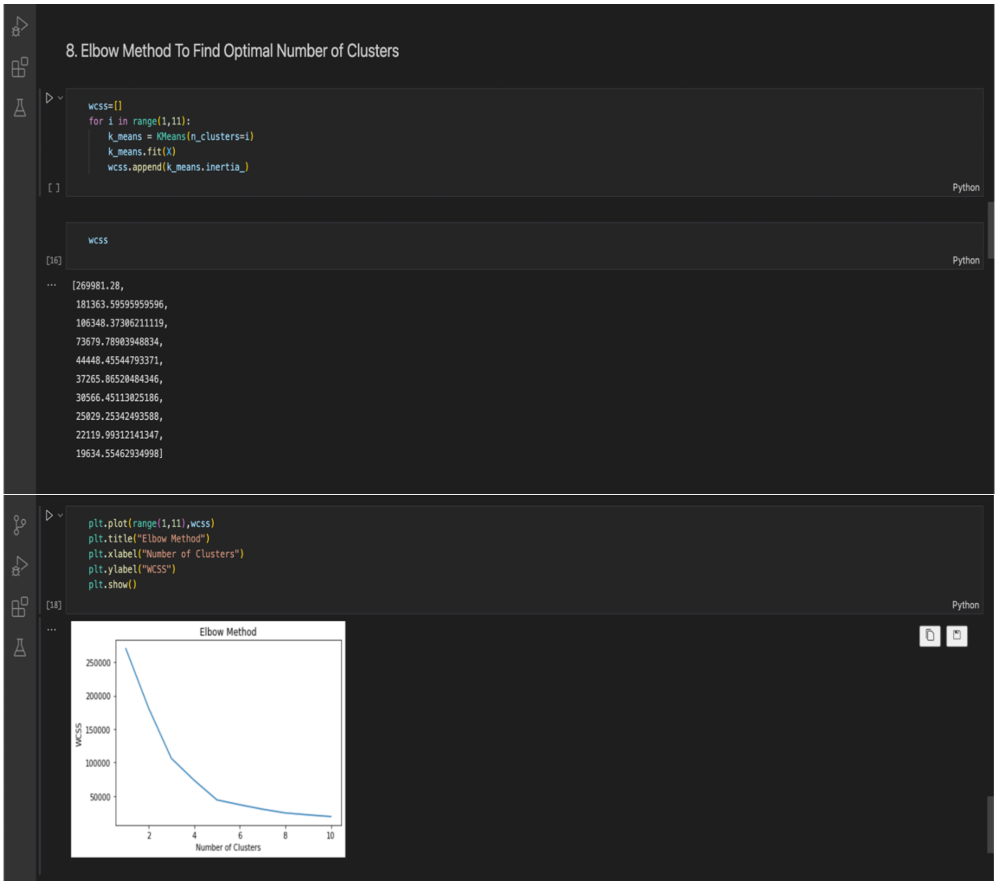
[13] Python

```
KMeans()
```

```
k_means = KMeans(n_clusters=5)
k_means.fit_predict(X)
```

[14] Python

```
array([4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0,
       4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 0, 4, 1,
       4, 0, 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, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 3, 2, 1, 2, 3, 2, 3, 2,
       1, 2, 3, 2, 3, 2, 3, 2, 3, 2, 1, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2,
       3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2,
       3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2,
       3, 2])
```



8. Elbow Method To Find Optimal Number of Clusters

```
wcss=[]
for i in range(1,11):
    k_means = KMeans(n_clusters=i)
    k_means.fit(X)
    wcss.append(k_means.inertia_)
```

[ ]

```
wcss
```

[16]

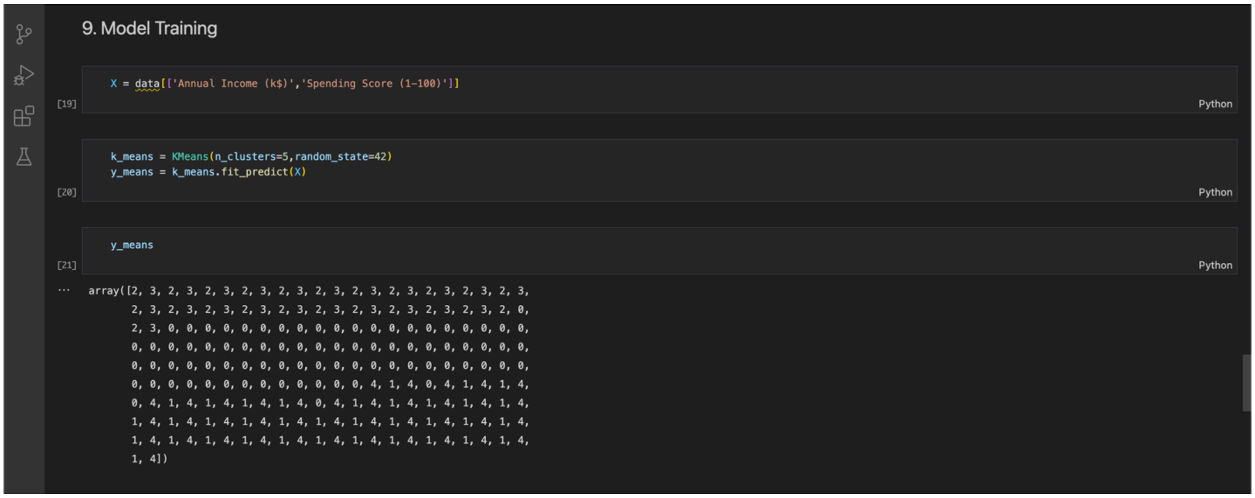
```
[269981.28,
 181363.59595959596,
 106348.37306211119,
 73679.78903948834,
 44448.45544793371,
 37265.86520484346,
 30566.45113025186,
 25029.25342493588,
 22119.99312141347,
 19634.55462934998]
```

```
plt.plot(range(1,11),wcss)
plt.title("Elbow Method")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
plt.show()
```

[18]

S9: Elbow method used in K-means clustering to find the optimal number of clusters

Segmentation using Annual Income and Spending Score.
The elbow method works by 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 the WCSS with the K value is plotted, 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.



9. Model Training

```
X = data[['Annual Income (k$)','Spending Score (1-100)']]
```

[19]

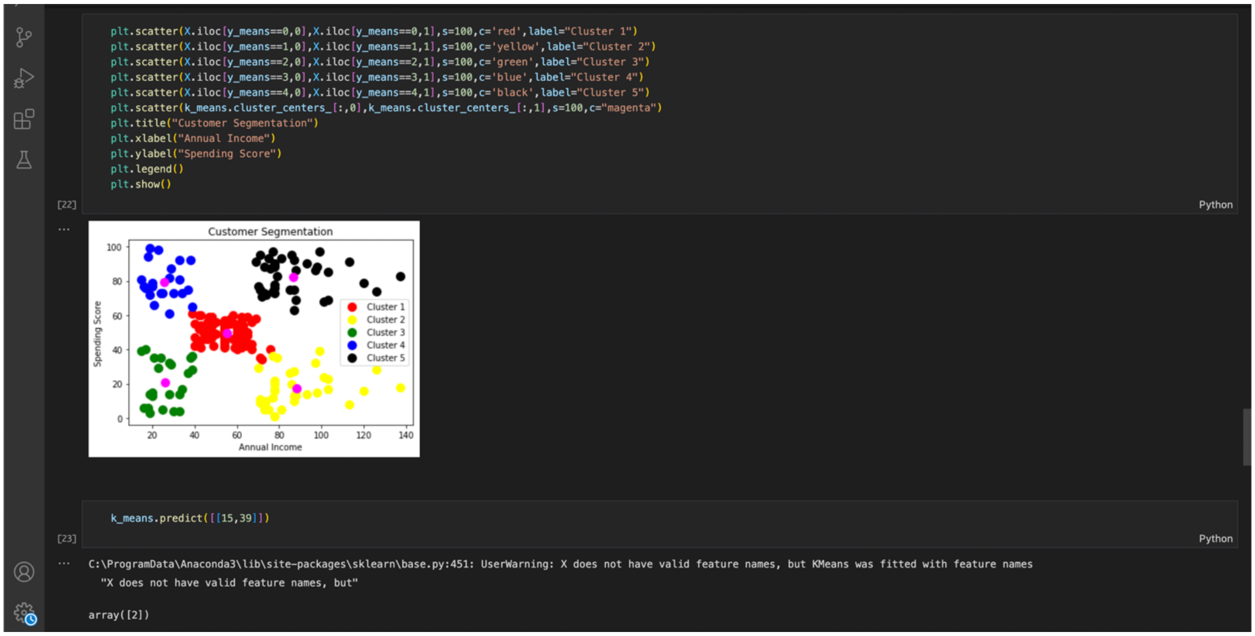
```
k_means = KMeans(n_clusters=5,random_state=42)
y_means = k_means.fit_predict(X)
```

[20]

```
y_means
```

[21]

```
array([2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3,
       2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 3, 2, 0,
       2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 1, 4, 0, 4, 1, 4, 1, 4,
       0, 4, 1, 4, 1, 4, 1, 4, 1, 4, 0, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4,
       1, 4])
```



```
plt.scatter(X.iloc[y_means==0,0],X.iloc[y_means==0,1],s=100,c='red',label="Cluster 1")
plt.scatter(X.iloc[y_means==1,0],X.iloc[y_means==1,1],s=100,c='yellow',label="Cluster 2")
plt.scatter(X.iloc[y_means==2,0],X.iloc[y_means==2,1],s=100,c='green',label="Cluster 3")
plt.scatter(X.iloc[y_means==3,0],X.iloc[y_means==3,1],s=100,c='blue',label="Cluster 4")
plt.scatter(X.iloc[y_means==4,0],X.iloc[y_means==4,1],s=100,c='black',label="Cluster 5")
plt.scatter(k_means.cluster_centers_[:,0],k_means.cluster_centers_[:,1],s=100,c="magenta")
plt.title("Customer Segmentation")
plt.xlabel("Annual Income")
plt.ylabel("Spending Score")
plt.legend()
plt.show()
```

[22]

```
k_means.predict([[15,39]])
```

[23]

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:451: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  "X does not have valid feature names, but"

array([2])
```

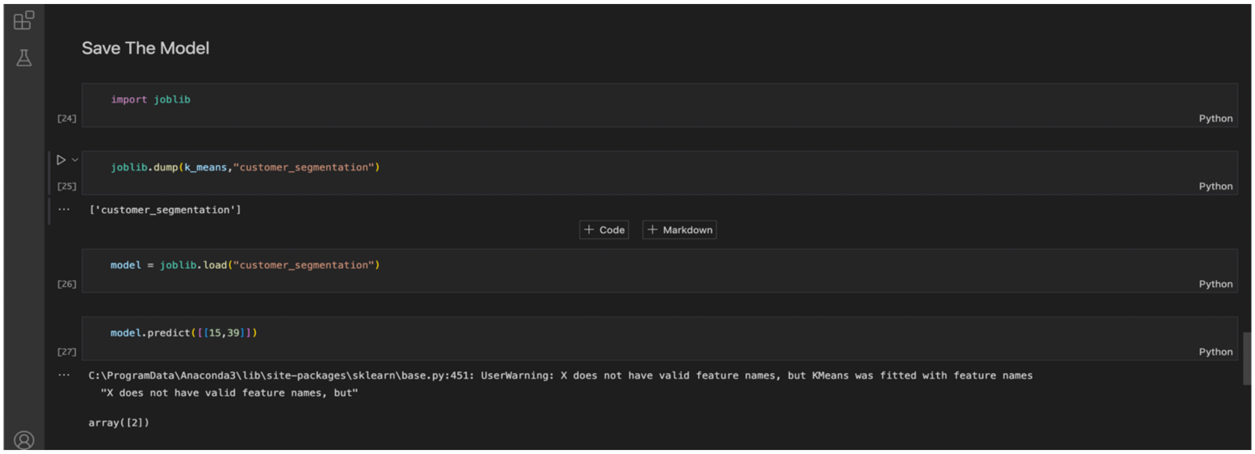
I selected the number of clusters for the dataset (K) , K number of centroids. In this case, I will use k = 5 which is the point that the error starts to decrease.

Observation

From the figure above, I identified 5 major groups:

- Low income roamers(green): earn less than 40k annually and shops occasionally.
- Low income fans(blue): earn less than 40k annually and shops regularly.
- High income roamers(yellow): earn more than 60k annually and shops occasionally.
- High income fans(black): earn less than 60k annually and shops regularly.
- Moderate income supporters(red): earn between 40k-80k annually and shops often

Launch, monitor, and maintain your system.



Save The Model

```
import joblib
```

[24]

```
joblib.dump(k_means,"customer_segmentation")
```

[25]

```
['customer_segmentation']
```

```
model = joblib.load("customer_segmentation")
```

[26]

```
model.predict([[15,39]])
```

[27]

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py:451: UserWarning: X does not have valid feature names, but KMeans was fitted with feature names
  "X does not have valid feature names, but"

array([2])
```

Marketing Strategy

It will be easier to turn our supporters into fans than our roamers even though we would still want to keep our fans very happy. Here are some recommendations for the marketing team:

- Future promotions should focus on customers who earn 40k-80k annually.
- Majority of the customers in the fans category are below 40 years. There for future promotions like discounts and coupons should focus on products for teenagers and young adults.
- From our analysis, I also observed that female customers tend to spend more money so more consideration/priority should be given to them in the marketing strategy.

Conclusion

- K-Means clustering was successful in grouping our customers into clusters which enabled us make recommendations to the marketing team
- Future work might include using other models like DBSCAN, Hierarchical clustering, and fine tuning hyper-parameters especially if we have increased dimensions. Hyperparameter tuning is important because it allows us find the best version of tbe model.