CUSTOMER SEGMENTATION

Phase 5

I. Introduction

To make predictions and find the clusters of potential customers of the mall and thus find appropriate measures to increase the revenue of the mall is one of the prevailing applications of unsupervised learning.

For example, a group of customers have high income but their spending score (amount spent in the mall) is low so from the analysis we can convert such type of customers into potential customers (whose spending score is high) by using strategies like better advertising, accepting feedback and improving the quality of products.

To identify such customers, this project analyses and forms clusters based on different criteria which are discussed in the further sections.

II. Dataset

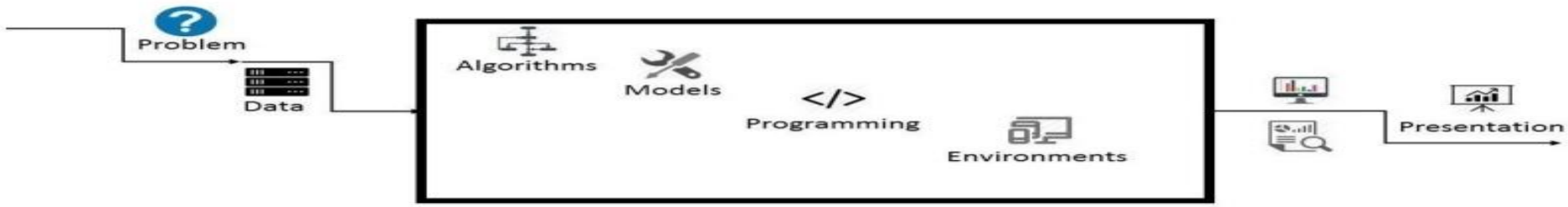
The dataset name is 'Mall_Customers.csv' consists of 5 columns which are CustomerID, Gender, Age, Annual Income (k$), Spending Score (1-100) where Gender is a categorical value and rest all features are numeric.

	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

2.1 Snapshot of Dataset

The size of the dataset is (200, 5) which is 200 rows and 5 columns.

III. Proposed Method and Architecture



3.1 Data Science Project Architecture

Problem Statement

Customer Segmentation is a popular application of unsupervised learning. Using clustering, identify segments of customers to target the potential user base. They divide customers into groups according to common characteristics like gender, age, interests, and spending habits so they can market to each group effectively.
Use K-means clustering and also visualize the gender and age distributions. Then analyze their annual incomes and spending scores.

Data

The size of the dataset is (200, 5) which is 200 rows and 5 columns. Also the dataset does not contain any NULL or NaN values.

Algorithms

K-means algorithm is used in this project to analyze and form clusters of customers based on their income and spending score features.

Model

K-means model is used and is hyper tuned parameters like *n_clusters=5* using elbow method to find the optimal number of clusters also *init='k-means++'* to avoid random initialization traps.

Programming and Environment

Programming Language: Python 3.6
Environment (Libraries and Technologies): Numpy, Pandas, Matplotlib, Seaborn, Jupyter Notebook, Google Colab.

IV. Methodology

The Data Science Methodology aims to answer basic questions in a prescribed sequence, that cover the five main aspects of data science projects. These aspects are:

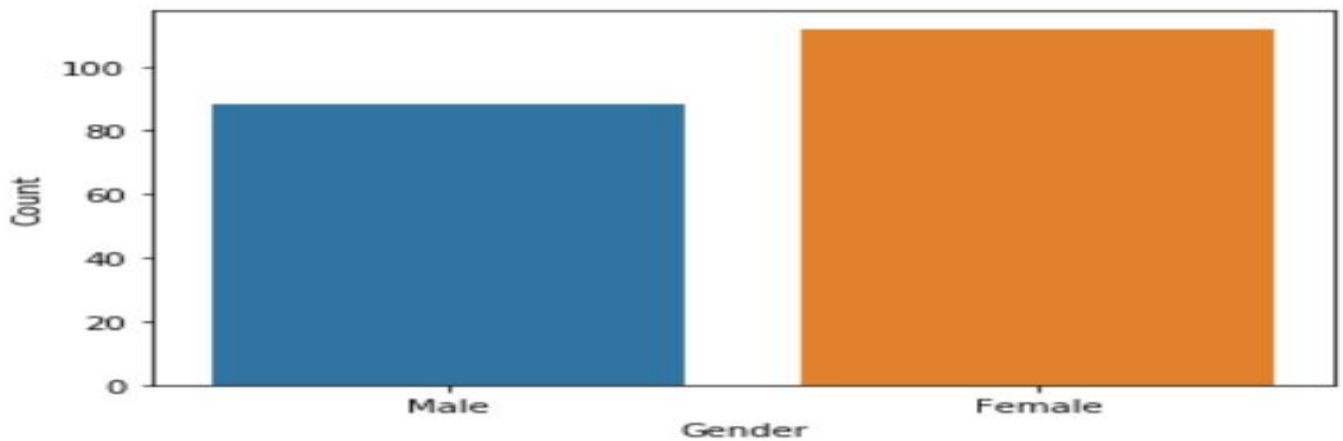
- From Problem to Approach
- From Requirements to Collection
- From Understanding to Preparation
- From Modelling to Evaluation
- From Deployment to Feedback

In this project, the prescribed sequence is:

- Creating an approach to solve the given problem statement
- Exploring the dataset and obtaining useful insight from the same
- Cleaning the dataset by handling nan values, remove duplicate records, etc.
- Data Visualization used to obtain important information from the data
- Data Preprocessing is performed to make the data ready to fit the model this includes feature scaling, splitting the dataset into features and labels, etc.
- Model Building

V. Implementation and Analysis

On performing data visualization on the dataset, a lot of insights were obtained such as:



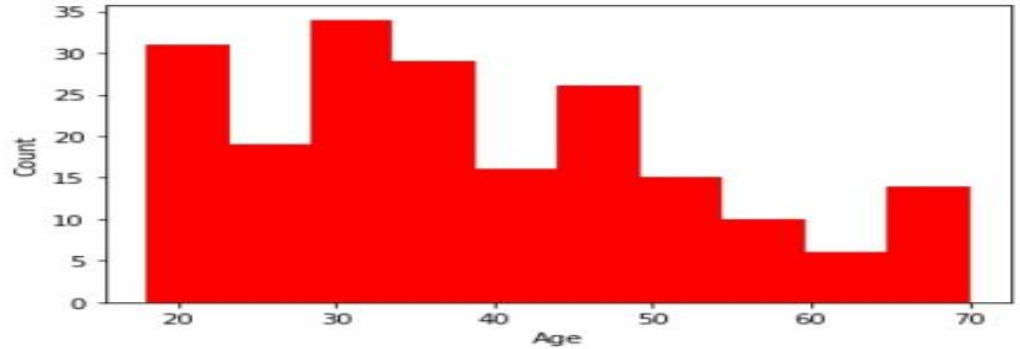
5.1 Gender Plot

Gender Plot Analysis

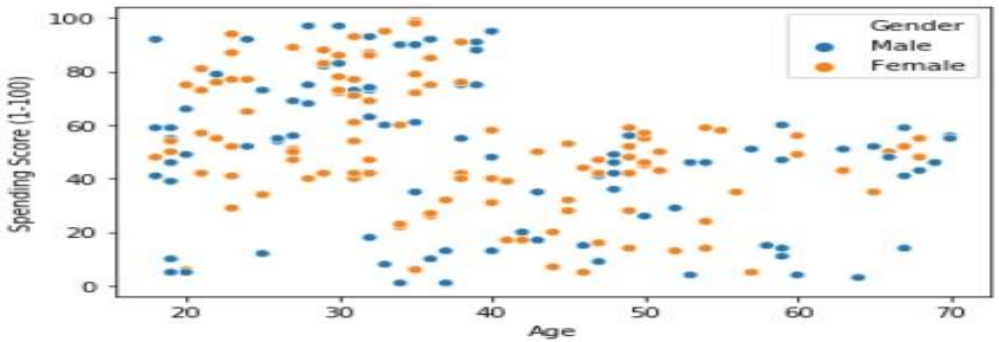
From the Count plot, it is observed that the number of Female customers is more than the total number of Male customers.

Age Plot Analysis

From the Histogram it is evident that there are 3 age groups that are more frequently shop at the mall, they are: 15-22 years, 30-40 years, and 45-50 years.



5.2 Age Plot



5.3 Age Vs Spending Score

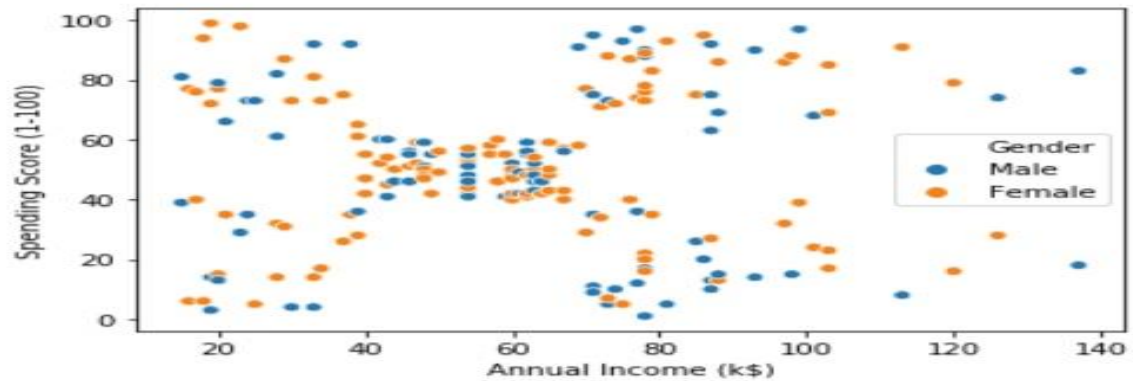
Age Vs Spending Score Analysis

1. From the Age Vs Spending Score plot we observe that customers whose spending score is more than 65 have their Age in the range of 15-42 years. Also from the Scatter plot it is observed that customers whose spending score is more than 65 consists of more Females than Males.
2. The customers having average spending score ie: in the range of 40-60 consists of the age group of the range 15-75 years and the count of males and females in this age group is also approximately the same.

Annual Income Vs Spending Score Analysis

We observe that there are 5 clusters and can be categorized as:

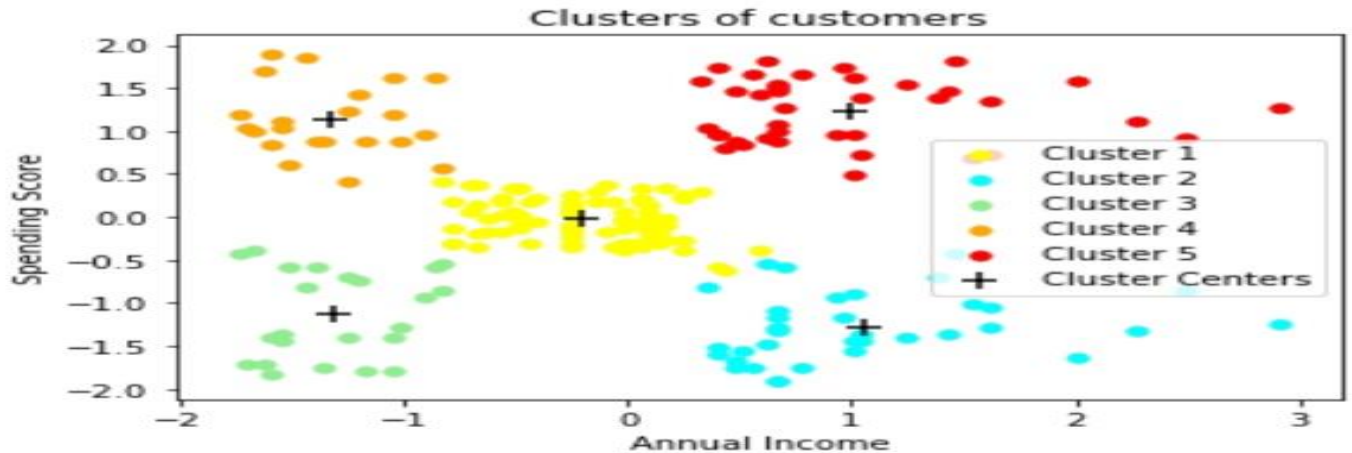
a. High Income, High Spending Score (Top Right Cluster)
b. High Income, Low Spending Score (Bottom Right Cluster)
c. Average Income, Average Spending Score (Center Cluster)
d. Low Income, High Spending Score (Top Left Cluster)
e. Low Income, Low Spending Score (Bottom Left Cluster)



5.4 Annual Income Vs Spending Score

VI. Conclusion

For this project, the K-means algorithm is used and performs the best (with n_clusters = 5 and init = 'kmeans++'). After the clustering algorithm is applied to the dataset, this is the output.



6.1 Annual Income Vs Spending Score after Clustering

Clustering Analysis

a. High Income, High Spending Score (Cluster 5) - Target these customers by sending new product alerts which would lead to an increase in the revenue collected by the mall as they are loyal customers.

b. High Income, Low Spending Score (Cluster 2) - Target these customers by asking the feedback and advertising the product in a better way to convert them into Cluster 5 customers.

c. Average Income, Average Spending Score (Cluster 1) - May or may not target these groups of customers based on the policy of the mall.

d. Low Income, High Spending Score (Cluster 4) - Can target these set of customers by providing them with Low-cost EMI's, etc.

e. Low Income, Low Spending Score (Cluster 3) - Don't target these customers since they have less income and need to save money.

Importing essential libraries

```
#importing essential libraries
Import numpy as np
import pandas as pd
# Loading the dataset
df = pd.read_csv(“Mall_Customers.csv”)
```

Exploring the dataset

- # Returns number of rows and columns of the dataset
- df.shape
- (200, 5)
- # Returns an object with all of the column headers
- df.columns
- Index([‘CustomerID’, ‘Gender’, ‘Age’, ‘Annual Income (k$)’
- ‘Spending Score (1-100)’],
- dtype=‘object’)
- # Returns different datatypes for each columns (float, int, string, bool, etc.)
- df.dtypes
- CustomerID int64
- Gender object
- Age int64
- Annual Income (k$) int64
- Spending Score (1-100) int64
- type: object

- # Returns the first x number of rows when head(x). Without a number it returns 5
- 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

- # Returns the last x number of rows when tail(x). Without a number it returns 5
- df.tail()

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
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

- # Returns basic information on all columns
- df.info()
- <class 'pandas.core.frame.DataFrame'>
- RangeIndex: 200 entries, 0 to 199
- Data columns (total 5 columns):
- CustomerID 200 non-null int64
- Gender 200 non-null object
- Age 200 non-null int64
- Annual Income (k$) 200 non-null int64
- Spending Score (1-100) 200 non-null int64
- dtypes: int64(4), object(1)
- memory usage: 7.9+ KB
- # Returns basic statistics on numeric columns
- df.describe().T
- count mean std min 25% 50% 75% max
- CustomerID 200.0 100.50 57.879185 1.0 50.75 100.5 150.25 200.0
- Age 200.0 38.85 13.969007 18.0 28.75 36.0 49.00 70.0
- Annual Income (k$) 200.0 60.56 26.264721 15.0 41.50 61.5 78.00 137.0
- Spending Score (1-100) 200.0 50.20 25.823522 1.0 34.75 50.0 73.00 99.0

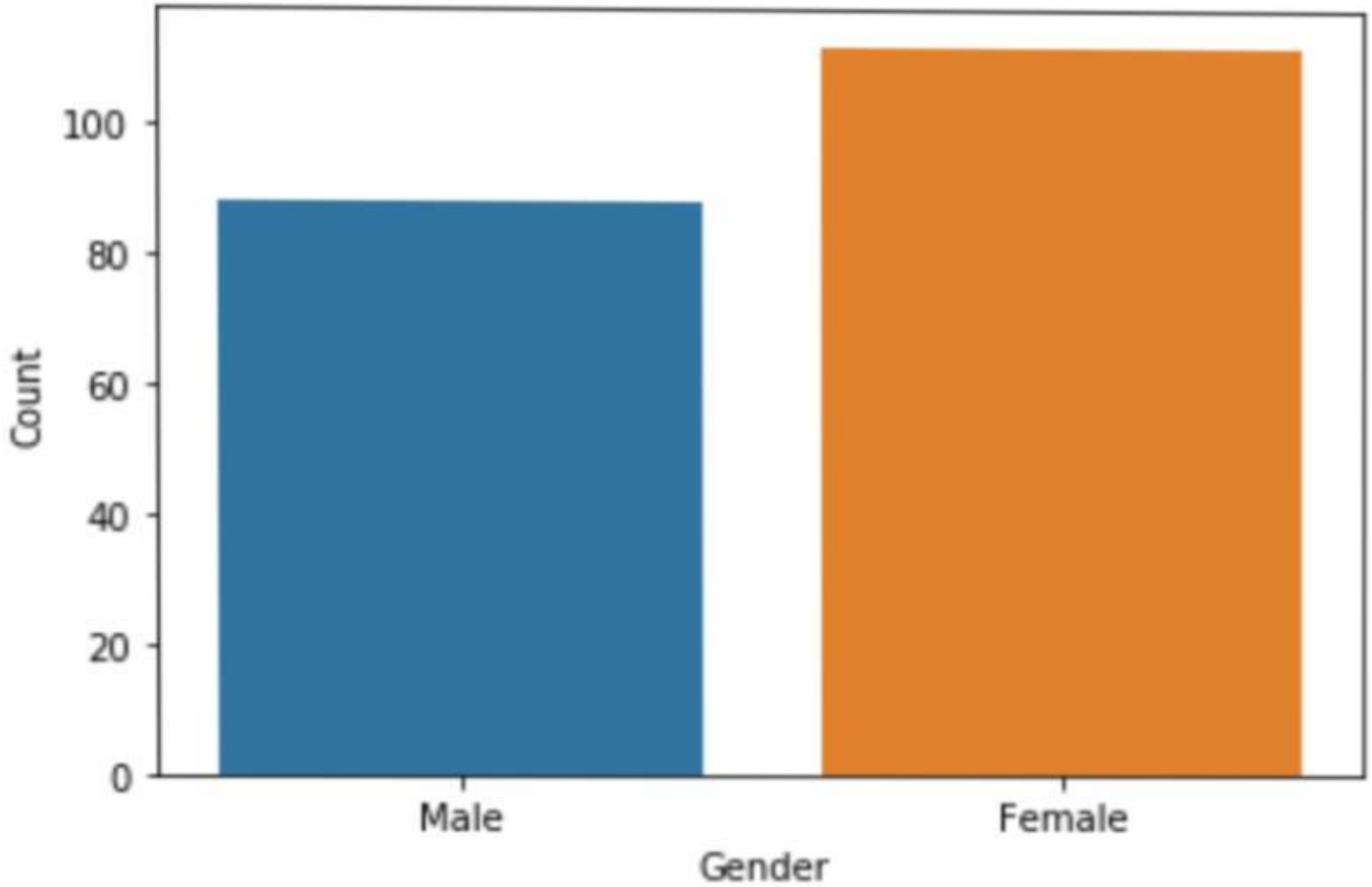
- # Returns true for a column having null values, else false
- df.isnull().any()
- CustomerID False
- Gender False
- Age False
- Annual Income (k$) False
- Spending Score (1-100) False
- dtype: bool

Data Cleaning

- Creating the copy of dataset
- df_copy = df.copy(deep=True)
- df_copy.head(3)
- CustomerID Gender Age Annual Income (k$) Spending Score (1-100)
- 0 1 Male 19 15 39
- 1 2 Male 21 15 81
- 2 3 Female 20 16 6
- # Dropping the column of 'CustomerID' as it does not provide any value
- df_copy.drop('CustomerID', axis=1, inplace=True)
- df_copy.columns
- Index(['Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)'], dtype='object')

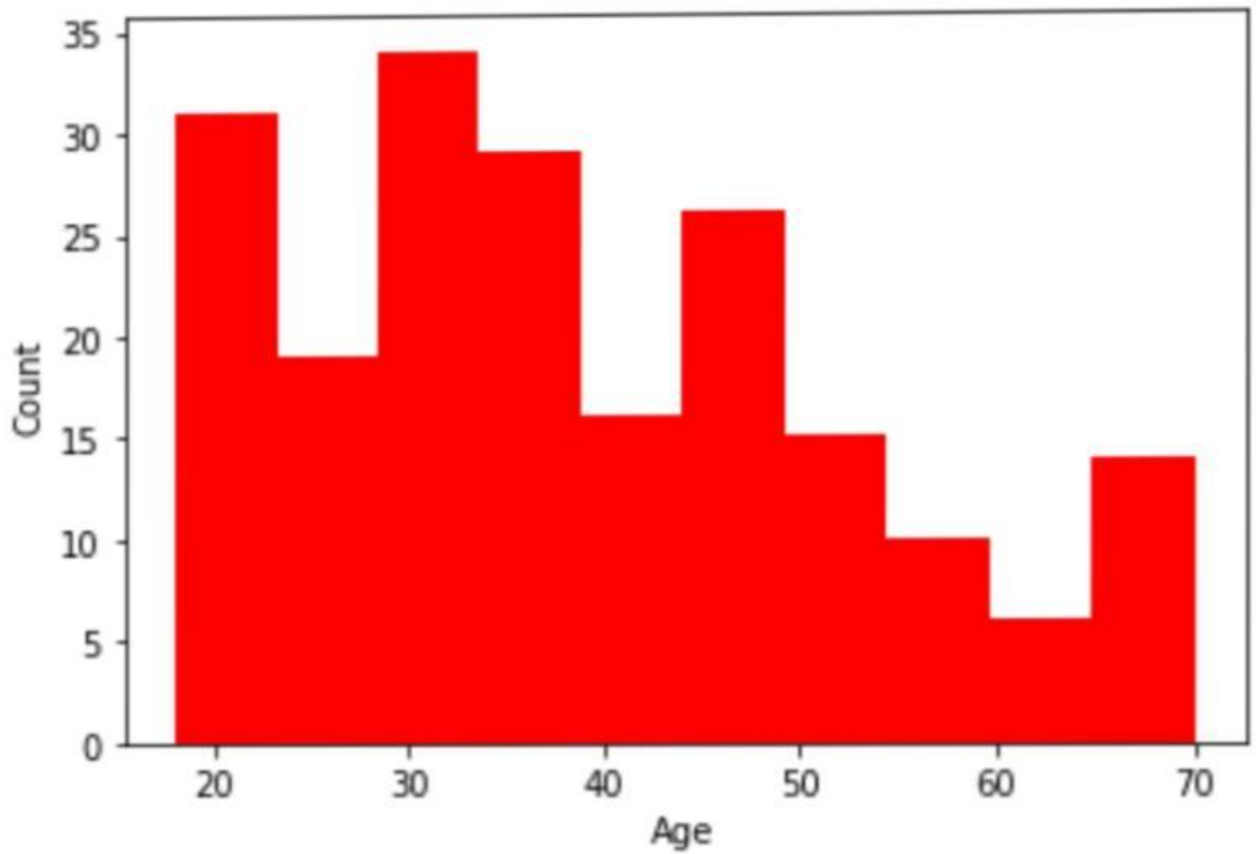
Data Visualization

- # Loading essential libraries
- import matplotlib.pyplot as plt
- import seaborn as sns
- df_copy.columns
- Index([‘Gender’, ‘Age’, ‘Annual Income (k$)’, ‘Spending Score (1-100)’], dtype=‘object’)
- Gender Plot
- # Visualising the columns ‘Gender’ using Countplot
- sns.countplot(x=‘Gender’, data=df_copy)
- plt.xlabel(‘Gender’)
- plt.ylabel(‘Count’)
- Text(0, 0.5, ‘Count’)


100
80
60
40
20
0
Count
Male
Female
Gender

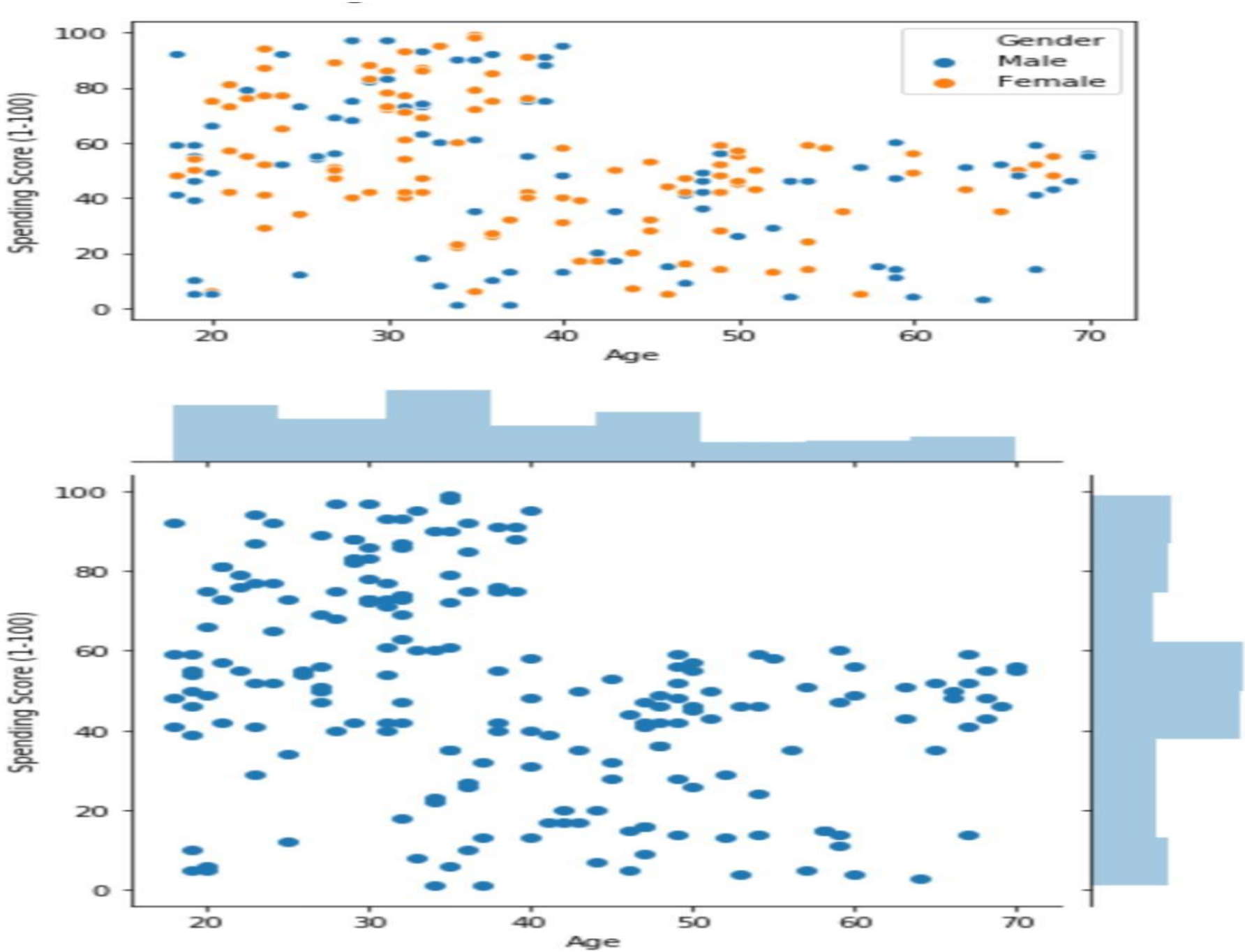
Age Plot

- # Visualising the columns ‘Age’ using Histogram
- plt.hist(x=df_copy[‘Age’], bins=10, orientation=‘vertical’, color=‘red’)
- plt.xlabel(‘Age’)
- plt.ylabel(‘Count’)
- plt.show()


35
30
25
20
15
10
5
0
Count
20
30
40
50
60
70
Age

Age Vs Spending Score

- # Visualising the columns ‘Age’, ‘Spending Score (1-100)’ using Scatterplot and Jointplot
- sns.scatterplot(data=df_copy, x=‘Age’, y=‘Spending Score (1-100)’, hue=‘Gender’)
- sns.jointplot(data=df_copy, x=‘Age’, y=‘Spending Score (1-100)’)
- <seaborn.axisgrid.JointGrid at 0x115a0f8c2c8>


Gender
Male
Female
Spending Score (1-100)
Age
0
20
40
60
80
100
20
30
40
50
60
70

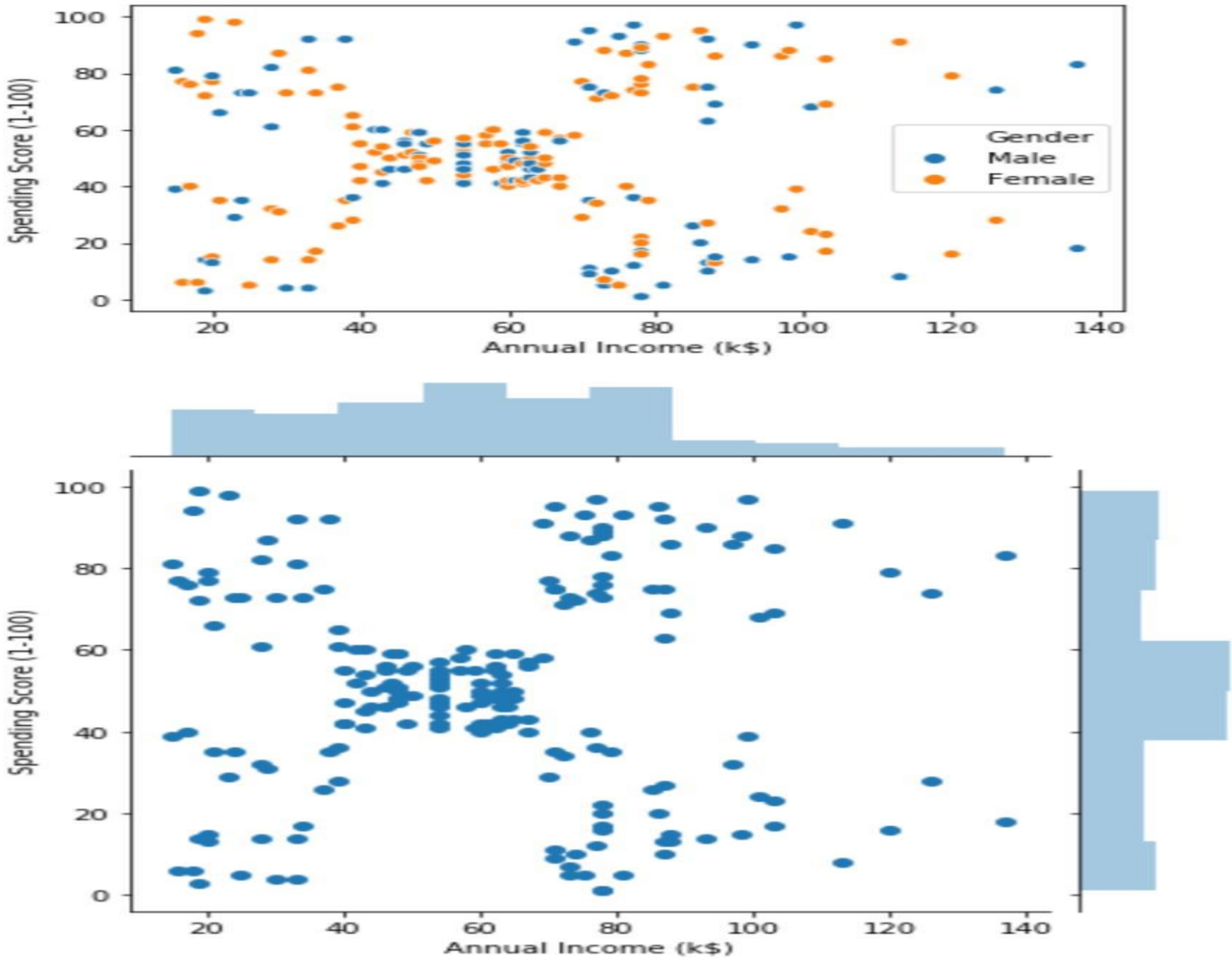
- Gender plot – Observation
-
- From the Count plot it is observed that the number of Female customers are more that the total number of Male customers
- Age plot – Observation
-
- From the Histogram it is evident that there are 3 age groups that are more frequently shop at the mall, they are: 15-22 years, 30-40 years and 45-50 years.

- Age Vs Spending Score – Observation
-
- 1. From the Age Vs Spending Score plot we observe that customers whose spending score is more than 65 have their Age in the range of 15-42 years. Also from the Scatter plot it is observed that customers whose spending score is more than 65 consists of more Females than Males.
-
- 2. Also, the customers having average spending score ie: in the range of 40-60 consists of age group of the range 15-75 years and the count of Male and Female in this age group is also approximatly the same.

- Age Vs Spending Score – Observation
-
- 1. From the Age Vs Spending Score plot we observe that customers whose spending score is more than 65 have their Age in the range of 15-42 years. Also from the Scatter plot it is observed that customers whose spending score is more than 65 consists of more Females than Males.
-
- 2. Also, the customers having average spending score ie: in the range of 40-60 consists of age group of the range 15-75 years and the count of Male and Female in this age group is also approximatly the same

Annual Income Vs Spending Score

- #Visualising the columns ‘Annual Income (k$)’, ‘Spending Score (1-100)’ using Scatterplot and Jointplot
- sns.scatterplot(data=df_copy, x=‘Annual Income (k$)’, y=‘Spending Score (1-100)’, hue=‘Gender’)
- sns.jointplot(data=df_copy, x=‘Annual Income (k$)’, y=‘Spending Score (1-100)’)


Spending Score (1-100)
Annual Income (k$)
Gender
Male
Female
0
20
40
60
80
100
120
140

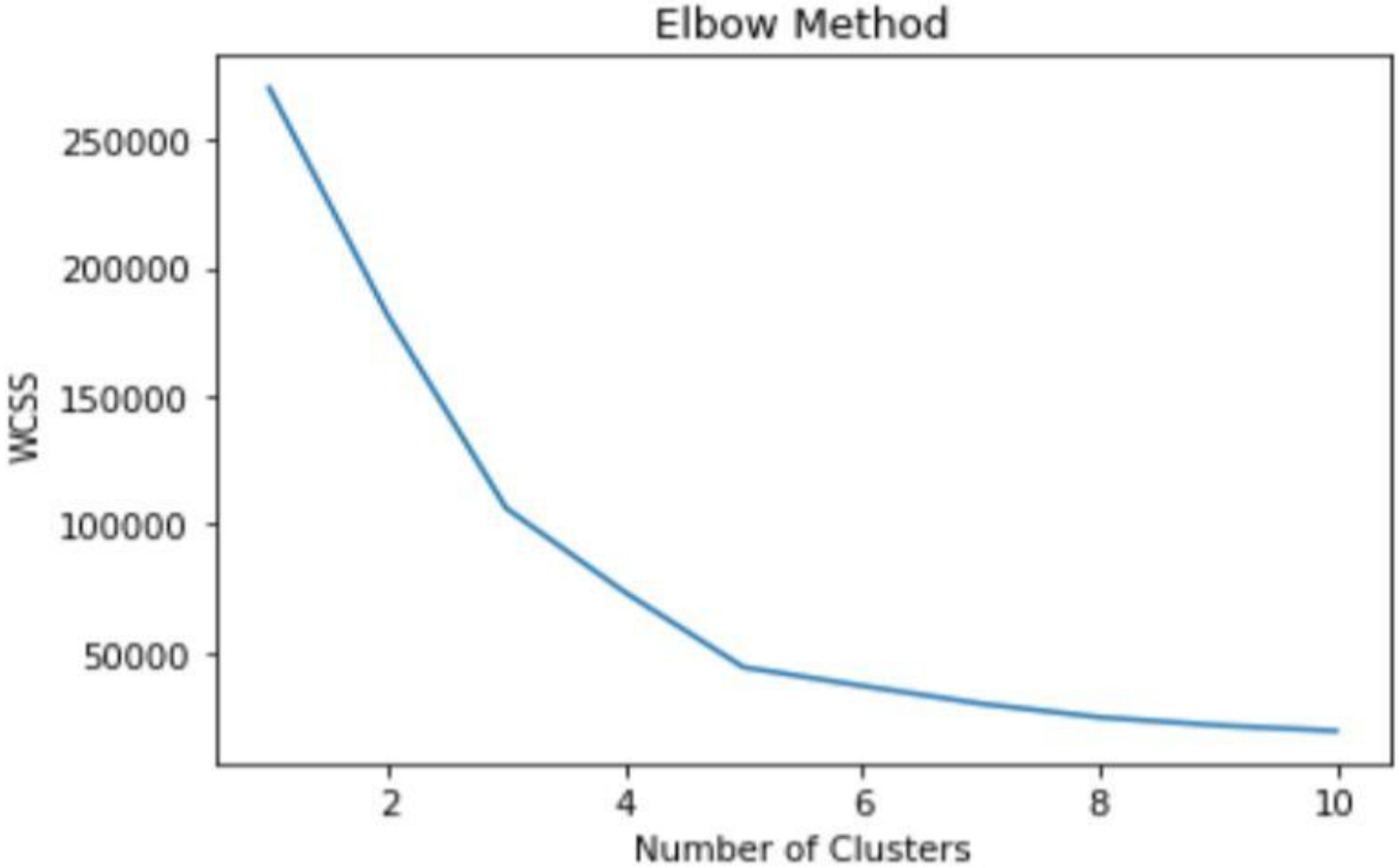
- Annual Income Vs Spending Score – Observation
-
- From the Annual Income Vs Spending Score plot we observe that there are 5 clusters and can be categorised as:
- A. High Income, High Spending Score (Top Right Cluster)
-
- b. High Income, Low Spending Score (Bottom Right Cluster)
-
- c. Average Income, Average Spending Score (Center Cluster)
-
- d. Low Income, High Spending Score (Top Left Cluster)
-
- e. Low Income, Low Spending Score (Bottom Left Cluster)

Data Preprocessing

- # Selecting ‘Annual Income’ and ‘Spending Score’ as the features for clustering
- X = df_copy.iloc[:, [2,3]]
- X.columns
- Index([‘Annual Income (k$)’, ‘Spending Score (1-100)’], dtype=‘object’)

Finding optimal number of clusters using Elbow Method

- # Calculating WCSS values for 1 to 10 clusters
- from sklearn.cluster import Kmeans
- wcss = []
- for i in range(1,11):
- kmeans_model = Kmeans(n_clusters=i, init='k-means++', random_state=42)
- kmeans_model.fit(X)
- wcss.append(kmeans_model.inertia_)
- # Plotting the WCSS values
- plt.plot(range(1,11), wcss)
- plt.title('Elbow Method')
- plt.xlabel('Number of Clusters')
- plt.ylabel('WCSS')
- plt.show()



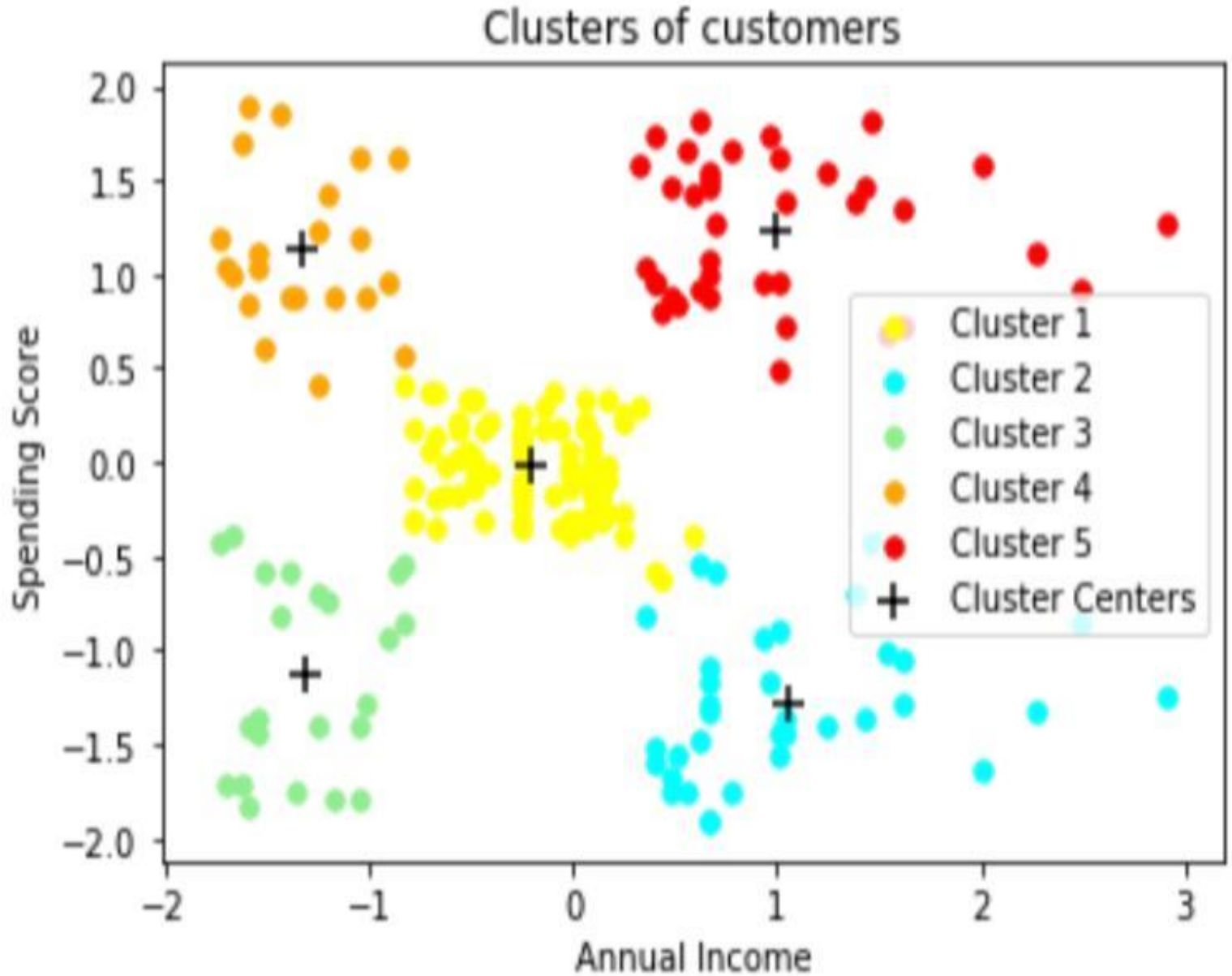
*From the above plot it is observed that 5 clusters** are optimal for the given dataset.*

Feature Scaling

- From sklearn.preprocessing import StandardScaler
- scaler = StandardScaler()
- X = scaler.fit_transform(X)
- Feature Scaling is performed because Kmeans uses Distance (Euclidean, Manhattan, etc.) and the model perfoms faster on scaling the values

Model Building

- # Training the Kmeans model with n_clusters=5
- kmeans_model = Kmeans(n_clusters=5, init='k-means++', random_state=42)
- y_kmeans = kmeans_model.fit_predict(X)
- # Visualising the clusters
- plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 30, c = 'yellow', label = 'Cluster 1')
- plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 30, c = 'cyan', label = 'Cluster 2')
- plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 30, c = 'lightgreen', label = 'Cluster 3')
- plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 30, c = 'orange', label = 'Cluster 4')
- plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 30, c = 'red', label = 'Cluster 5')
- plt.scatter(x=kmeans_model.cluster_centers_[:, 0], y=kmeans_model.cluster_centers_[:, 1], s=100, c='black', marker='+', label='Cluster Centers')
- plt.legend()
- plt.title('Clusters of customers')
- plt.xlabel('Annual Income')
- plt.ylabel('Spending Score')
- plt.show()


Clusters of customers
Spending Score
Annual Income
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2
3
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster Centers

- Clustering – Observation
-
- a. High Income, High Spending Score (Cluster 5) – Target these customers by sending new product alerts which would lead to increase in the revenue collected by the mall as they are loyal customers.
-
- B. High Income, Low Spending Score (Cluster 3) – Target these customers by asking the feedback and advertising the product in a better way to convert them into Cluster 5 customers.
-
- C. Average Income, Average Spending Score (Cluster 2) – Can target these set of customers by providing them with Low cost EMI's etc.
-
- D. Low Income, High Spending Score (Cluster 1) – May or may not target these group of customers based on the policy of the mall.
-
- E. Low Income, Low Spending Score (Cluster 4) – Don't target these customers since they have less income and need to save money.

THANK YOU.