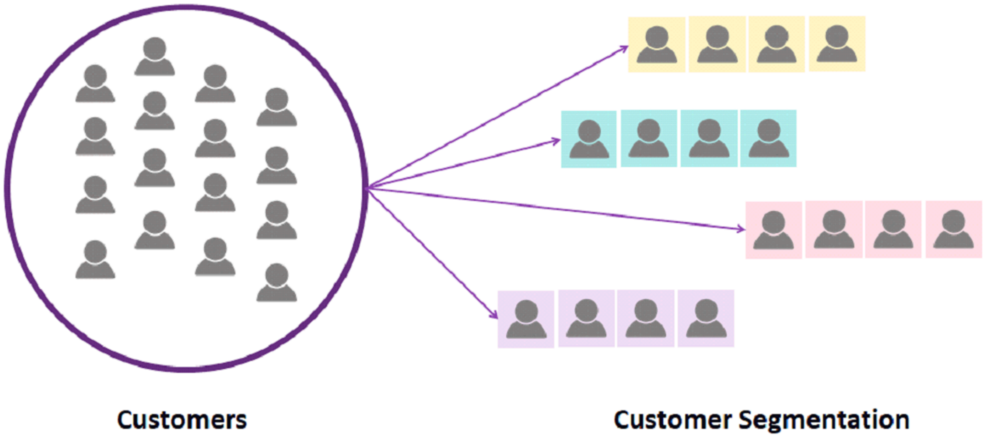
Customer Segmentation Using Machine Learning by **Farhan Ashraf**



Aim of the Project

Customer Segmentation is an unsupervised method of targeting the customers in order to increase sales and market goods in a better way.

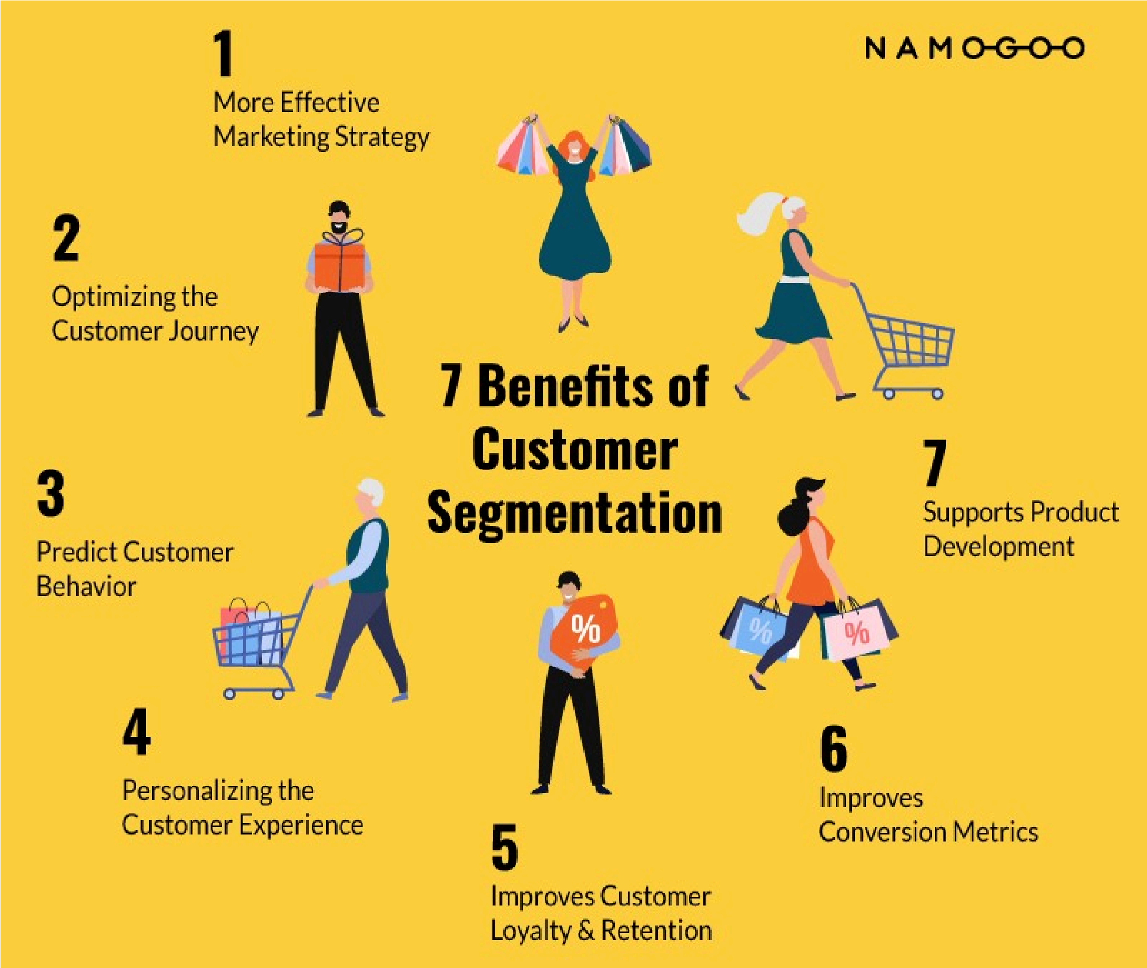
This project deals with real-time data where we have to segment the customers in the form of clusters using the K-Means algorithm.

The dataset consists of important variables like Age, Gender, Annual Income, etc.

With the help of the algorithms, we can easily visualize the data and can get a segmentation of each customer so that we can target the customers in a better way.

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. Using the above data, companies can then outperform the competition by developing uniquely appealing products and services. ervices.

Advantages of Customer Segmentation


NAMOGOO
7 Benefits of Customer Segmentation
1 More Effective Marketing Strategy
2 Optimizing the Customer Journey
3 Predict Customer Behavior
4 Personalizing the Customer Experience
5 Improves Customer Loyalty & Retention
6 Improves Conversion Metrics
7 Supports Product Development
%
%
%

Some More Examples:

1. **Set Appropriate Pricing:**
    - Understand customer segments to set prices reflecting perceived value, boosting profitability.
2. **Customize Marketing Campaigns:**
    - Tailor campaigns to resonate with segment needs, boosting engagement and conversion.
3. **Optimize Distribution Strategy:**
    - Identify preferred purchase channels per segment to optimize product availability.
4. **Deploy Specific Product Features:**
    - Develop features based on segment preferences to meet target needs effectively.
5. **Prioritize New Product Development:**
    - Identify gaps and needs within segments to guide new product development efforts.
6. **Enhance Customer Retention:**

- Tailor service and engagement strategies for improved satisfaction and loyalty.

7. **Allocate Resources Efficiently:**

   - Focus resources on profitable customer groups to optimize expenditures.

8. **Gain Competitive Edge:**

   - Address unique segment needs to differentiate from competitors.

9. **Understand Customer Behavior:**

   - Segmentehavior and preferences for informed decisions.

10. **Increase Market Share:**

    - Target segments effectively to expand customer base and market share.insights reveal deeper b

Requirement and techniques i have used:

Tools Used

- **Anaconda**: Provides Jupyter Notebook for interactive Python coding and package management.
- **Python Libraries**:
  - **scikit-learn**: For clustering (e.g., K-means), classification (e.g., decision trees), and other ML tasks.
  - **pandas**: Data manipulation and analysis.
  - **numpy**: Numerical operations and array handling.
  - **matplotlib** and **seaborn**: Data visualization.

Machine Learning Techniques

- **Clustering**: Utilize K-means or hierarchical clustering for grouping customers based on behavior or attributes.
- **Dimensionality Reduction**: Techniques like PCA for visualizing and reducing high-dimensional data.
- **Classification**: Implement algorithms such as logistic regression or decision trees for predicting customer behavior.

Best Practices

- **Data Preparation**: Clean, normalize, and format data before applying ML algorithms.
- **Model Evaluation**: Use metrics like accuracy, F1-score for classification, and inertia or silhouette score for clustering.
- **Cross-validation**: Implement k-fold cross-validation to assess model performance.
- **Documentation**: Use Jupyter Notebooks for documenting code, visualizations, and analysis steps to ensure reproducibility and collaboration.

Machine Learning

Machine Learning is an application of Artificial Intelligence (AI) which enables a program to learn from experiences and improve itself at a task without being explicitly programmed. For example, you can develop a program to identify fruits based on properties like color, shape, and siz

.

Unsupervised Learning

Unsupervised learning is a machine learning technique where models find patterns and insights from data without explicit supervision or labeled outcomes. It's akin to how humans learn new things without direct instructions.

K-Means Clustering

K-means clustering is a type of unsupervised learning used when data lacks predefined categories. Its goal is to group data points into clusters based on feature similarity, with the number of clusters (K) specified by i

Getting into the Project: Customer Segmentation Using Python in Machine Learningne Learning

Customer Segmentation is an unsupervised method to target customers effectively, aiming to increase sales and optimize marketion import numpy as np

Library Used

Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with structured (tabular, multidimensional, potentially heterogeneous) and time-series data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python.

pandas (software) It is imported as *import pandas as pd*

Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Matplotlib: Quick and pretty (enough) to get you started. It is imported as *import matplotlib.pyplot as plt*

Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

How to build beautiful plots with Python and Seaborn Seaborn helps you explore and understand your data.

It is imported as *import seaborn as sns*

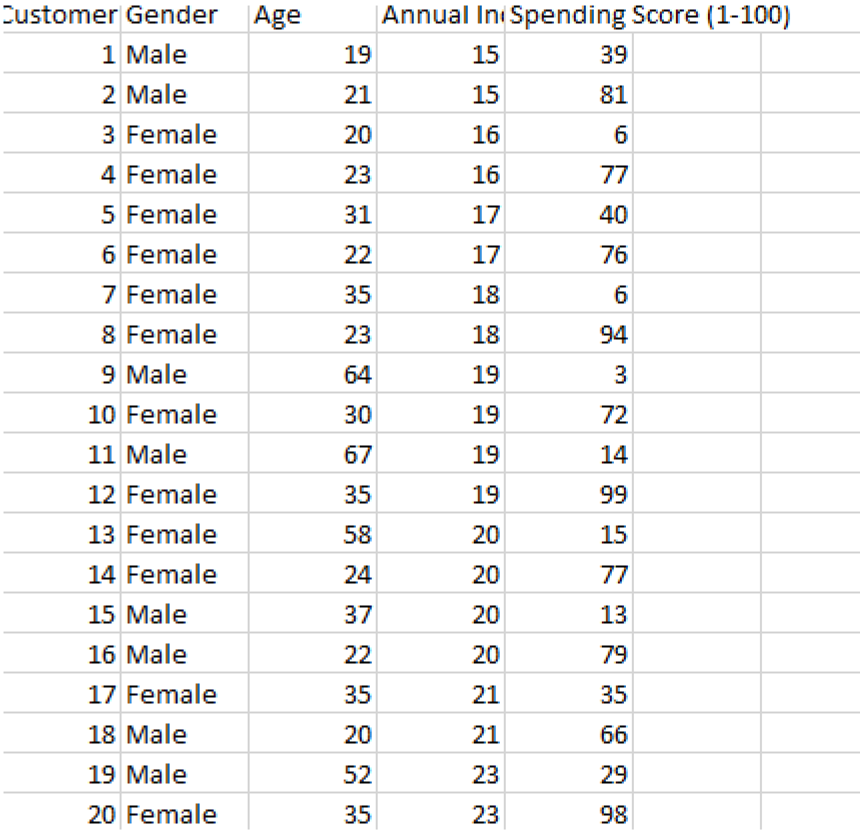
Analyzing and Knowing the Data Set

It is crucial to understand the contents of the dataset before proceeding with analysis. Checking for null values or undefined elements ensures the data is clean and ready for visualization or further processing.

The dataset comprises the following variables:

- **Customer**: Identifies each individual customer.
- **Age**: Represents the age of each customer.
- **Gender**: Indicates the gender of each customer.
- **Annual Income**: Shows the annual income of each customer.
- **Spending Scores**: Reflects the spending score assigned to each customer.

These variables provide essential information for understanding customer behavior and preferences. . s



Customer	Gender	Age	Annual In	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3
10	Female	30	19	72
11	Male	67	19	14
12	Female	35	19	99
13	Female	58	20	15
14	Female	24	20	77
15	Male	37	20	13
16	Male	22	20	79
17	Female	35	21	35
18	Male	20	21	66
19	Male	52	23	29
20	Female	35	23	98

Visualizing the Data

Data visualization is a powerful tool for understanding patterns and relationships within data. Effective visualizations can convey insights clearly, whereas poor visuals may obscure or misrepresent information.

In [1]:
```
#Required Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df = pd.read_csv("D:/Github projects/Mall_Customers.csv")
```

In [3]:
```
df.head()
```

Out[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
3	4	Female	23	16	77
4	5	Female	31	17	40

In [4]:

```
df.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
```

In [5]:

```
df.tail()
```

Out[5]:

	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

As the data set got loaded now we have to find out gender distribution between males and females. This helps to know the average number of male and female customers who visits the mall.

In [6]:

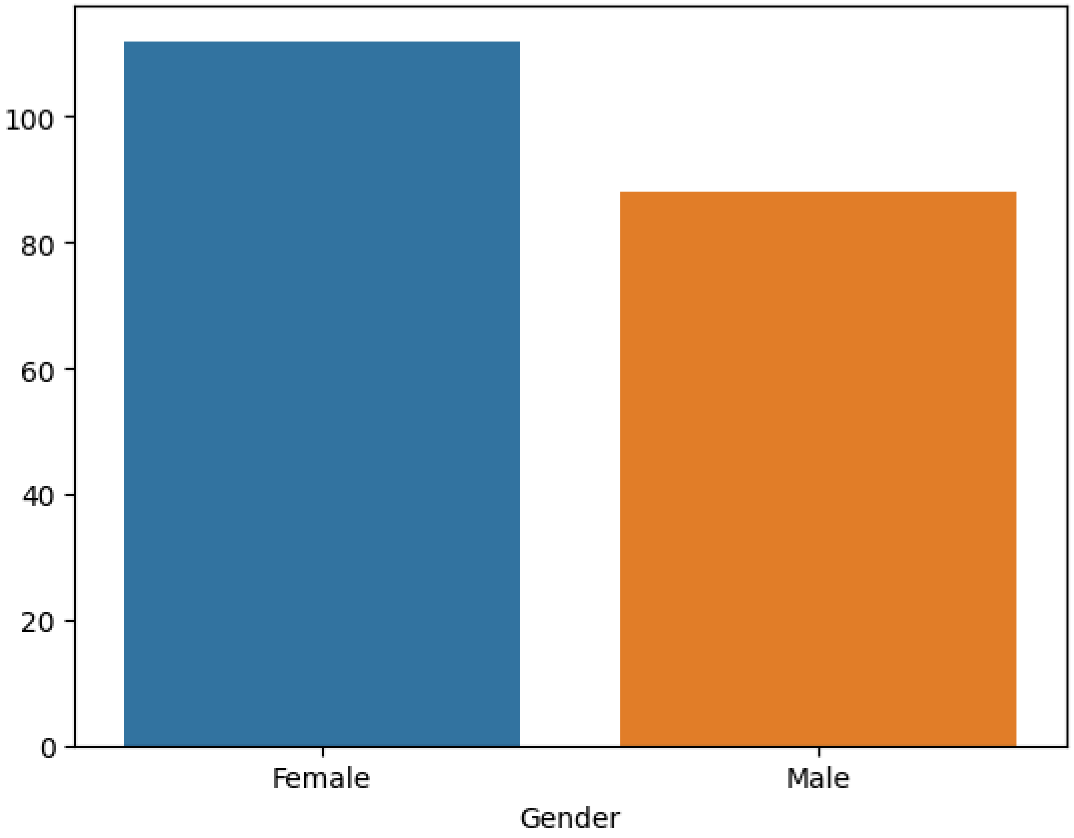
```
genders=df.Gender.value_counts()
```

In [7]:

```
sns.barplot(x=genders.index,y=genders.values)
```

Out[7]:

```
<Axes: xlabel='Gender'>
```



The data consist of various customers with their ages now I have visualized the different customers with age groups This technique helps to know which age group visits the mall on a frequent basis so that we can easily target the customers. The code for the above can be written by defining the different age groups from 18-26,27-35,36-45,46-55 and >55 and plotting it in the form of a bar graph.

In [8]:

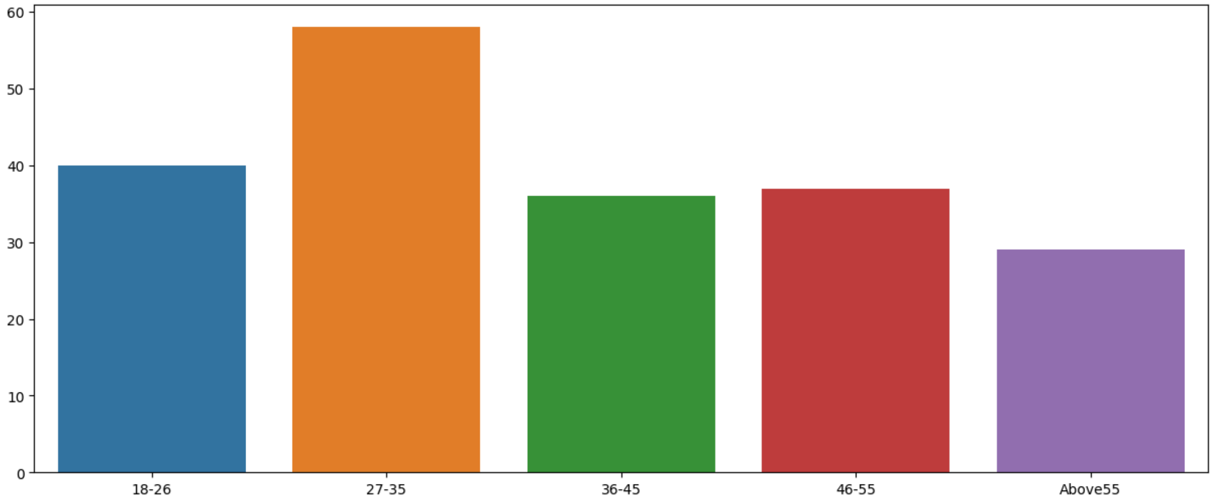
```
age18_26 = df.Age[(df.Age<=26)&(df.Age>=18)]
age27_35 = df.Age[(df.Age<=35)&(df.Age>=27)]
age36_45 = df.Age[(df.Age<=45)&(df.Age>=36)]
age46_55 = df.Age[(df.Age<=55)&(df.Age>=46)]
age55above = df.Age[(df.Age>=56)]
```

In [9]:

```
x=["18-26","27-35","36-45","46-55","Above55"]
y=[len(age18_26.values), len(age27_35.values),len(age36_45.values),len(age46_55.val
```

In [10]:

```
plt.figure(figsize=(15,6))
plt.title=("Number of customers and ages")
plt.xlabel=("Ages")
plt.ylabel=("Number of customers")
sns.barplot(x=x,y=y)
plt.show()
```



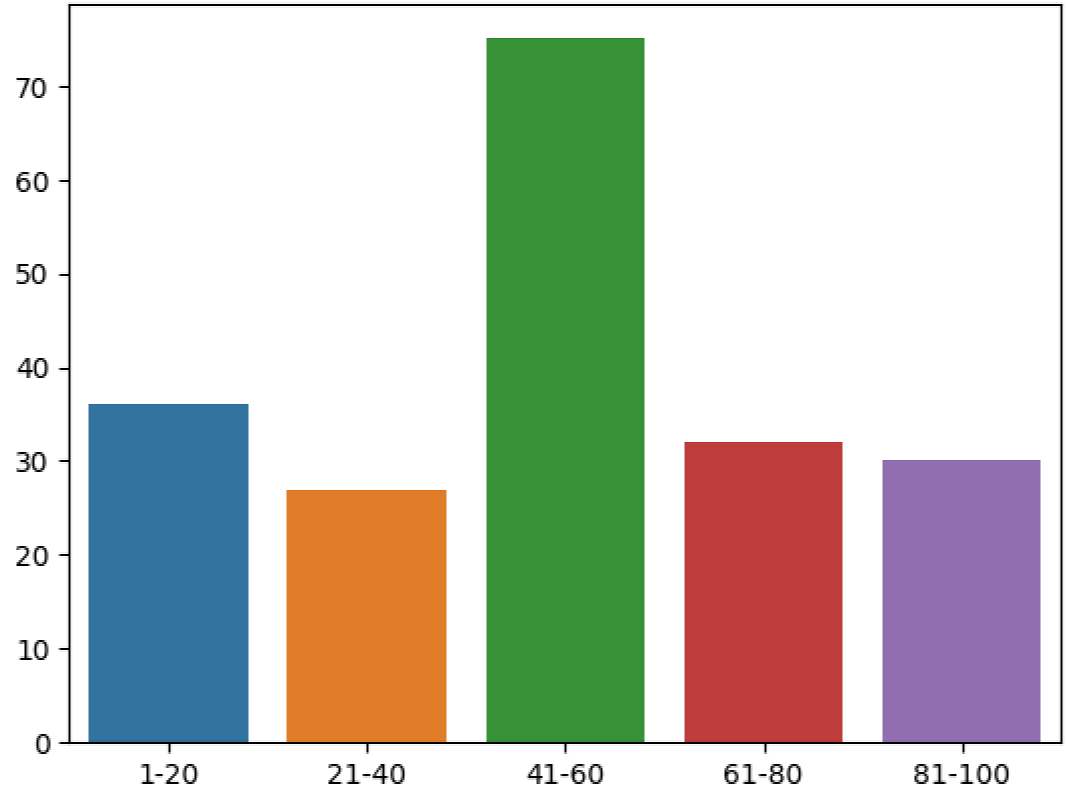
The graphical representation of ages and number of customers. So by this, we can say that customers of age group 27-35 are more in number than the other age groups

Now we are going to visualize the highest spending scores among the customers. This helps to find out the majority of spending scores of the customers

In [11]:
```
ss1_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Sp
ss21_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) & (df["
ss41_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) & (df["
ss61_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) & (df["
ss81_100 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 81) & (df[
```

In [12]:
```
x=["1-20","21-40","41-60","61-80","81-100"]
y=[len(ss1_20.values), len(ss21_40.values),len(ss41_60.values),len(ss61_80.values),
```

In [13]:
```
sns.barplot(x=x , y=y)
plt.figure(figsize=(10,5))
plt.title=("Spending scores of the customers")
plt.xlabel=("Spending Scores")
plt.ylabel=("score of customers")
plt.show()
```



```
<Figure size 1000x500 with 0 Axes>
```

So based on the bar graph we can see that the majority of spending scores among the customers is between 41-60

Now we are going to visualize the annual income of the customers From the obtained graph we can say that most of the customers are having an annual income between 61-90$ The annual income is being spitted into groups from 0-30,31-60,61-90,91-120,121-150.

In [14]:

```
ai0_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=0)&(df["Annual Income
ai31_60 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=31)&(df["Annual Incom
ai61_90 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=61)&(df["Annual Incom
ai91_120 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=91)&(df["Annual Inco
ai121_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=121)&(df["Annual In
```

In [15]:

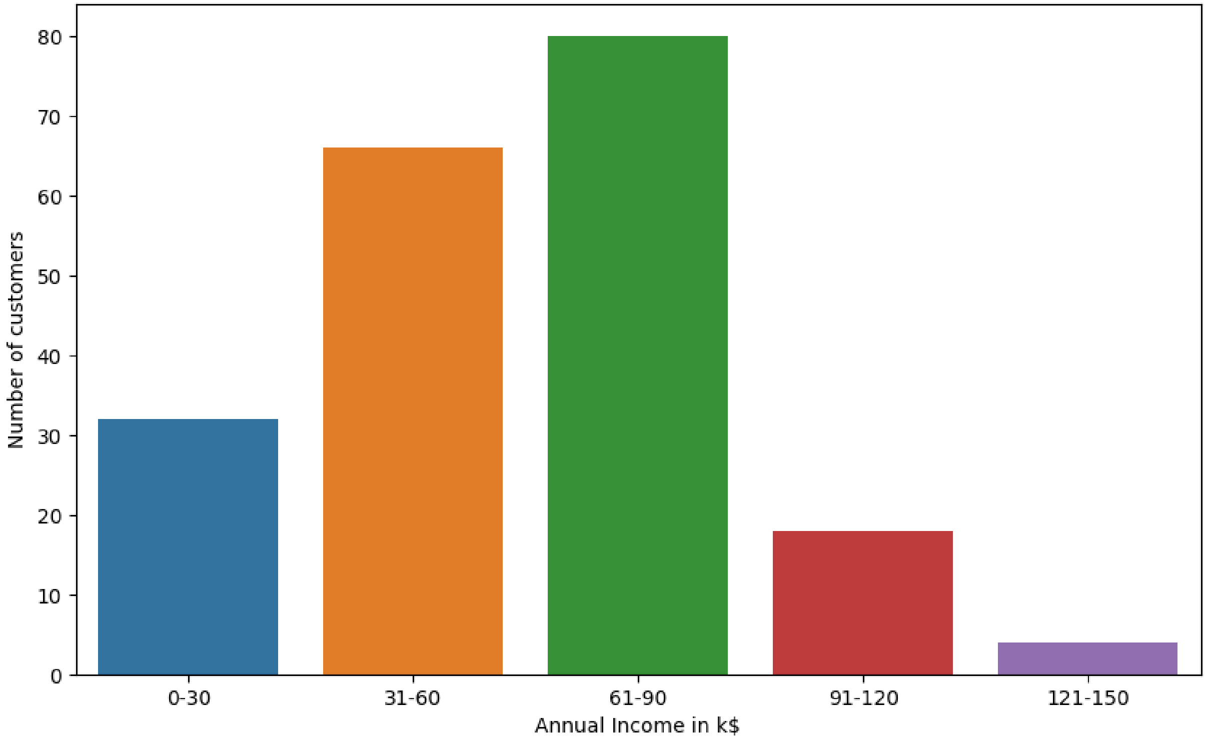
```
x = ["0-30", "31-60", "61-90", "91-120", "121-150"]
y = [
    len(ai0_30.values),
    len(ai31_60.values),
    len(ai61_90.values),
    len(ai91_120.values),
    len(ai121_150.values)
]
```

In [16]:

```
import matplotlib.pyplot as plt
from importlib import reload
plt=reload(plt)

plt.figure(figsize=(10,6))
```

```
sns.barplot(x=x,y=y,)
plt.title=("Annual Income of customers")
plt.xlabel("Annual Income in k$")
plt.ylabel("Number of customers")
plt.show()
```



The graph obtained shows that the majority of customers have the annual income between 61-90$

Now we will cluster the data by using the K- means algorithm. First, we need to place the values of the 2 variables which are spending scores and annual income in the variable named x in a form of an array. Now we have to find the number of clusters to be used the fundamental method which goes with the unsupervised method is the Elbow Method

Elbow Method

The Elbow method is used to find out the optimal value to be used in Kmeans In the line chart it resembles the arm with the elbow. The inflection in the curve resembles the underlined model that fits best at that point. So now we will use the elbow method to find out the number of clusters needed. We will first we will import means from sklearn lib and use wcss formula WCSS measures the sum of distances of observations from their cluster centroids which is given by the below formula.

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2$$

where Yi is centroid for observation Xi. The main goal is to maximize the number of clusters and in limiting cases each data point becomes its own cluster centroid.

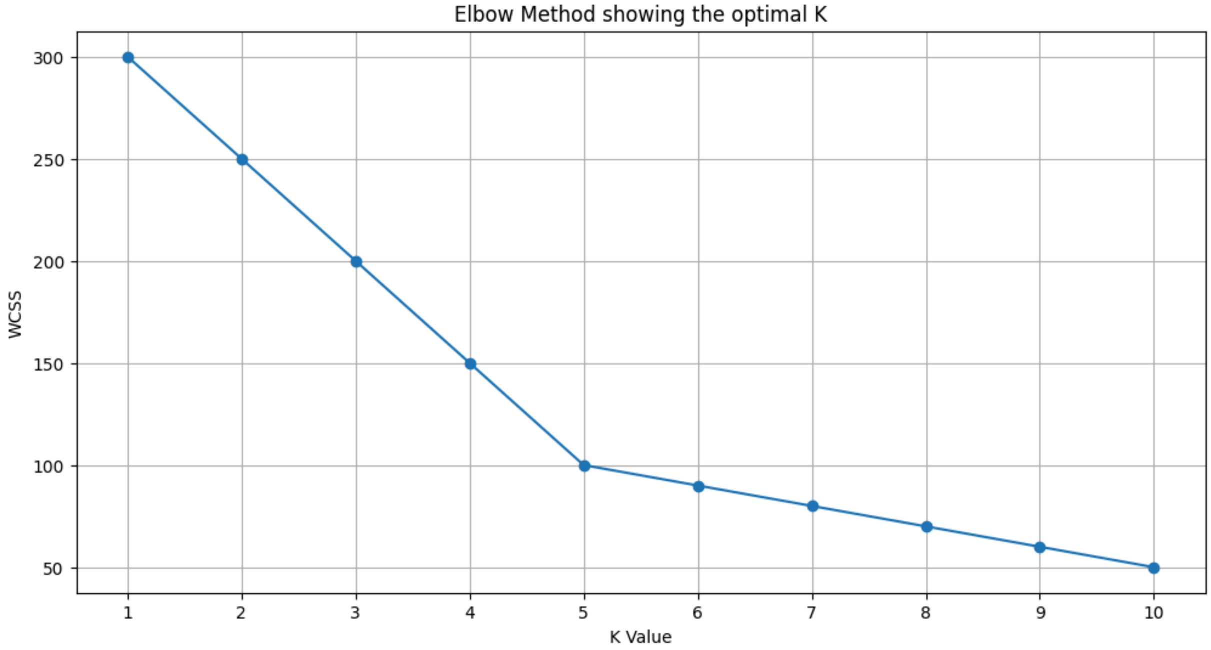
In [17]:
```
from sklearn.cluster import KMeans
wcss=[] #Within cluster sum of squares
```

In [18]:
```
x = np.random.rand(100, 2)  # Example: 100 samples with 2 features
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', n_init=10, random_state=0)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)
```

In [19]:
```
import matplotlib.pyplot as plt
from importlib import reload
plt=reload(plt)

wcss = [300, 250, 200, 150, 100, 90, 80, 70, 60, 50]  # Replace with your actual WC

plt.figure(figsize=(12, 6))
plt.grid()
plt.plot(range(1, 11), wcss, marker='o')  # Adjust marker for better visibility of
plt.xlabel('K Value')
plt.xticks(np.arange(1, 11, 1))
plt.ylabel('WCSS')
plt.title('Elbow Method showing the optimal K')
plt.show()
```

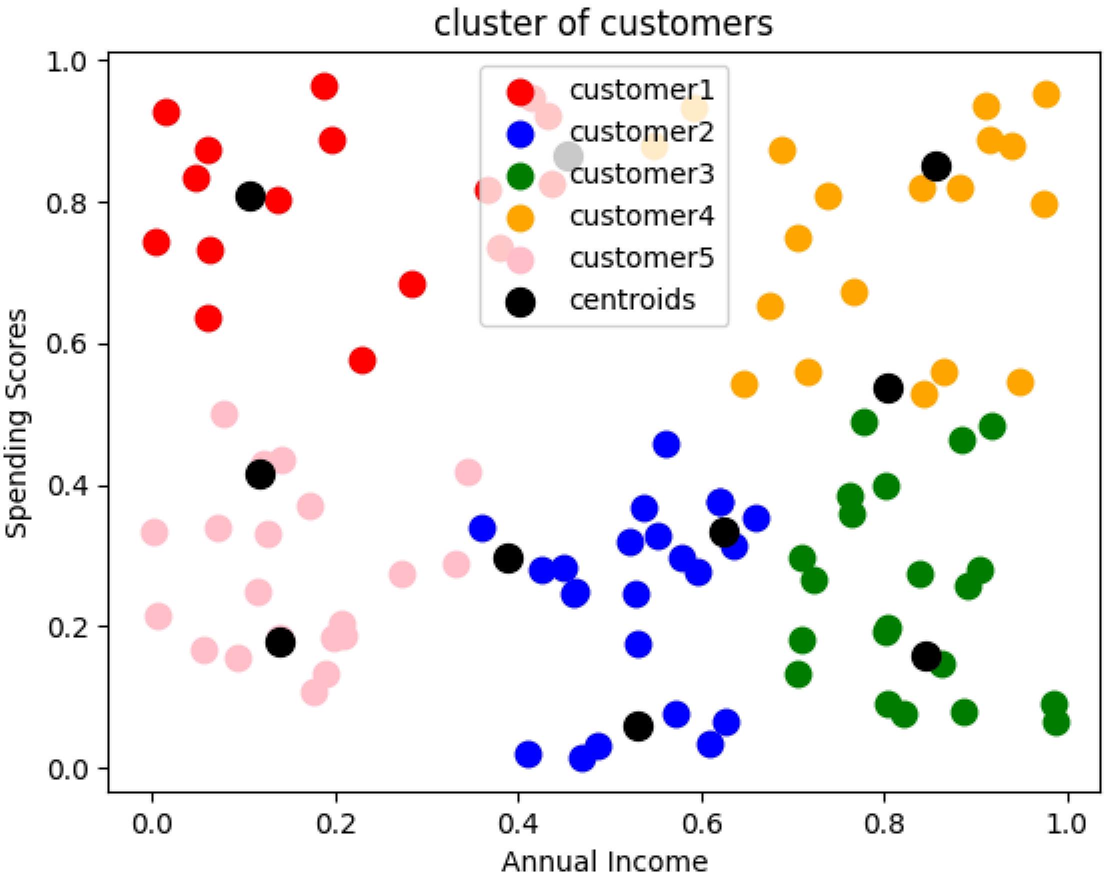


KMeans Algorithm

As we already know the number of clusters that are required is 5.Now Finally I made a plot to visualize the spending score of the customers with their annual income. The data points are separated into 5 classes which are represented in different colors as shown in the plot.

```
In [20]: kmeansmodel = KMeans(n_clusters=5, init='k-means++', random_state=0, n_init=10)

         y_kmeans=kmeansmodel.fit_predict(x)
         plt.scatter(x[y_kmeans==0,0],x[y_kmeans==0,1],s=80,c='red',label='customer1')
         plt.scatter(x[y_kmeans==1,0],x[y_kmeans==1,1],s=80,c='blue',label='customer2')
         plt.scatter(x[y_kmeans==2,0],x[y_kmeans==2,1],s=80,c='green',label='customer3')
         plt.scatter(x[y_kmeans==3,0],x[y_kmeans==3,1],s=80,c='orange',label='customer4')
         plt.scatter(x[y_kmeans==4,0],x[y_kmeans==4,1],s=80,c='pink',label='customer5')
         plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=100,c="blac
         plt.title(str('cluster of customers'))
         plt.xlabel(str('Annual Income'))
         plt.ylabel(str('Spending Scores'))
         plt.legend()
         plt.show()
```



Here each customer represents one color customer1 as red, customer 2 as blue, customer3 as green, customer4 as orange, and customer5 as pink. The black dots represent the centroids of the cluster.

We have classified the customers into 5 clusters through which we can see that customer1 is having average spending scores with the average income so this range of customers can be targeted in order to increase sales

Summary

K means clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major applications of K means clustering is the segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.

Contact Information.

Contact us for further inquiries or collaboration opportunities.

Email:

farhanashraf284284@gmail.com

Thank You for Your Attention!