|  |
| --- |
| **A Report on**  MALL CUSTOMER DATA ANALYSIS  Submitted for partial fulfillment of award of  **DEGREE**  **OF**  **BACHELOR OF COMPUTER APPLICATIONS**  Submitted By  RUDRAKSH KAUSHIK    Under the supervision of  PRATEEK GUPTA  C:\Users\aa\AppData\Local\Temp\samagra-2017.png    **INSTITUTE OF TECHNOLOGY & SCIENCE**  **MOHAN NAGAR, GHAZIABAD**  **Batch: 2021-2024**  **CERTIFICATE** This is to Certify that RUDRAKSH KAUSHIK has carried out the project work presented in this report entitled “Mall Customer Data Analysis” for the award of Bachelor Of Computer Applications from Institute of Technology & Science, Mohan Nagar, Ghaziabad, under my supervision. The report embodies result of original work and studies carried out by Student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else. Mr. Prateek Gupta  (Data Scientist) |

#### ACKNOWLEDGEMENT

I extend my heartfelt appreciation to all the individuals and organizations who have been a part of this college project and have contributed to its successful completion. I would like to express my gratitude to my project supervisor, trainer Mr. Prateek Gupta for their guidance, expertise, and constant support throughout the project duration

.Furthermore, I would like to acknowledge the support received from ITS for granting permission to conduct the research and providing necessary resources. Lastly, I would like to thank my friends and family for their unwavering support and encouragement during this journey.

ABSTRACT

A lot of customers buy products from the mall and to generate more revenue for the mall, the authorities need to attract these customers and for this large amount of capital is required. After the advertisement, the output is only around 30-40%. Hence customer segmentation comes into

the picture.

Customer Segmentation is a popular application of unsupervised learning and by using this technique we'll only focus on the potential customers (customers whose probability of buying the product is very high). With this technique, the output will drastically increase to 90-95%.

# 

# TABLE OF CONTENTS

* **Chapter 1: Introduction ……………………………………………..**

• 1.1 Overview……………………………………………………..

• 1.2 Problem statement……………………………………………

• 1.3 Scope…………………………………………………………

• 1.4 Purpose……………………………………………………….

• 1.5 Tools and methodology………………………………………

* **Chapter 2: System analysis………………………………………….**

• 2.1 Identification…………………………………………………

• 2.2 Preliminary investigation……………………………………

* **Chapter 3: Means of Project………………………………………...**

• 3.1 Hardware require…………………………………………….

• 3.2 Software requirement………………………………………..

• 3.3 Product Perspective…………………………………………

* **Chapter 4: Overall description………………………………………**

• 4.1 Product Perspective………………………………………….

• 2.1 Data flow diagram…………...……………………

* **Chapter 5: Specific Diagrams………………………………………**

• 5.1 Dataset………………………………………………………

• 5.2 Implementation and analysis………………………………..

• 5.3 Age plot analysis……………………………………………

• 5.4 Age vs spending score analysis……………………………..

• 5.5 Annual income vs spending score analysis………………....

* **Chapter 6: Product and project features……………………………**

• 6.1 Advantages of customer segmentation………………………

• 6.2 Why use customer segmentation?..........................................

• 6.3 Why use k-means clustering for customer segmentation?.....

* **Chapter 7: Code and results…………………………………………**
* **Chapter 8: Test cases and hypothesis problem…………………….**

• 8.1 How to find optimal number of clusters?................................

* **Chapters 9: Conclusion……………………………………………….**

• 9.1 Clustering analysis……………………………………………

* **Chapter 11: Future Scope of the Project…………………………….**

* **Chapter 10: Bibliography…………………………………………….**

**LIST OF TABLE**

5.1 Snapshot of dataset

8.1 Customer data

**LIST OF IMAGE**

5.2 Gender plot

5.3 Age plot

5.4 Age vs spending score

5.5 Annual income vs spending score

8.3 Aptimal cluster

9.1 Annual income vs spending score after clustering

# Chapter 1: INTRODUCATION

**1.1 Overview and Problem statement**

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.

**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.

**1.2 Purpose**

Data analysis helps businesses acquire relevant, accurate information, suitable for developing future marketing strategies, business plans, and realigning the company’s vision or mission.This project aims to perform customer segmentation on a Mall customer dataset using the K-Means clustering algorithm. The goal of this project is to cluster the customers based on their purchasing behavior and demographic characteristics.

**1.3 Scope**

The project helps us to understand the lifestyle of people visiting malls and such places on a regular basis. This assessment will help us and all the benefiting authorities to have a good knowledge about the people visiting these places which will eventually help the open up new arenas of development for personal and social benefits.

They would have a clear data of the needed ones to asses and examine all their future losses and gains and take informed decisions.

**1.4 Tools used:**

This project makes use of libraries of python and concepts of data analysis.

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

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.

**1.5 Methodology used:**

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

**1.6 Technology used**

The project makes use of libraries of python such as pandas numpy matplotlib. It enforces all the programs in the IDE environment. Programming Language: Python 3.6 Environment (Libraries and Technologies): Numpy, Pandas, Matplotlib, Seaborn, Jupyter

Notebook, Google Colab. 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

**Chapter 2: SYSTEM ANALYSIS.**

**2.1 Identification of need**

ABOUT PYTHON :

Python is a dynamic, interpreted (bytecode-compiled) language. There are no type declarations of variables, parameters, functions, or methods in source code. This makes the code short and flexible, and you lose the compile-time type checking of the source code. Python tracks the types of all values at runtime and flags code that does not make sense as it runs..

Identify details like age, geographical location, relationship status, and income for B2C customers. For B2B customers, identify details like company revenue, industry, products and services, and the technologies they use.

**2.2 Preliminary investigation**

The basic purpose behind Preliminary Investigation is to first clarify, understand and evaluate the Project Request. Our preliminary investigation started off in two ways i.e observing the visting patterns of customers that happened on site. And by asking people about their monthly income and expenses to understand them better.

Preliminary Investigation basically refers to the collection of information that guides the management of an organization to evaluate the merits and demerits of the project request and make an informed judgment about the feasibility of the proposed system.

**1)On site Observation:**

Here a detailed study was carried out, checking the existing visitors to understand the patterns effectively.This information helped us to understand how the project should operate. But after interviewing the persons, we got more details that further explain the project and shown whether assistance is merited economically, operationally and technically.

**2) Conducting Interviews:**

This method of investigation conducted by us involved questioning the concerned personnel to get the user’s (client) view about the project and the features they desired it to have. Some of the questions we asked are:

a) The number of customers regularly visiting their stores

b) The growth of Crowd with changing times and conditions

c) The level of money expenses made in a month acculumated sums up to the profit.

**Chapter 3: MEANS OF PROJECT**

**3.1 Hardware required:**

There are no requirement of hardware resources as the study is based on mall customers who visit the places frequently. We will be analysing their expenses income and more such details by studying their attributes and characteristics carefully.

Our study is solely based on the patterns of the customers. There day to day practices and their visits to the malls and such public spots.

**3.2 Software requirement**

• Excel

Microsoft Excel is one of the most common software used for data analysis. In addition to offering spreadsheet functions capable of managing and organising large data sets, Excel also includes graphing tools and computing capabilities like automated summation or “AutoSum.” Excel also includes Analysis ToolPak, which features data analysis tools capable of performing variance, regression, and statistical analysis

• Python

Python is routinely ranked as the most popular programming language in the world today

Unlike other programming languages, Python is relatively easy to learn and can be used for various tasks, including software, web development, and data analysis. In the world of data, Python is used to streamline, model, visualise, and analyse data using its built-in data analytics tools. One of the key features of Python that appeals to data analytics professionals is its many libraries, such as Pandas and Numpy, which offer a variety of powerful tools for many analytics needs.

• Jupyter Notebook

Jupyter Notebook is a web-based interactive environment for sharing computational documents or “notebooks.” Data analysts use Jupyter Notebooks to write and run code, clean data, data visualisation, machine learning, statistical analysis, and many other forms of data analysis. Furthermore, Jupyter Notebook allows users to combine data visualisations, code, comments, and numerous different programming languages in one place, allowing for an improved space to document a data analysis process and share them with others.

**Chapter 4: OVERALL DESCRIPTION**

Over the years, the competition amongst businesses is increased and the large historical data that is available has resulted in the widespread use of data mining techniques in extracting the meaningful and strategic information from the database of the organisation. Data mining is the process where methods are applied to extract data patterns in order to present it in the human readable format which can be used for the purpose of decision support.

According to Clustering techniques consider data tuples as objects. They partition the data objects into groups or clusters, 2 so that objects within a cluster are similar to one another and dissimilar to objects in other clusters. Customer Segmentation is the process of division of customer base into several groups called as customer segments such that each customer segment consists of customers who have similar characteristics. The segmentation is based on the similarity in different ways that are relevant to marketing such as gender, age, interests, and miscellaneous spending habits.

**4.1 Product Perspective :**

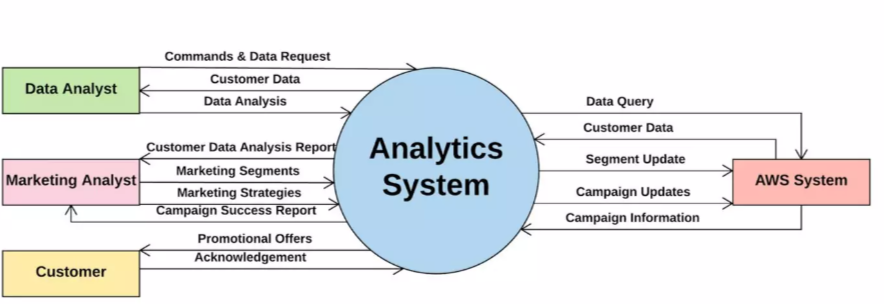
The overall product that the study deals with in the project report is the concept of people spending their resources. The existing method is storing customer data through paperwork and computer software (digital data) is increasing day by day. At end of the day they will analyse their data as how many things are sold or actual customer count etc. By analysing the collected data they got to know who is beneficial to their business and increase their sales. It requires more time and more paperwork. Also, it is not much effective solution to find the desired customers data.

Customer segmentation simply means grouping your customers according to various characteristics (for example grouping customers by age).

It’s a way for organizations to understand their customers. Knowing the differences between customer groups, it’s easier to make strategic decisions regarding product growth and marketing.

The opportunities to segment are endless and depend mainly on how much customer data you have at your use. Starting from the basic criteria, like gender, hobby, or age, it goes all the way to things like “time spent of website X” or “time since user opened our app”.

**4.2 Data Flow Diagram**

****

**Chapter 5: SPECIFIC DIAGRMS AND IMPLEMENTATION**

**5.1 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.

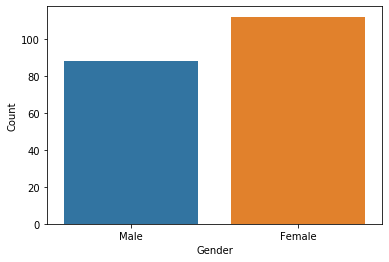


5.1 Snapshot of Dataset

**5.2 Implementation and analysis:**

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

**Gender Plot Analysis**

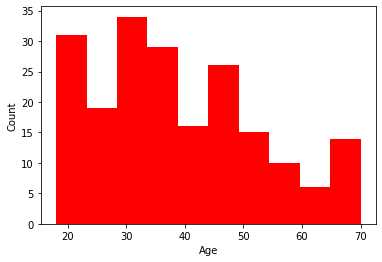


From the Count plot, it is observed that the number of Female customers is more than the total number of Male customers.

5.2 Gender Plot

**5.3 Age Plot Analysis**

5.3 Age plot

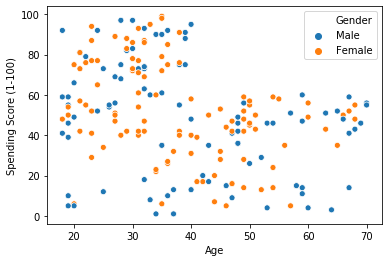


From the Histogram it is evident that there are 3 age groups that are more frequently shop at the mall, they are: 15-22 years, 30-40 years, and 45-50 years.

**5.4 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.

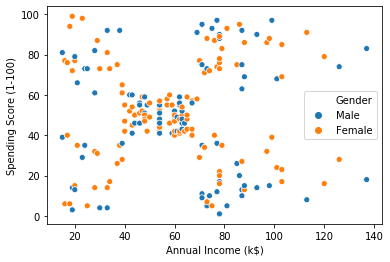
2. The customers having average spending score ie: in the range of 40-60 consists of the age group of the range 15-75 years and the count of males and females in this age group is also



5.4 Age Vs Spending Sore

**5.5 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)

5.5 Annual Income Vs Spendi

**Chapter 6: PRODUCT AND PROJECT FEATURES**

**6.1 Advantages of customer segmentation**

Implementing customer segmentation leads to plenty of new business opportunities. You can do a lot of optimization in:

• Budgeting

Nobody likes to invest in campaigns that don’t generate any new customers. Most companies don’t have huge marketing budgets, so that money has to be spent right. Segmentation enables you to target customers with the highest potential value first, so you get the most out of your marketing budget.

• Product design

Customer segmentation helps you understand what your users need. You can identify the most active users/customers, and optimize your application/offer towards their needs.

• Promotion

Properly implemented customer segmentation helps you plan special offers and deals. Frequent deals have become a staple of e-commerce and commercial software in the past few years. If you reach a customer with just the right offer, at the right time, there’s a huge chance they’re going to buy. Customer segmentation will help you tailor your special offers perfectly.

• Marketing

The marketing strategy can be directly improved with segmentation because you can plan personalized marketing campaigns for different customer segments, using the channels that they use the most.

• Customer satisfaction

By studying different customer groups, you learn what they value the most about your company. This information will help you create personalized products and services that perfectly fit your customers’ preferences.

**6.2 Why use customer segmentation??**

• More time

Manual customer segmentation is time-consuming. It takes months, even years to analyze piles of data and find patterns manually. Also if done heuristically, it may not have the accuracy to be useful as expected.

Customer segmentation used to be done manually and wasn’t too precise. You’d manually create and populating different data tables, and analyze the data like a detective with a looking glass. Now, it’s much better (and relatively easy thanks to rapid progress in ML) to just use machine learning, which can free up your time to focus on more demanding problems that require creativity to solve.

• Ease of retraining

Customer Segmentation is not a “develop once and use forever” type of project. Data is ever-changing, trends oscillate, everything keeps changing after your model is deployed. Usually, more labeled data becomes available after development, and it’s a great resource for improving the overall performance of your model.

There are many ways to update customer segmentation models, but here are the two main approaches:

✓ Use the old model as the starting point and retrain it.

✓ Keep the existing model and combine its output with a new model.

• Better scaling

Machine learning models deployed in production support scalability, thanks to cloud infrastructure. These models are quite flexible for future changes and feedback. For example, consider a company that has 10000 customers today, and they’ve implemented a customer segmentation model. After a year, if the company has 1 million customers, then ideally we don’t need to create a separate project to handle this increased data. Machine Learning models have the inherent capability to handle more data and scale in production.

• Higher accuracy

The value of an optimal number of clusters for given customer data is easy to find using machine learning methods like the elbow method. Not only the optimal number of clusters but also the performance of the model is far better when we use machine learning.

**6.3 Why use K-means clustering for customer segmentation?**

Unlike supervised learning algorithms, K-means clustering is an unsupervised machine learning algorithm. This algorithm is used when we have unlabelled data. Unlabelled data means input data without categories or groups provided. Our customer segmentation data is like this for this problem.

**Chapter 7: CODE & OUTPUT**

#Importing the necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from mpl\_toolkits.mplot3d import Axes3D

%matplotlib inline

#Reading the excel file

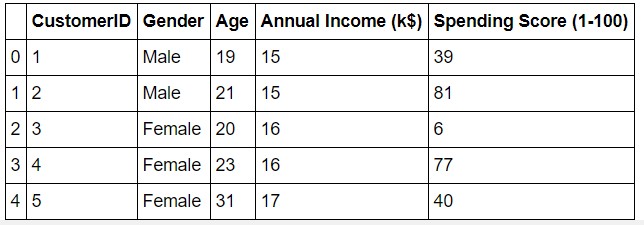
data=pd.read\_excel("Mall\_Customers.xlsx")

#Number of customers we have

print("Number of customers we have data for-" , len(data))

**The data has 200 entries, that is data from 200 customers.**

**So let us have a look at the data.**

****

data.head()

|  | **CustomerID** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
| --- | --- | --- | --- | --- |
| **CustomerID** | 1.000000 | -0.026763 | 0.977548 | 0.013835 |
| **Age** | -0.026763 | 1.000000 | -0.012398 | -0.327227 |
| **Annual Income (k$)** | 0.977548 | -0.012398 | 1.000000 | 0.009903 |
| **Spending Score (1-100)** | 0.013835 | -0.327227 | 0.009903 | 1.000000 |

The data seems to be interesting. Let us look at the data distribution.

**Annual Income Distribution:**

#Distribution of Annnual Income

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

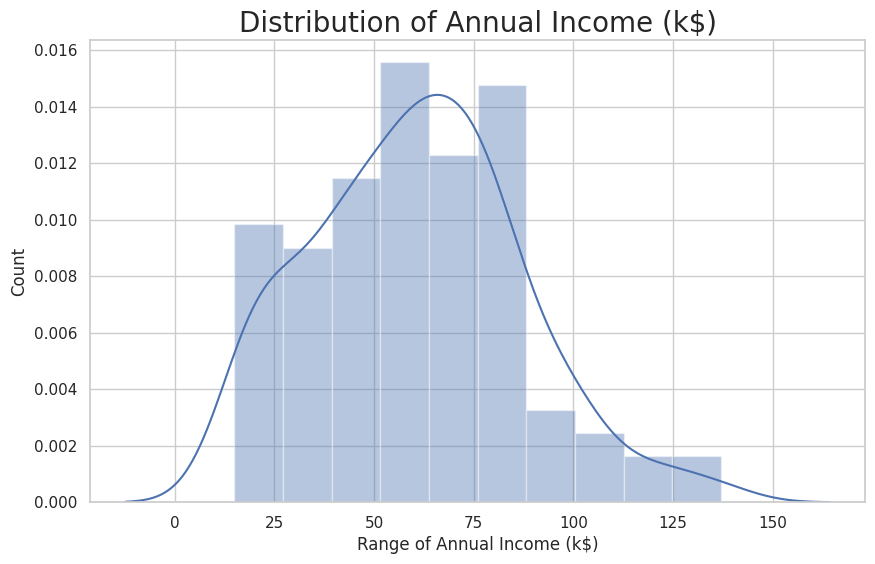
sns.set(style = 'whitegrid')

sns.distplot(data['Annual Income (k$)'])

plt.title('Distribution of Annual Income (k$)', fontsize = 20)

plt.xlabel('Range of Annual Income (k$)')

plt.ylabel('Count')



**Most of the annual income falls between 50K to 85K.**

**Age distrubation:**

#Distribution of age

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

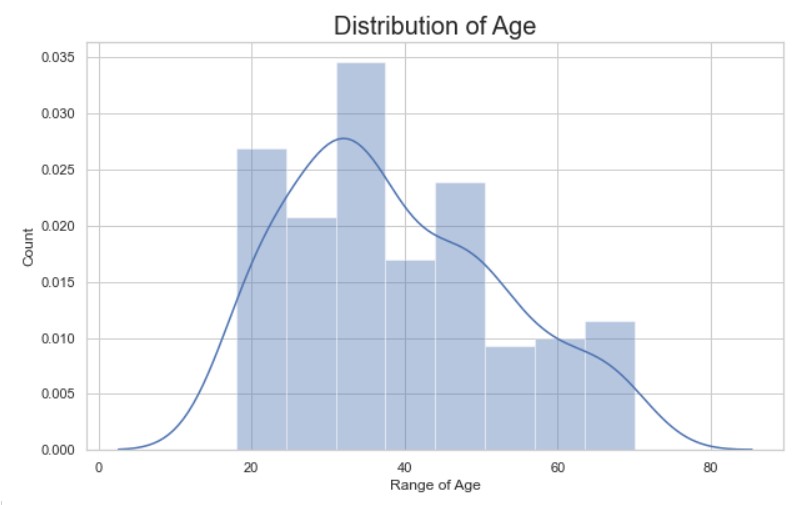
sns.set(style = 'whitegrid')

sns.distplot(data['Age'])

plt.title('Distribution of Age', fontsize = 20)

plt.xlabel('Range of Age')

plt.ylabel('Count



**There are customers of a wide variety of ages.**

**Spending score distrbuation:**

#Distribution of spending score

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

sns.set(style = 'whitegrid')

sns.distplot(data['Spending Score (1-100)'])

plt.title('Distribution of Spending Score (1-100)', fontsize = 20)

plt.xlabel('Range of Spending Score (1-100)')

plt.ylabel('Count')



**The maximum spending score is in the range of 40 to 60.**

**Gender analysis:**

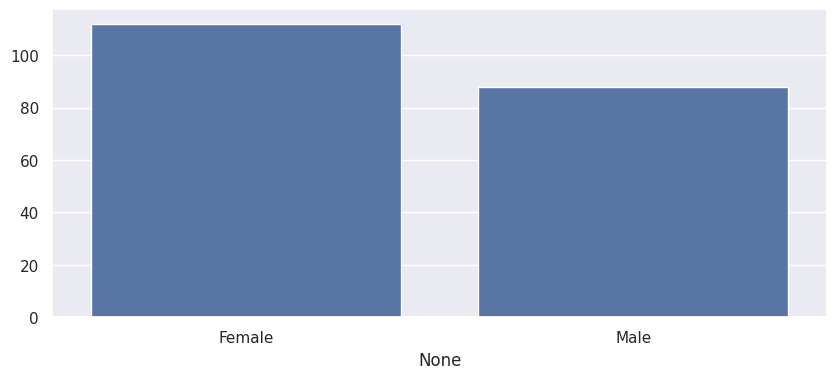
genders = data["Genre"].value\_counts()

sns.set\_style("darkgrid")

plt.figure(figsize=(10,4))

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

plt.show()



age18\_25 = data.Age[(data.Age <= 25) & (data.Age >= 18)]

age26\_35 = data.Age[(data.Age <= 35) & (data.Age >= 26)]

age36\_45 = data.Age[(data.Age <= 45) & (data.Age >= 36)]

age46\_55 = data.Age[(data.Age <= 55) & (data.Age >= 46)]

age55above = data.Age[data.Age >= 56]

x = ["18-25","26-35","36-45","46-55","55+"]

y = [len(age18\_25.values),len(age26\_35.values),len(age36\_45.values),len(age46\_55.values),len(age55above.values)]

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

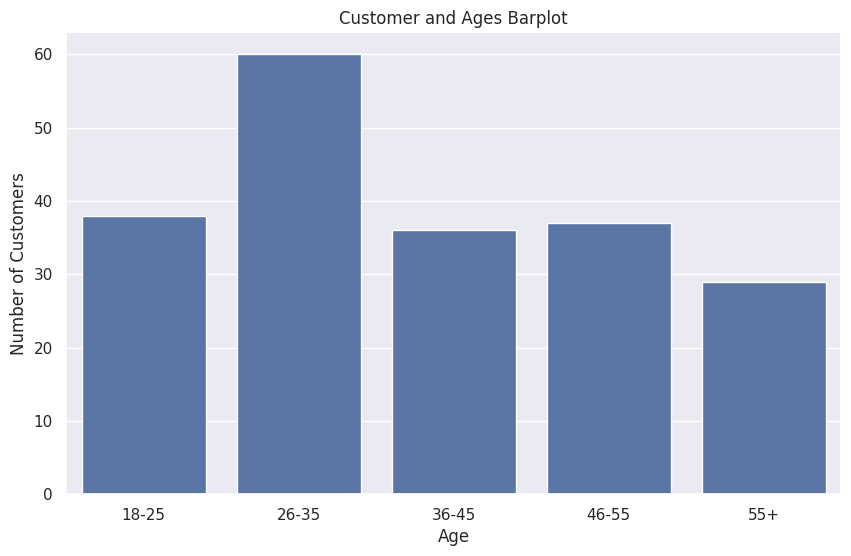
sns.barplot(x=x, y=y)

plt.title("Customer and Ages Barplot")

plt.xlabel("Age")

plt.ylabel("Number of Customers")

plt.show()



ss1\_20 = data["Spending Score (1-100)"][(data["Spending Score (1-100)"] >= 1) & (data["Spending Score (1-100)"] <= 20)]

ss21\_40 = data["Spending Score (1-100)"][(data["Spending Score (1-100)"] >= 21) & (data["Spending Score (1-100)"] <= 40)]

ss41\_60 = data["Spending Score (1-100)"][(data["Spending Score (1-100)"] >= 41) & (data["Spending Score (1-100)"] <= 60)]

ss61\_80 = data["Spending Score (1-100)"][(data["Spending Score (1-100)"] >= 61) & (data["Spending Score (1-100)"] <= 80)]

ss81\_100 = data["Spending Score (1-100)"][(data["Spending Score (1-100)"] >= 81) & (data["Spending Score (1-100)"] <= 100)]

score\_x = ["1-20", "21-40", "41-60", "61-80", "81-100"]

score\_y = [len(ss1\_20.values), len(ss21\_40.values), len(ss41\_60.values), len(ss61\_80.values), len(ss81\_100.values)]

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

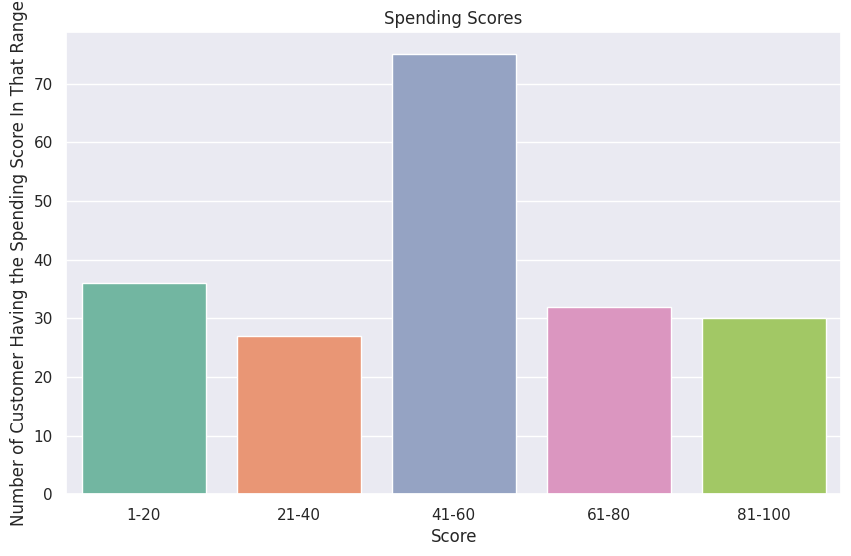
sns.barplot(x=score\_x, y=score\_y,palette="Set2")

plt.title("Spending Scores")

plt.xlabel("Score")

plt.ylabel("Number of Customer Having the Spending Score In That Range")

plt.show()



ai0\_30 = data["Annual Income (k$)"][(data["Annual Income (k$)"] >= 0) & (data["Annual Income (k$)"] <= 30)]

ai31\_60 = data["Annual Income (k$)"][(data["Annual Income (k$)"] >= 31) & (data["Annual Income (k$)"] <= 60)]

ai61\_90 = data["Annual Income (k$)"][(data["Annual Income (k$)"] >= 61) & (data["Annual Income (k$)"] <= 90)]

ai91\_120 = data["Annual Income (k$)"][(data["Annual Income (k$)"] >= 91) & (data["Annual Income (k$)"] <= 120)]

ai121\_150 = data["Annual Income (k$)"][(data["Annual Income (k$)"] >= 121) & (data["Annual Income (k$)"] <= 150)]

income\_x = ["$ 0 - 30,000", "$ 30,001 - 60,000", "$ 60,001 - 90,000", "$ 90,001 - 120,000", "$ 120,001 - 150,000"]

income\_y = [len(ai0\_30.values), len(ai31\_60.values), len(ai61\_90.values), len(ai91\_120.values), len(ai121\_150.values)]

plt.figure(figsize=(15,6))

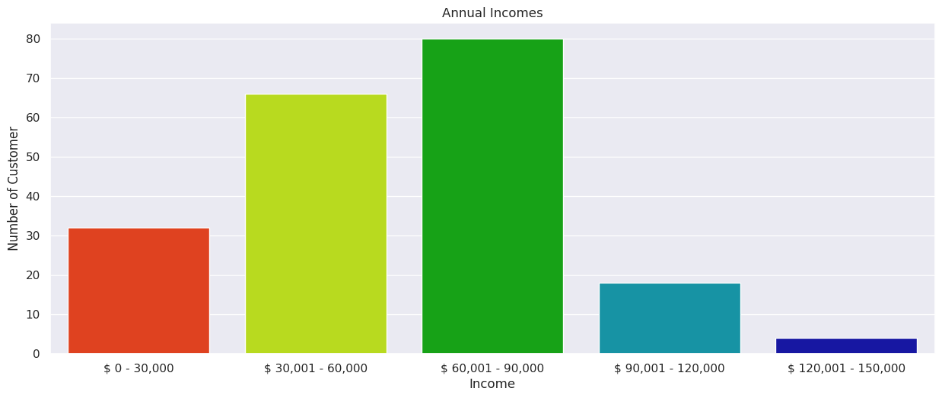
sns.barplot(x=income\_x, y=income\_y, palette="nipy\_spectral\_r")

plt.title("Annual Incomes")

plt.xlabel("Income")

plt.ylabel("Number of Customer")

plt.show()



#Taking another look at the data

data

| **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 |
| **...** | ... | ... | ... | ... | ... |
| **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 |

200 rows × 5 columns

**Clustering based on 2 features**

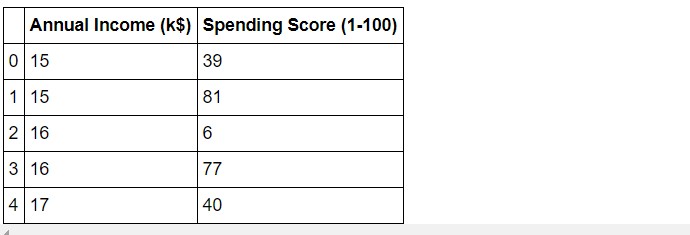
First, we work with two features only, annual income and spending score.

df1 = data[["CustomerID", "Age", "Annual Income (k$)", "Spending Score (1-100)"]]

X = df1[["Annual Income (k$)", "Spending Score (1-100)"]]

#The input data

X.head()



#Scatterplot of the input data

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

sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',  data = X  ,s = 60 )

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Spending Score (1-100) vs Annual Income (k$)')

plt.show()

**The data does seem to hold some patterns.**

****

#Importing KMeans from sklearn

from sklearn.cluster import KMeans

wcss=[]

for i in range(1,11):

    km=KMeans(n\_clusters=i)

    km.fit(X)

    wcss.append(km.inertia\_)

#The elbow curve

plt.figure(figsize=(12,6))

plt.plot(range(1,11),wcss)

plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")

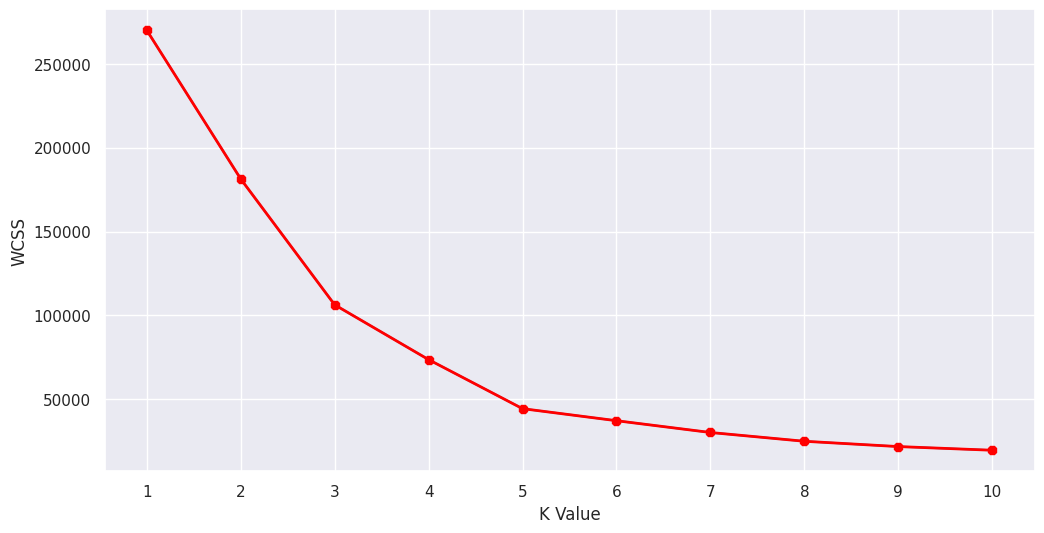
plt.xlabel("K Value")

plt.xticks(np.arange(1,11,1))

plt.ylabel("WCSS")

plt.show()

**The Plot:**



This is known as the elbow graph, the x-axis being the number of clusters, the number of clusters is taken at the elbow joint point. This point is the point where making clusters is most relevant as here the value of WCSS suddenly stops decreasing. Here in the graph, after 5 the drop is minimal, so we take 5 to be the number of clusters**.**

#this is known as the elbow graph , the x axis being the number of clusters

#the number of clusters is taken at the elbow joint point

#this point is the point where making clusters is most relevant

#the numbers of clusters is kept at maximum

#Taking 5 clusters

km1=KMeans(n\_clusters=5)

#Fitting the input data

km1.fit(X)

#predicting the labels of the input data

y=km1.predict(X)

#The new dataframe with the clustering done

df1.head()

| **index** | **CustomerID** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** |
| --- | --- | --- | --- | --- |
| **0** | 1 | 19 | 15 | 39 |
| **1** | 2 | 21 | 15 | 81 |
| **2** | 3 | 20 | 16 | 6 |
| **3** | 4 | 23 | 16 | 77 |
| **4** | 5 | 31 | 17 | 40 |

from sklearn.cluster import KMeans

import matplotlib.pyplot as plt

import numpy as np

wcss=[]

for i in range(1,11):

    km=KMeans(n\_clusters=i)

    km.fit(X)

    wcss.append(km.inertia\_)

#The elbow curve

plt.figure(figsize=(12,6))

plt.plot(range(1,11),wcss)

plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")

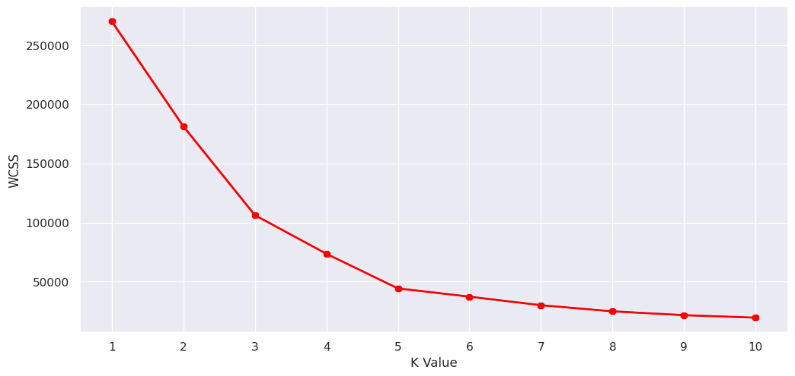
plt.xlabel("K Value")

plt.xticks(np.arange(1,11,1))

plt.ylabel("WCSS")

plt.show()

**The plot:**



#this is known as the elbow graph , the x axis being the number of clusters

#the number of clusters is taken at the elbow joint point

#this point is the point where making clusters is most relevant

#the numbers of clusters is kept at maximum

#Taking 5 clusters

km1=KMeans(n\_clusters=5)

#Fitting the input data

km1.fit(X)

#predicting the labels of the input data

y=km1.predict(X)

#adding the labels to a column named label

df1["label"] = y

#The new dataframe with the clustering done

df1.head()

| **index** | **CustomerID** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** | **label** |
| --- | --- | --- | --- | --- | --- |
| **0** | 1 | 19 | 15 | 39 | 4 |
| **1** | 2 | 21 | 15 | 81 | 0 |
| **2** | 3 | 20 | 16 | 6 | 4 |
| **3** | 4 | 23 | 16 | 77 | 0 |
| **4** | 5 | 31 | 17 | 40 | 4 |

#Scatterplot of the clusters

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

sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',hue="label",

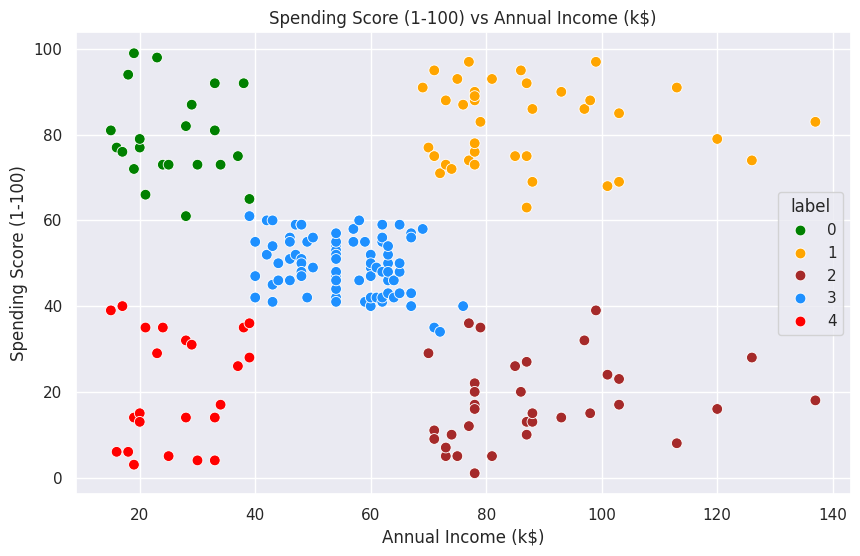
                 palette=['green','orange','brown','dodgerblue','red'], legend='full',data = df1  ,s = 60 )

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score (1-100)')

plt.title('Spending Score (1-100) vs Annual Income (k$)')

plt.show()



cust1=df1[df1["label"]==1]

print('Number of customer in 1st group=', len(cust1))

print('They are -', cust1["CustomerID"].values)

print("--------------------------------------------")

cust2=df1[df1["label"]==2]

print('Number of customer in 2nd group=', len(cust2))

print('They are -', cust2["CustomerID"].values)

print("--------------------------------------------")

cust3=df1[df1["label"]==0]

print('Number of customer in 3rd group=', len(cust3))

print('They are -', cust3["CustomerID"].values)

print("--------------------------------------------")

cust4=df1[df1["label"]==3]

print('Number of customer in 4th group=', len(cust4))

print('They are -', cust4["CustomerID"].values)

print("--------------------------------------------")

cust5=df1[df1["label"]==4]

print('Number of customer in 5th group=', len(cust5))

print('They are -', cust5["CustomerID"].values)

print("--------------------------------------------")

**OUTPUT:**

Number of customer in 1st group= 39

They are - [124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158

160 162 164 166 168 170 172 174 176 178 180 182 184 186 188 190 192 194

196 198 200]

--------------------------------------------

Number of customer in 2nd group= 22

They are - [ 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 46]

--------------------------------------------

Number of customer in 3rd group= 81

They are - [ 44 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63

64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81

82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117

118 119 120 121 122 123 127 133 143]

--------------------------------------------

Number of customer in 4th group= 23

They are - [ 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45]

--------------------------------------------

Number of customer in 5th group= 35

They are - [125 129 131 135 137 139 141 145 147 149 151 153 155 157 159 161 163 165

167 169 171 173 175 177 179 181 183 185 187 189 191 193 195 197 199]

**k-Means Clustering on the basis of 3D data**

df2 = data[["CustomerID", "Genre", "Age", "Annual Income (k$)", "Spending Score (1-100)"]]

df2.head()

#Taking the features

X2=df2[["Age","Annual Income (k$)","Spending Score (1-100)"]]

#Now we calculate the Within Cluster Sum of Squared Errors (WSS) for different values of k.

wcss = []

for k in range(1,11):

    kmeans = KMeans(n\_clusters=k, init="k-means++")

    kmeans.fit(X2)

    wcss.append(kmeans.inertia\_)

plt.figure(figsize=(12,6))

plt.plot(range(1,11),wcss, linewidth=2, color="red", marker ="8")

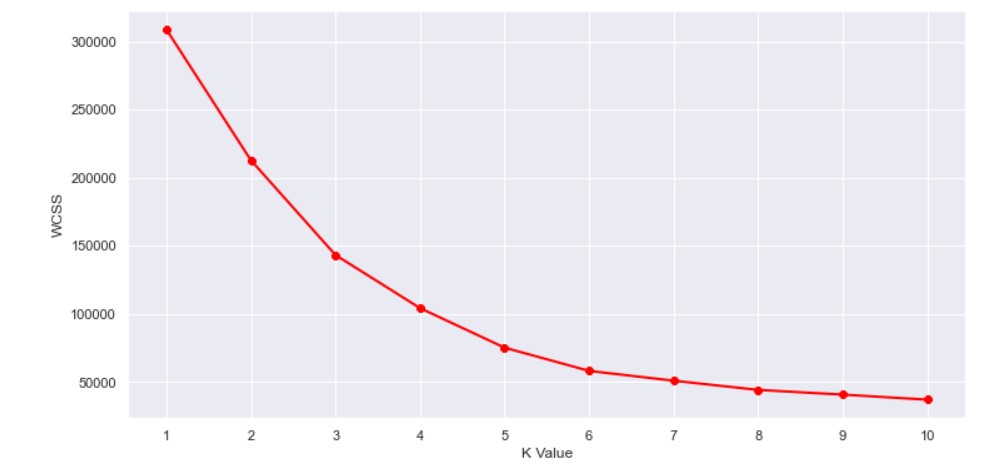
plt.xlabel("K Value")

plt.xticks(np.arange(1,11,1))

plt.ylabel("WCSS")

plt.show()

**The plot:**



**Here can assume that K=5 will be a good value.**

#We choose the k for which WSS starts to diminish

km2 = KMeans(n\_clusters=5)

y2 = km.fit\_predict(X2)

df2["label"] = y2

#The new data

df2.head()

**The data:**

| **CustomerID** | **Genre** | **Age** | **Annual Income (k$)** | **Spending Score (1-100)** | **label** |
| --- | --- | --- | --- | --- | --- |
| **0** | 1 | Male | 19 | 15 | 39 | 3 |
| **1** | 2 | Male | 21 | 15 | 81 | 7 |
| **2** | 3 | Female | 20 | 16 | 6 | 3 |
| **3** | 4 | Female | 23 | 16 | 77 | 7 |
| **4** | 5 | Female | 31 | 17 | 40 | 3 |

#3D Plot as we did the clustering on the basis of 3 input features

fig = plt.figure(figsize=(20,10))

ax = fig.add\_subplot(111, projection='3d')

ax.scatter(df2.Age[df2.label == 0], df2["Annual Income (k$)"][df2.label == 0], df2["Spending Score (1-100)"][df2.label == 0], c='purple', s=60)

ax.scatter(df2.Age[df2.label == 1], df2["Annual Income (k$)"][df2.label == 1], df2["Spending Score (1-100)"][df2.label == 1], c='red', s=60)

ax.scatter(df2.Age[df2.label == 2], df2["Annual Income (k$)"][df2.label == 2], df2["Spending Score (1-100)"][df2.label == 2], c='blue', s=60)

ax.scatter(df2.Age[df2.label == 3], df2["Annual Income (k$)"][df2.label == 3], df2["Spending Score (1-100)"][df2.label == 3], c='green', s=60)

ax.scatter(df2.Age[df2.label == 4], df2["Annual Income (k$)"][df2.label == 4], df2["Spending Score (1-100)"][df2.label == 4], c='yellow', s=60)

ax.view\_init(35, 185)

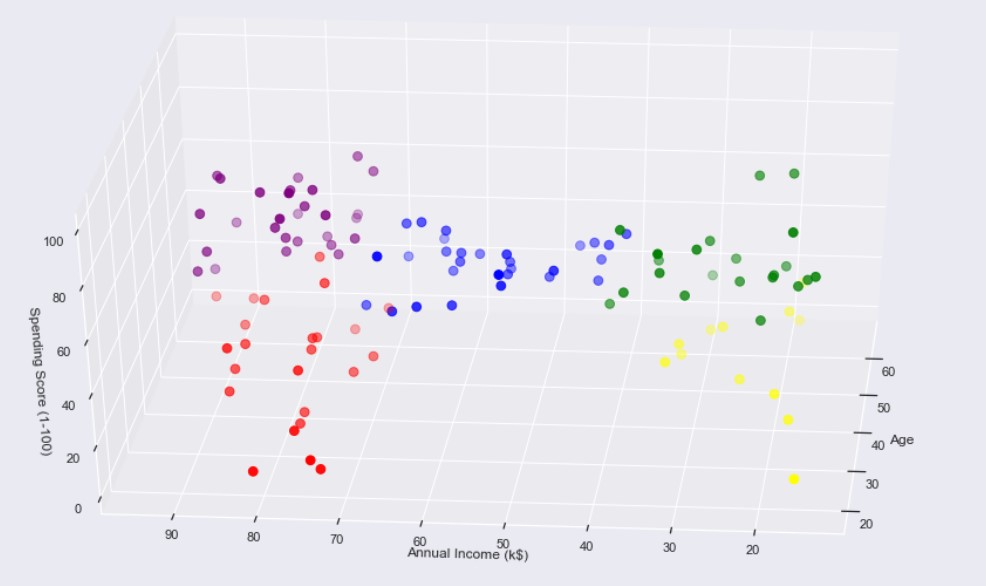
plt.xlabel("Age")

plt.ylabel("Annual Income (k$)")

ax.set\_zlabel('Spending Score (1-100)')

plt.show()

**The output:**



What we get is a 3D plot. Now, if we want to know the customer IDs, we can do that too.

cust1=df2[df2["label"]==1]

print('Number of customer in 1st group=', len(cust1))

print('They are -', cust1["CustomerID"].values)

print("--------------------------------------------")

cust2=df2[df2["label"]==2]

print('Number of customer in 2nd group=', len(cust2))

print('They are -', cust2["CustomerID"].values)

print("--------------------------------------------")

cust3=df2[df2["label"]==0]

print('Number of customer in 3rd group=', len(cust3))

print('They are -', cust3["CustomerID"].values)

print("--------------------------------------------")

cust4=df2[df2["label"]==3]

print('Number of customer in 4th group=', len(cust4))

print('They are -', cust4["CustomerID"].values)

print("--------------------------------------------")

cust5=df2[df2["label"]==4]

print('Number of customer in 5th group=', len(cust5))

print('They are -', cust5["CustomerID"].values)

print("--------------------------------------------")

**The output we get:**

Number of customer in 1st group= 11

They are - [180 182 184 186 188 190 192 194 196 198 200]

--------------------------------------------

Number of customer in 2nd group= 10

They are - [181 183 185 187 189 191 193 195 197 199]

--------------------------------------------

Number of customer in 3rd group= 28

They are - [ 41 47 51 54 55 56 57 58 60 61 63 64 65 68 71 73 74 75

81 83 87 91 103 107 109 110 111 117]

--------------------------------------------

Number of customer in 4th group= 9

They are - [ 1 3 5 17 21 27 29 39 43]

--------------------------------------------

Number of customer in 5th group= 28

They are - [124 126 128 130 132 134 136 138 140 142 144 146 148 150 152 154 156 158

160 162 164 166 168 170 172 174 176 178]

**Chapter 8: TEST CASES AND HYPOTHESIS PROBLEM**

**Labels for each input data point.**

At the end of implementation, we’re going to get output such as a group of clusters along with which customer belongs to which cluster.:

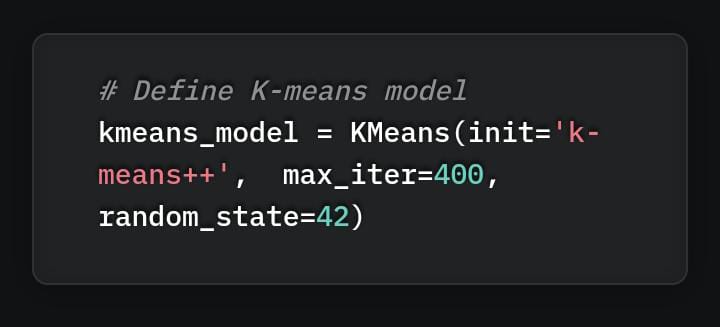
First, we need to implement the required Python libraries as shown in the table below. 

We’ve imported the pandas, NumPy sklearn, plotly and matplotlib libraries. Pandas and NumPy are used for data wrangling and manipulation, sklearn is used for modelling, and plotly along with matplotlib will be used to plot graphs and images.

After importing the library, our next step is to load the data in the pandas data frame. For this, we’re going to use the read\_csv method of pandas

After loading the data, we need to define the K- means model. This is done with the help of the KMeans class that we imported from sklearn, as shown in the code below.





After defining the model, we want to train is using a training dataset. This is implemented with the use of the fit method, as shown in the code below.



Though we have trained a K-means model up to these points, we haven’t found the optimal number of clusters required in this case of customer segmentation. Finding the optimal number of clusters, for the given dataset is important for producing a high-performant k-means clustering model.

In the upcoming section, we’re going to find the optimal number of clusters of the given dataset and then re-train the k-means clustering model with these optimal values of k. This will produce our final model.

**8.1 How to find optimal number of clusters**

Finding the optimal number of clusters is one of the key tasks when implementing a k-means clustering algorithm. It’s worth noting that a k-means clustering model might converge for any value of K, but at the same time, not all values of K will produce the best model.

For some datasets, data visualization can help understand the optimal number of clusters, but this doesn’t apply to all datasets. We have a few methods, such as the elbow method, gap statistic method and average silhouette method, to assess the optimal number of clusters for a given dataset. We’ll discuss them one by one.

The elbow method finds the value of the optimal number of clusters using the total within-cluster sum of square values. This represents how spread-apart the generated clusters are from one another. In this case, the K-means algorithm is evaluated for several values of k, and the within-cluster sum of square values is calculated for each value of k. After this, we plot the K versus the sum of square values. After analyzing this graph, the number of clusters is selected, so that adding a new cluster doesn’t change the values of the sum of square values significantly.

Average silhouette method is a measure of how well each data point fits its corresponding cluster. This method evaluates the quality of clustering. As a general rule, a high average silhouette width denotes better clustering output

Gap statistic method is a measure of the value of gap statistics. Gap statistics is the difference between the total intracluster changes for various values of k compared to their expected values. This calculation is done using the null reference distribution of data points. The optimal number of clusters is the value that maximizes the value of gap statistics

We’re going to use the elbow method. The K-means clustering algorithm clusters data by separating given data points in k groups of equal variances. This effectively minimizes a parameter named inertia. Inertia is nothing but within-cluster sum-of-squares distances in this case.

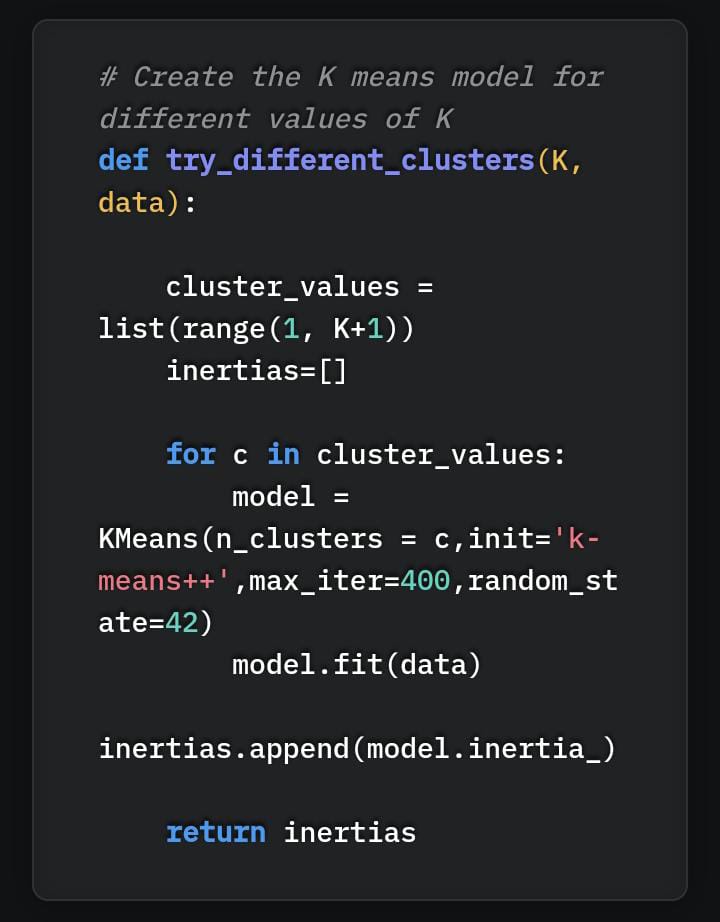
When we use the elbow method, we gradually increase the number of clusters from 2 until we reach the number of clusters where adding more clusters won’t cause a significant drop in the values of inertia.

The stage at this number of clusters is called the elbow of the clustering model. We’ll see that in our case it’s K =5.

For implementing the elbow method, the below function named “try\_different\_clusters” is created first. It takes two values as input:

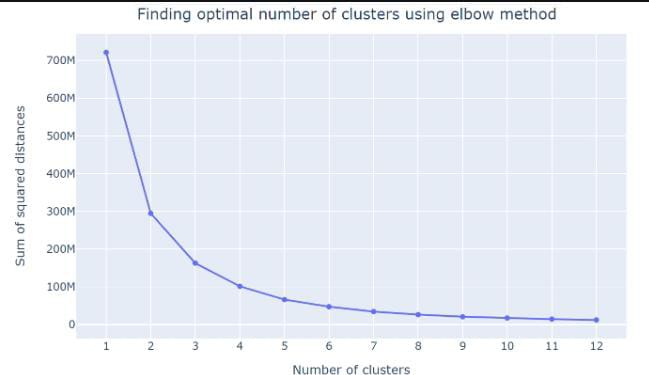
K (number of clusters),

data (input data).



The method try\_different\_clusters is called using the below code, where we pass the values of K from 1 to 12 and calculate the inertia for each value of k.

We can generate the below plot using the above code. The elbow of the code is at K=5. We have chosen 5 as if we increase the number of clusters to more than 5, there is very small change in the inertia or sum of the squared distance.

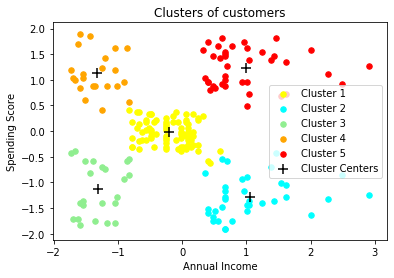


**Optimal value of k is 5**

The stage at which the number of clusters is optimal is called the elbow of the clustering model. For example, in the below image, the elbow is at five clusters (K =5). Adding more than 5 clusters will cause the creation of an inefficient or less performant clustering model.

**Chapter 9: 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.



9.1 Annual Income Vs Spending Score after Clustering

**9.1Clustering 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.

**Chapter 10: Future Scope of the Project**

Data Analytics helps government organizations and companies collect data and identify patterns in that data. These extensive insights into the data help organizations in decision-making based on the data, automating the process

Some of these benefits are:

1) Data Aggregation and Analysis

Data analytics can help in collecting big data and solving major issues that government organizations face. For example, the government has decided that it will partner with many IT firms to reduce power loss issues.

Discovering Untapped Markets and Their Potential

India has several large untapped sectors that data analytics and data analytics courses with placements can help in identifying, analysing, and tapping into.

IT Sector

The major share (43%) of the data analytics industry in the market is in the IT industry. A few of the IT giants in India are Accenture, Tata Consultancy Services, Cognizant, Infosys, Capgemini, and Wipro.

The tech demand in this sector and the gross employee addition, is set to rise in the second half of the financial year 2022. Additionally, India has also noticed great innovation in many industries through the IT sector.

2. Financial Services, Banking, and Insurance Sectors

This is the second-largest sector, as it has a market share of 13.9%. It has observed several companies engaging in data analytics. India’s fintech market is the third largest in the world. Several players, such as PhonePe, Paytm, Policybazaar, and MobiKwik, have made use of data analytics and AI to grow their

**Chapter 11: BIBLIOGRAPHY**

1. GOGGLE CHROME
2. GITHUB
3. RESEARCHGATE
4. KAGGLE
5. MEDIUM
6. YOUTUBE CHANNELS
7. OPENAI
8. ANALYTICSVIDHYA.COM
9. IJRASET