DATA SCIENCE PROJECT :- CUSTOMER SEGMENTATION USING MACHINE LEARNING IN PYTHON

PROJECT REPORT

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (COMPUTER SCIENCE)

By

Dhanshree Rajput

Seat No.:- 452115

Under the esteemed guidance of

Mrs. Minakshi Dhande

Mrs.Sheetal Vekhande

DEPARTMENT OF COMPUTER SCIENCE

MODEL COLLEGE OF SCIENCE AND COMMERCE

(Affiliated to university of Mumbai)

KALYAN, 421 306

MAHARASHTRA

2020-2021

PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No:- **Roll no:-** 452115

Name of the Student :- Dhanshree B.Rajput

Title of the Project :- Stock Price Prediction using Machine Learning in Python

Name of the Guide: - Minakshi Dhande and Sheetal Vekhande

Is this your first submission? Yes No

Signature of the Student **Signature of the Guide**

Date: …… **Date:**.....

Signature of the Coordinator **Date:**……..

MODEL COLLEGE OF SCIENCE AND COMMERCE

(Affiliated to University Of Mumbai)

KALYAN-MAHARASHTRA-421 306

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the project entitled, “Stock Price PredictionUsing Machine Learning”, is Bonafede work of Dhanshree Rajput bearing

Seat No.: submitted in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE in COMPUTER SCIENCE from University of Mumbai.

Internal Guide Coordinator

External Examiner

ABSTRACT

A client is a critical aspect in the success of any organisation. The retention of an existing client base and the development of a new client base are crucial for a company's long-term success. This necessitates an understanding of client behaviour in respect to the firm. As a result, a company seeking a competitive advantage in the market must gain a 360-degree perspective of its clients using Python, ML. Customer Segmentation is one such strategy in this approach, which aids in identifying groups of similar consumers based on their interactions with the product and then effectively implementing various marketing strategies for the suitable consumers.

Customer segmentation is useful in understanding what demographic and psychographic sub-populations there are within your customers in a business case.

By understanding this, you can better understand how to market and serve them. This is similar and related but slightly different from the UX methodology of creating user personas: creating your ideal customers, their pain points, a defining quote, and so on, to understand their perspective.

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to Prof. K.S. BRAMHAWLE principal of MODEL COLLEGE OF SCIENCE AND COMMERCE for providing me the opportunity to do my project on “Customer Segmentation Using Machine Learning in Python”. I sincerely thank my project guide MRS. MINAKSHI DHANDE and MRS.SHEETAL VEKHANDE for guidance and encouragement in carrying out this project work. Special thanks to all the lab assistant for their seemingly small but valuable help for timely internet access and lab access. Last but not least, I wish to avail myself of this opportunity, express a sense of gratitude and love of my friends and my beloved parents for their manual support, strength and help for everything.

DECLARATION

I hereby declare that the project entitled, "Customer Segmentation Using Machine Learning in Python" done at college, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfilment of the requirements for the award of degree of BACHELOR OF SCIENCE (COMPUTER SCIENCE) to be submitted as final semester project as part of our curriculum.

DHANSHREE RAJPUT

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 What is Data Science ?

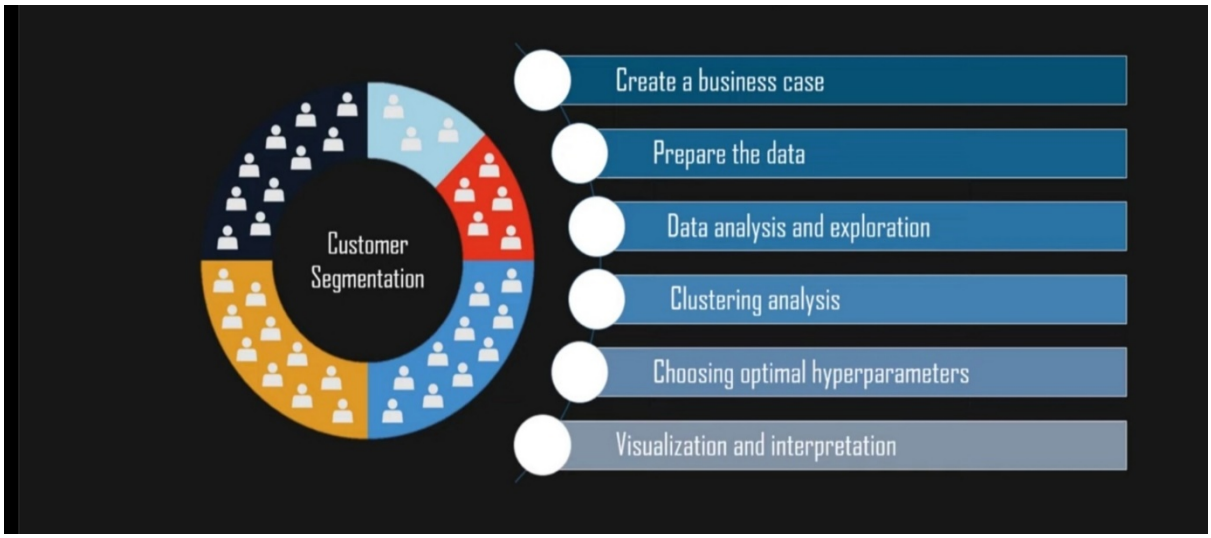
Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Python for Data Science

Python is open source, interpreted, high level language and provides great approach for object-oriented programming. It is one of the best language used by data scientist for various data science projects/application. Python provide great functionality to deal with mathematics, statistics and scientific function. It provides great libraries to deals with data science application.

One of the main reasons why Python is widely used in the scientific and research communities is because of its ease of use and simple syntax which makes it easy to adapt for people who do not have an engineering background.

1.2 What is Customer Segmentation ?



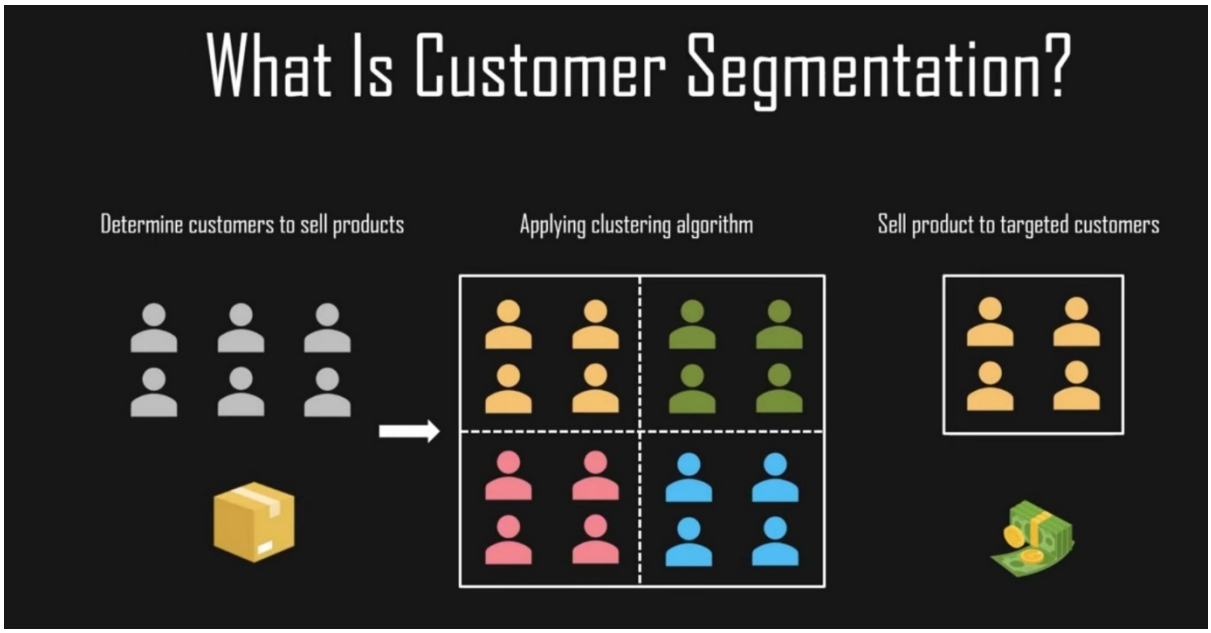
Customer Segmentation is an unsupervised method of targeting the customers in order to increase sales and market goods in a better way.

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

The data set 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 the 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.



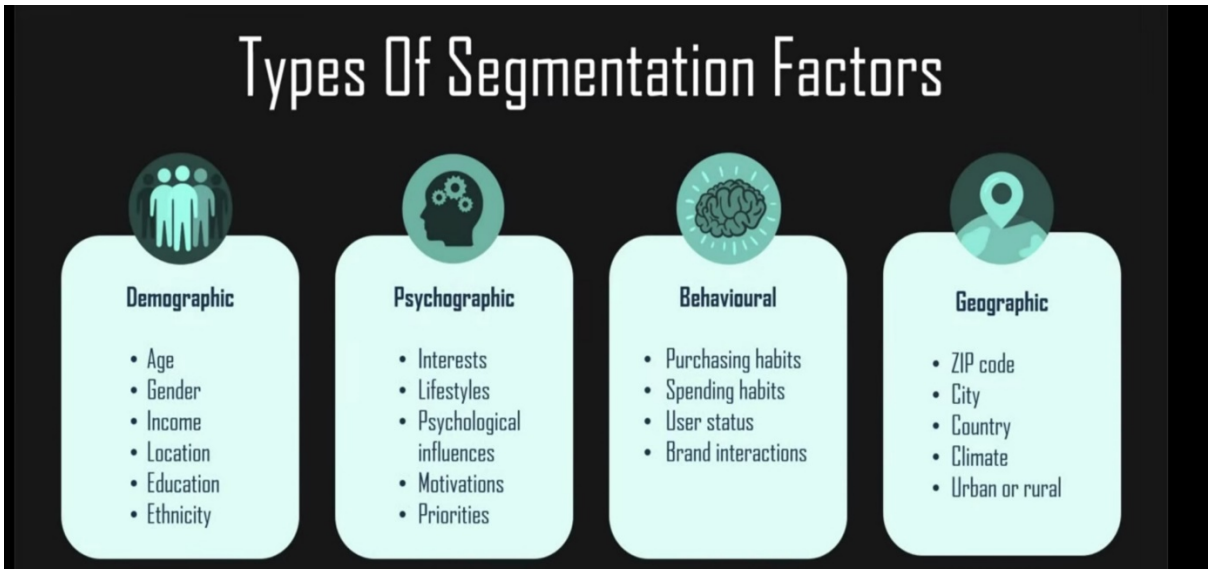
Customer Segmentation simply means grouping your customers according to various characteristics (for example grouping customers by age).

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".

1.3 Types of Segmentation Factors

There are different methodologies for customer segmentation, and they depend on four types of parameters:



1) **Demographic -** This segmentation is related to the structure, size, and movements of customers over space and time. Many companies use gender differences to create and market products. Parental status is another important feature. You can obtain data like this from customer surveys.

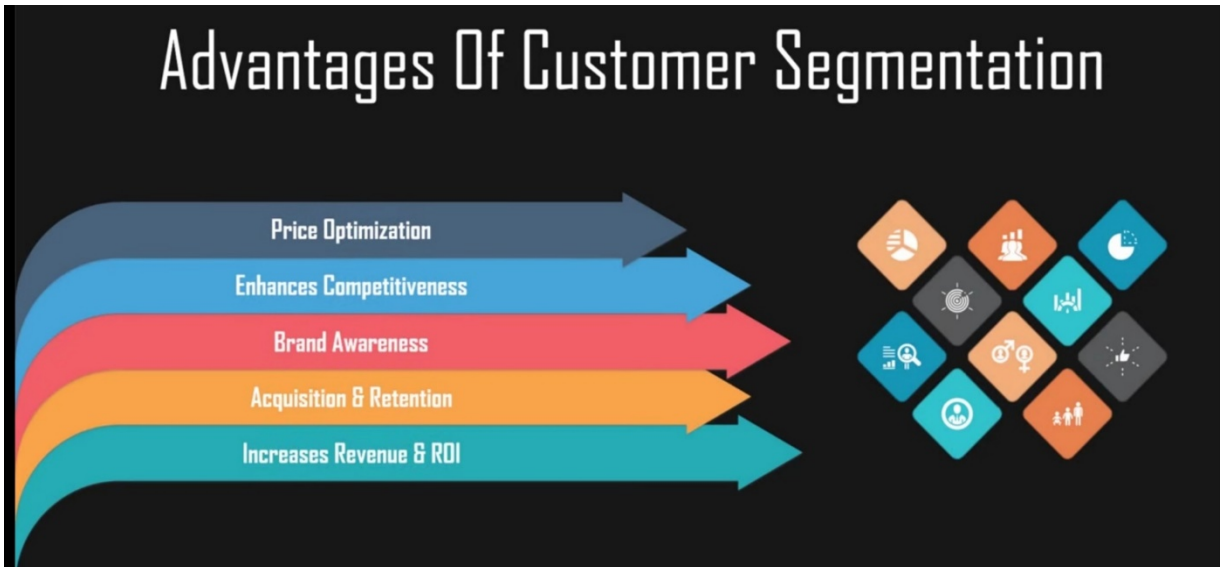
2) **Psychographic –** Psychographical segmentation of customers generally deals with things like personality traits, attitudes, or beliefs. This data is obtained using customer surveys, and it can be used to gauge customer sentiment.

3) **Behavioral -** This customer segmentation is based on past observed behaviors of customers that can be used to predict future actions. For example, brands that customers purchase, or moments when they buy the most. The behavioral aspect of customer segmentation not only tries to understand reasons for purchase but also how those reasons change throughout the year.

4) **Geographic -** This customer segmentation is very simple, it's all about the user's location. This can be implemented in various ways. You can group by country, state, city, or zip code.

1.4 Advantages of Customer Segmentation

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



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.

1.5 Machine Learning for customer segmentation

Machine learning methodologies are a great tool for analyzing customer data and finding insights and patterns. Artificially intelligent models are powerful tools for decision-makers. They can precisely identify customer segments, which is much harder to do manually or with conventional analytical methods.

There are many machine learning algorithms, each suitable for a specific type of problem**.** One very common machine learning algorithm that's suitable for customer segmentation problems is the k-means clustering algorithm**.** There are other clustering algorithms as well such as DBSCAN, Agglomerative Clustering, and BIRCH, etc

Why would you implement machine learning for 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

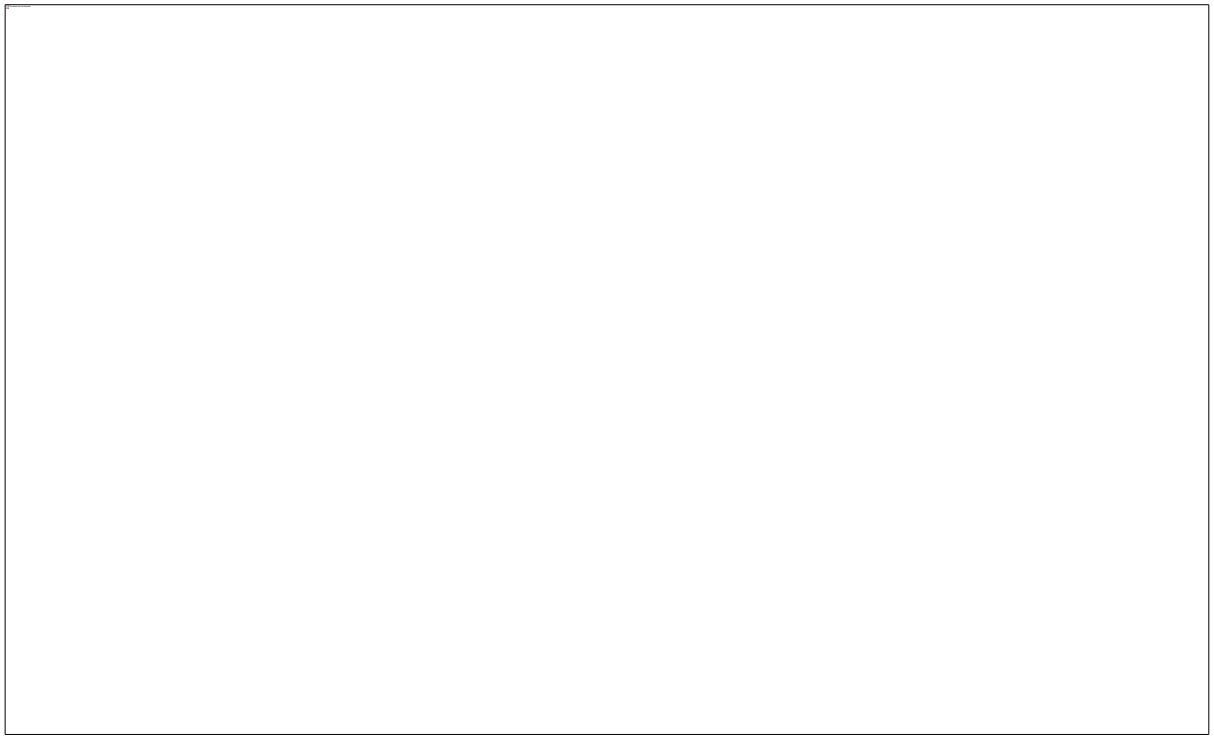
Machine learning models deployed in production support scalability, thanks to cloud infrastructure. These models are quite flexible for future changes and feedback. For example, consider a company that has 10000 customers today, and they've implemented a customer segmentation model. After a year, if 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

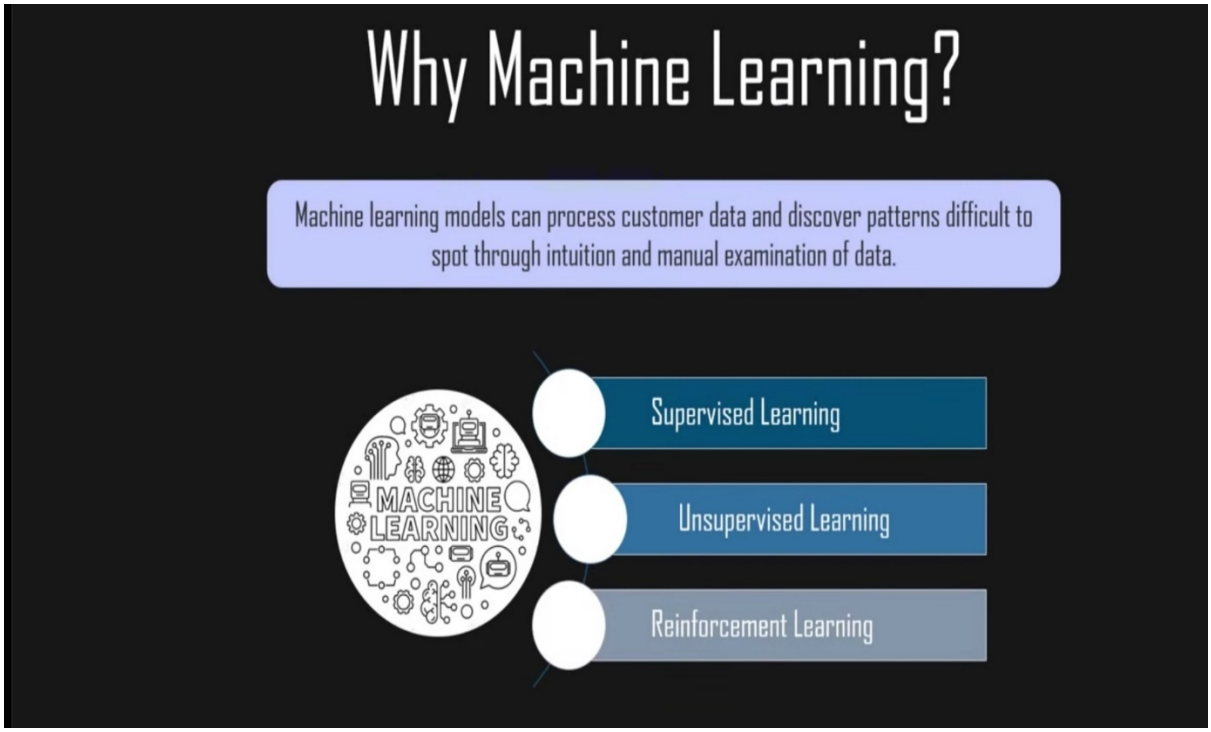
The value of an optimal number of clusters for given customer data is easy to find using machine learning methods like the elbow method. Not only the optimal number of clusters but also the performance of the model is far better when we use machine learning.

1.6 Why Machine Learning Important ?

Machine Learning is an application of Artificial Intelligence (AI) which enables a program(software) to learn from the experiences and improve itself at a task without being explicitly programmed. For example, how would you write a program that can identify fruits based on their various properties, such as color, shape, size, or any other



Machine Learning today has all the attention it needs. Machine Learning can automate many tasks, especially the ones that only humans can perform with their innate intelligence. Replicating this intelligence to machines can be achieved only with the help of machine learning.



1.7 Types of Machine Learning

- **Supervised learning :** In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.

- **Unsupervised learning :** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.

- **Reinforcement learning :** Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

1.8 How does supervised machine learning work?

Supervised machine learning requires the data scientist to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

- **Binary classification:** Dividing data into two categories.
- **Multi-class classification:** Choosing between more than two types of answers.
- **Regression modeling:** Predicting continuous values.
- **Ensembling:** Combining the predictions of multiple machine learning models to produce an accurate prediction.

1.9 How does unsupervised machine learning work?

Unsupervised machine learning algorithms do not require data to be labeled. They sift through unlabeled data to look for patterns that can be used to group data points into subsets. Most types of deep learning, including neural networks, are unsupervised algorithms. Unsupervised learning algorithms are good for the following tasks:

- **Clustering:** Splitting the dataset into groups based on similarity.
- **Anomaly detection:** Identifying unusual data points in a data set.
- **Association mining:** Identifying sets of items in a data set that frequently occur together.
- **Dimensionality reduction:** Reducing the number of variables in a data set.

1.9.1 How does reinforcement learning work?

Reinforcement learning works by programming an algorithm with a distinct goal and a prescribed set of rules for accomplishing that goal. Data scientists also program the algorithm to seek positive rewards -- which it receives when it performs an action that is beneficial toward the ultimate goal -- and avoid punishments -- which it receives when it performs an action that

gets it farther away from its ultimate goal. Reinforcement learning is often used in areas such as:

- **Robotics:** Robots can learn to perform tasks the physical world using this technique.
- **Video gameplay:** Reinforcement learning has been used to teach bots to play a number of video games.
- **Resource management:** Given finite resources and a defined goal, reinforcement learning can help enterprises plan out how to allocate resources.

1.9.2 OBJECTIVES

- Customer segmentation is the practice of dividing a company's customers into groups that reflect similarity among customers in each group.

- The goal of segmenting customers is to decide how to relate to customers in each segment in order to maximize the value of each customer to the business.

1.9.3 SCOPE

- Customer segmentation is the process of grouping customers together based on common characteristics.

- These customer groups are beneficial in marketing campaigns, in identifying potentially profitable customers, and in developing customer loyalty.

`

1.9.4 APPLICABILITY

Segmentation enables you to target customers with the highest potential value first, so you get the most out of your marketing budget. 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

1.9.5 PURPOSE OF THE PROJECT

Customer segmentation is the process by which you divide your customers up based on common characteristics – such as demographics or behaviors, so you can market to those customers more effectively.

These customer segmentation groups can also be used to begin discussions of building a marketing persona.

The purpose of segmenting customers is to determine how to correlate to customers in multiple segments to maximize customer benefits.

Perfectly done customer segmentation empowers marketers to interact with every customer in the best efficient approach.

CHAPTER 2 : REQUIREMENT SPECIFICATION

2.1 INTRODUCTION:

Customer segmentation is important for businesses to understand their target audience. Different advertisements can be curated and sent to different audience segments based on their demographic profile, interests, and affluence level.

There are many unsupervised machine learning algorithms that can help companies identify their user base and create consumer segments.

This algorithm can take in unlabelled customer data and assign each data point to clusters. The goal of K-Means is to group all the data available into non-overlapping sub-groups that are distinct from each other. That means each sub-group/cluster will consist of features that distinguish them from other clusters. K-Means clustering is a commonly used technique by data scientists to help companies with customer segmentation. It is an important skill to have, and most data science interviews will test your understanding of this algorithm/your ability to apply it to real life scenarios.

In this article, you will learn the following:

- Data pre-processing for K-Means clustering
- Building a K-Means clustering algorithm from scratch
- The metrics used to evaluate the performance of a clustering model
- Visualizing clusters built
- Interpretation and analysis of clusters built

2.2 HARDWARE SPECIFICATION:

Hardware choice is essential to the standard and potency of any software package.

In Hardware choice, size and power necessities are necessary.

Customer isolation will be with success run on the system with AN i3 processor with a minimum of four GB RAM and disc drive with 500GB and fifteen.6 inches to observe system performance. (Printer is needed for text output).

• Pentium processor -------- two GHz or on top of

• RAM capability -------- four GB

• Hard Disk -------- five hundred GB

2.3 SOFTWARE SPECIFICATION:

One of the foremost troublesome tasks is, software package choice, as long because the would like for the program is thought to search out out if a specific software package package fits the wants. once the primary choice of alternatives safety is needed to urge the need for a few software package compared to the opposite candidates. This section initial summarizes the application's question so proposes an in depth comparison.

- **Operating System :** Windows seven or ten
- **Software :** In this project, we have used Anaconda which has various in built software's like Spyder, R, PyCharm, Jupyter, and much more.

For this specific project, we have used Jupyter notebook to run the codes.

Anaconda is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment.

Databases :: Excel sheets

Front End :: Python

Python Libraries :: There are many libraries that go along with python few of them are Numpy, pandas,matplotlib In this project I have used few libraries in order to visualize the data in order to achieve better results.

Numpy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms basic linear algebra, basic statistical operations, random simulation and much more.

It is imported based on specific syntax like **import numpy as np**

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.

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.

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.

Seaborn helps you explore and understand your data.

It is imported as **import seaborn as sns**

CHAPTER 3 :ANALYSIS

3.1 FEASIBILITY STUDY:

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

3.1.1 Economic Feasibility:

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

3.1.2 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

3.1.3 Operational Feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 Software Specification:

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

3.2.1 Characteristics of Python:

In this project, I have used python with machine learning. Python is an interpreted high-level general-purpose programming language.

Python is an interactive and object-oriented scripting language.Python is designed to be highly readable.

It supports functional and structured programming methods as well as OOP. It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

CHAPTER 4 : LITERATURE SURVEY

4.1 Customer Classification

Over the years, the commercial world has become more competitive, as organizations such as these have to meet the needs and wants of their customers, attract new customers, and thus improve their businesses. The task of identifying and meeting the needs and requirements of each customer in the business is a very difficult task. This is because customers may vary according to their needs, wants, demographics, shapes, taste and taste, features and so on. As it is, it is a bad practice to treat all customers equally in business. This challenge has led to the adoption of the concept of customer segmentation or market segmentation, where consumers are divided into subgroups or segments where members of each subcategory exhibit similar market behaviors or features. Accordingly, customer segmentation is the process of dividing the market into indigenous groups.

4.2 Big Data

Recently, Big Data research has gained momentum. defines big data as - a term that describes a large number of formal and informal data, which cannot be analyzed using traditional methods and algorithms. Companies include billions of data about their customers, suppliers, and operations, and millions of internally connected sensors are sent to the real world on devices such as mobile phones and cars, sensing, creating, and communicating data. the ability to improve forecasting, save money, increase efficiency and improve decisionmaking in various fields such as traffic control, weather forecasting, disaster prevention, finance, fraud control, business transactions, national security, education, and healthcare. Big data is seen mainly in the three Vs namely: volume, variability and speed. There are other 2Vs available - authenticity and value, thus making it 5V.

4.3 Data Collection

Data collection is the process of collecting and measuring information against targeted variations in an established system, enabling one to answer relevant questions and evaluate

results. Data collection is part of research in all fields of study including physical and social sciences, humanities and business. The purpose of all data collection is to obtain quality evidence that allows analysis to lead to the creation of convincing and misleading answers to the questions submitted. We collected data from the UCI Machine Learning Repository.

4.4 Partition Method

Partition method is used to split the customers according to given attributes. This clustering method classifies the information into multiple groups based on the characteristics and similarity of the data. Its the data analysts to specify the number of clusters that has to be generated for the clustering methods.

In the partitioning method when database(D) that contains multiple(N) objects then the partitioning method constructs user-specified(K) partitions of the data in which each partition represents a cluster and a particular region. There are many algorithms that come under partitioning method some of the popular ones are K-Mean, PAM(K-Mediods), CLARA algorithm (Clustering Large Applications) etc.

4.5 K-Means Clustering

K-Means clustering is an unsupervised machine learning algorithm that divides the given data into the given number of clusters. Here, the “K” is the given number of predefined clusters, that need to be created.

It is a centroid based algorithm in which each cluster is associated with a centroid. The main idea is to reduce the distance between the data points and their respective cluster centroid.

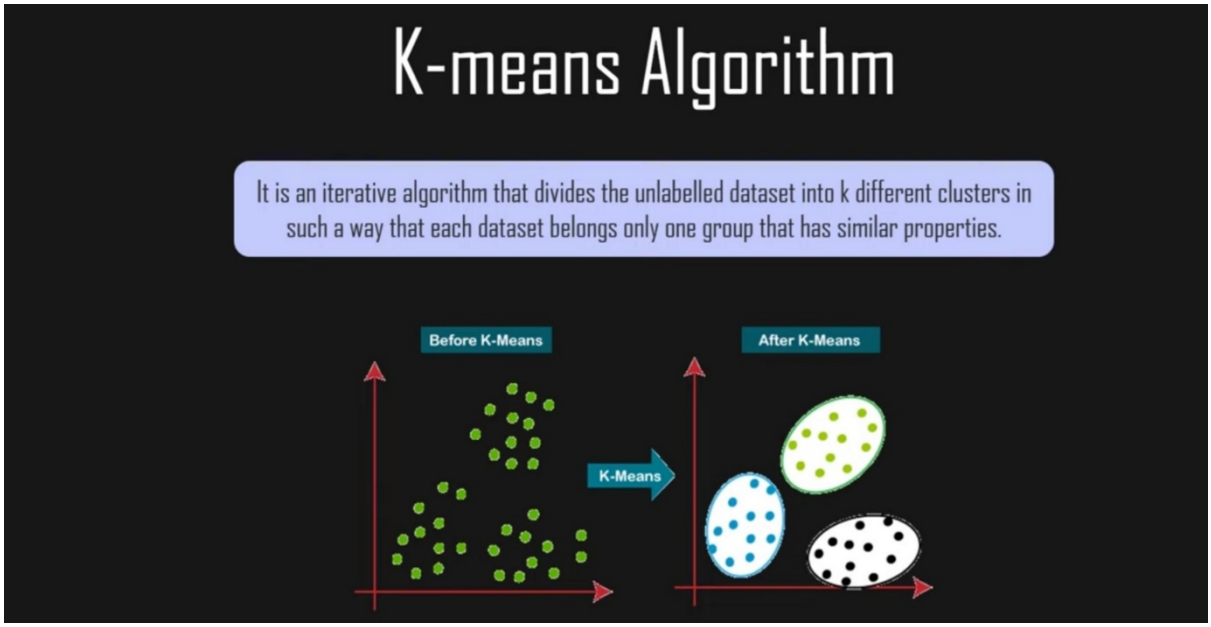
The algorithm takes raw unlabelled data as an input and divides the dataset into clusters and the process is repeated until the best clusters are found.

K-Means is very easy and simple to implement. It is highly scalable, can be applied to both small and large datasets. There is, however, a problem with choosing the number of clusters

or K. Also, with the increase in dimensions, stability decreases. But, overall K Means is a simple and robust algorithm that makes clustering very easy.

4.6 K-Means Algorithm

Cluster the data using the k-means method, and the number of clusters (k) we want in the final result is the first step. The first step in the cluster creation procedure is to randomly choose k objects from the dataset to be the initial cluster centres. The objects you picked are also known as cluster means, sometimes known as centroids. The objects that remain are then all given to the centroid that is the closest to it. Euclidean the closer an item is to the cluster mean, the more likely it is to be drawn from that cluster. In this stage, we have a reference to the "cluster assignment". The process is implemented to automatically adjust the mean value for each cluster in the data when the assignment is done. Once the recalculation of the centres is complete, the observations are reviewed to see whether they are now closer to a different cluster. This modification to the cluster mean is used to reallocate the objects. This is performed until the cluster assignments aren't changing anymore. Similar to the clusters detected in the previous iteration, the new clusters found in this iteration are the same



Summing up the K-means clustering –

- The number of clusters we need to construct is specified.
- The method chooses k items from the dataset at random. The starting cluster or mean is this item.A new observation is assigned to the centroid that is closest to it. In this assignment, the Euclidean distance between the item and the centroid is taken into consideration.
- For each of the cluster's data points, a new mean value is calculated by multiplying the previous mean value by the total number of data points belonging to that cluster. The centroid of the kth cluster has a length of p and contains the means of all of the variables for the kth cluster's observations. This expression is known as the p expression and represents the number of variables.
- Within the sum of squares, iterative minimization of the total. The assignment will then cease wagging when we reach maximum iteration via iterative minimization of the total sum of the square.

- **eps**: The distance that specifies the neighborhoods. Two points are considered to be neighbors if the distance between them are less than or equal to eps.
- **minPts:** Minimum number of data points to define a cluster.

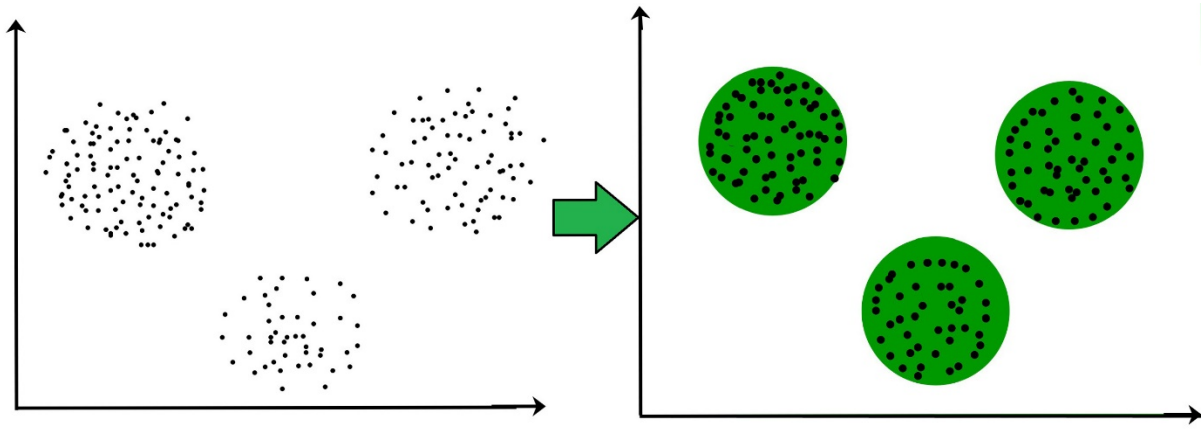
4.7 Clustering in Machine Learning

4.7.1. Introduction to Clustering

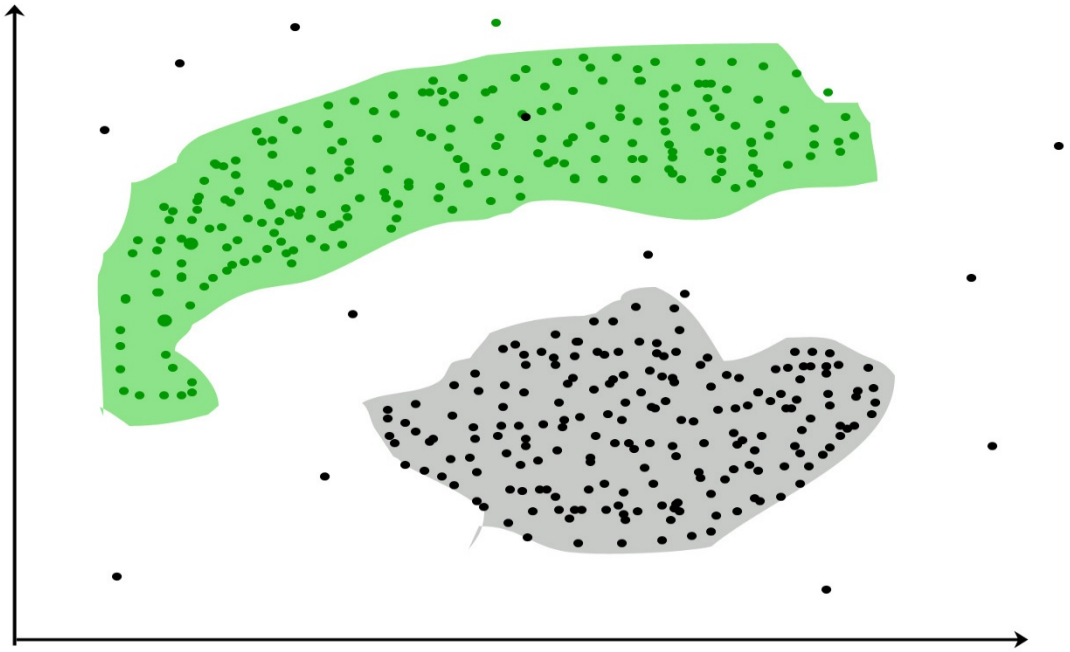
It is basically a type of unsupervised learning method. An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

For ex– The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



It is not necessary for clusters to be spherical. Such as :



4.7.2.Uses of Clustering

Marketing:

In the field of marketing, clustering can be used to identify various customer groups with existing customer data. Based on that, customers can be provided with discounts, offers, promo codes etc.

Real Estate:
Clustering can be used to understand and divide various property locations based on value and importance. Clustering algorithms can process through the data and identify various groups of property on the basis of probable price.

BookStore and Library management:
Libraries and Bookstores can use Clustering to better manage the book database. With proper book ordering, better operations can be implemented.

Document Analysis:
Often, we need to group together various research texts and documents according to similarity. And in such cases, we don't have any labels. Manually labelling large amounts of data is also not possible. Using clustering, the algorithm can process the text and group it into different themes.

4.7.3. Clustering Methods :

- **Density-Based Methods:** These methods consider the clusters as the dense region having some similarities and differences from the lower dense region of the space. These methods have good accuracy and the ability to merge two clusters. Example DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points to Identify Clustering Structure), etc.

- **Hierarchical Based Methods:** The clusters formed in this method form a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one. It is divided into two category

    - **Agglomerative** (bottom-up approach)
    - **Divisive** (top-down approach)

examples CURE (Clustering Using Representatives), BIRCH (Balanced Iterative Reducing Clustering and using Hierarchies), etc.

- **Partitioning Methods:** These methods partition the objects into k clusters and each partition forms one cluster. This method is used to optimize an objective criterion similarity function such as when the distance is a major parameter example K-means, CLARANS (Clustering Large Applications based upon Randomized Search), etc.

- **Grid-based Methods:** In this method, the data space is formulated into a finite number of cells that form a grid-like structure. All the clustering operations done on these grids are fast and independent of the number of data objects example STING (Statistical Information Grid), wave cluster, CLIQUE (CLustering In Quest), etc.

- **Elbow Method :** The elbow approach is a heuristic approach that serves the purposes of evaluation and confirmation of cluster analysis consistency in order to discover the best number of clusters, the dataset is examined. This method is used in order to get the percentage of variation explained with respect to the number of clusters: This will be the case when we have enough clusters that the addition of another cluster does not drastically improve data modelling. When the number of clusters is plotted against the percentage of variation explained by the clusters, the early clusters will offer a lot of information (provide a lot of variation), but the marginal increase will level out at some point, resulting in a concave curve in the graph. As the elbow criteria has been set, the number of clusters to be picked is referred to as the "elbow".

4.8 Machine Learning techniques are broadly divided into two parts :

1. Supervised Machine Learning
2. Unsupervised Machine Learning

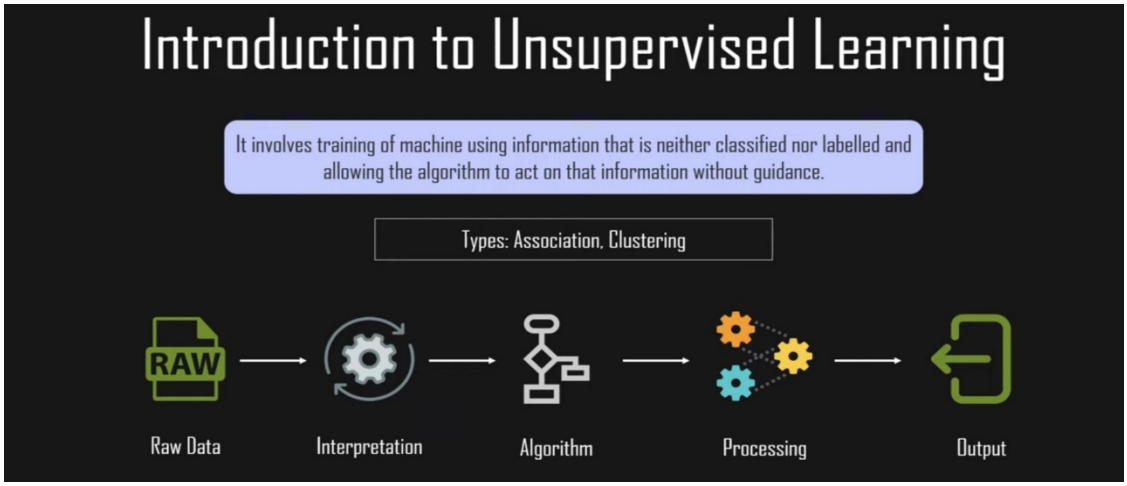
In Supervised Machine Learning, the data is labelled and the algorithm learns from labelled training data. Examples of this method are Classification and Regression.

In Unsupervised Machine Learning, we do not need to supervise the model. Such a method deals with unlabelled data. Unsupervised machine learning helps us find hidden and unknown patterns in data.

Often it easier to get unlabelled data as compared to labelled data, and in such cases, we can use unsupervised machine learning to work on the data. Data, which needs categorization can be categorized with the help of unsupervised machine learning.

Clustering is a type of unsupervised machine learning in which the algorithm processes our data and divided them into “clusters”.

4.8.1 Unsupervised Learning :



Unsupervised learning is a machine learning technique in which models are not supervised using a training dataset. Instead, models themselves find the hidden patterns and insights from the given data. It can be compared to learning which takes place in the human brain while learning new things. Unsupervised learning cannot be directly applied to a

regression or classification problem because, unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of the dataset, group that data according to similarities, and represent that dataset in a compressed format.

4.9 METHODOLOGY

- Create a business case.
- Prepare the data.
- Data analysis and exploration
- Clustering analysis
- Choosing optimal Hyper Parameters.
- Visualization and interpretation.

CHAPTER 5 :SCRIPTS AND STEPS

Mall Customer Data: Implementation of K-Means in Python

Mall Customer data is an interesting dataset that has hypothetical customer data. It puts you in the shoes of the owner of a supermarket. You have customer data, and on this basis of the data, you have to divide the customers into various groups.

The data includes the following features :

1. Customer ID

2. Customer Gender

3. Customer Age

4. Annual Income of the customer (in Thousand Dollars)

5. Spending score of the customer (based on customer behaviour and spending nature)

Data

This project is a part of the Mall Customer Segmentation Data competition held on Kaggle.

The dataset can be downloaded from the kaggle website which can be found here.

Let us proceed with the code

Import the libraries

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

The necessary libraries are imported.

```
In [2]: df=pd.read_csv("Mall_Customers.csv")
        df.head()
```

The data is read.

So let us have a look at the data.

Out[2]:

	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

Shape() function returns the number of rows and columns and df

```
In [3]: df.shape
Out[3]: (200, 5)
```

describe() function summarize different columns of the data set and finding out aggrigate measures like mean,count,standard deviation, maximum and minimum values etc.

```
In [4]: df.describe()
```

Out[4]:

	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

Now, using the **dtypes ()** function we get to now the datatypes of various columns .

```
In [5]: df.dtypes

Out[5]: CustomerID                int64
        Gender                   object
        Age                       int64
        Annual Income (k$)        int64
        Spending Score (1-100)    int64
        dtype: object
```

isnull().sum() function gives us a total count of the null values and each column and here we see that there are null .so, there is no processing required with respect to that.

```
In [6]: #Checking for null values
        df.isnull().sum()

Out[6]: CustomerID              0
        Gender                  0
        Age                     0
        Annual Income (k$)      0
        Spending Score (1-100)  0
        dtype: int64
```

Since we do not required the customerid column for clustering.we proceed to remove that from the dataset using the **drop()** function giving customerid as a parameter.

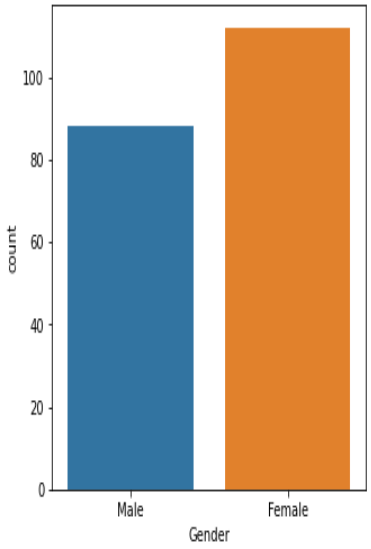
```
In [7]: #axis = 1 -> refers to column,we want to drop customer id columns as it has no use in the analysis
        df.drop(["CustomerID"],axis=1,inplace=True)

In [8]: df.head()
```

Out[8]:

	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

Our next step will be knowing more about the distribution of customers that we have ,where is visualization are used for this purpose. Through the **countplot** we get the count of the male and female customers in the given data . to get the distribution with respect to the age .

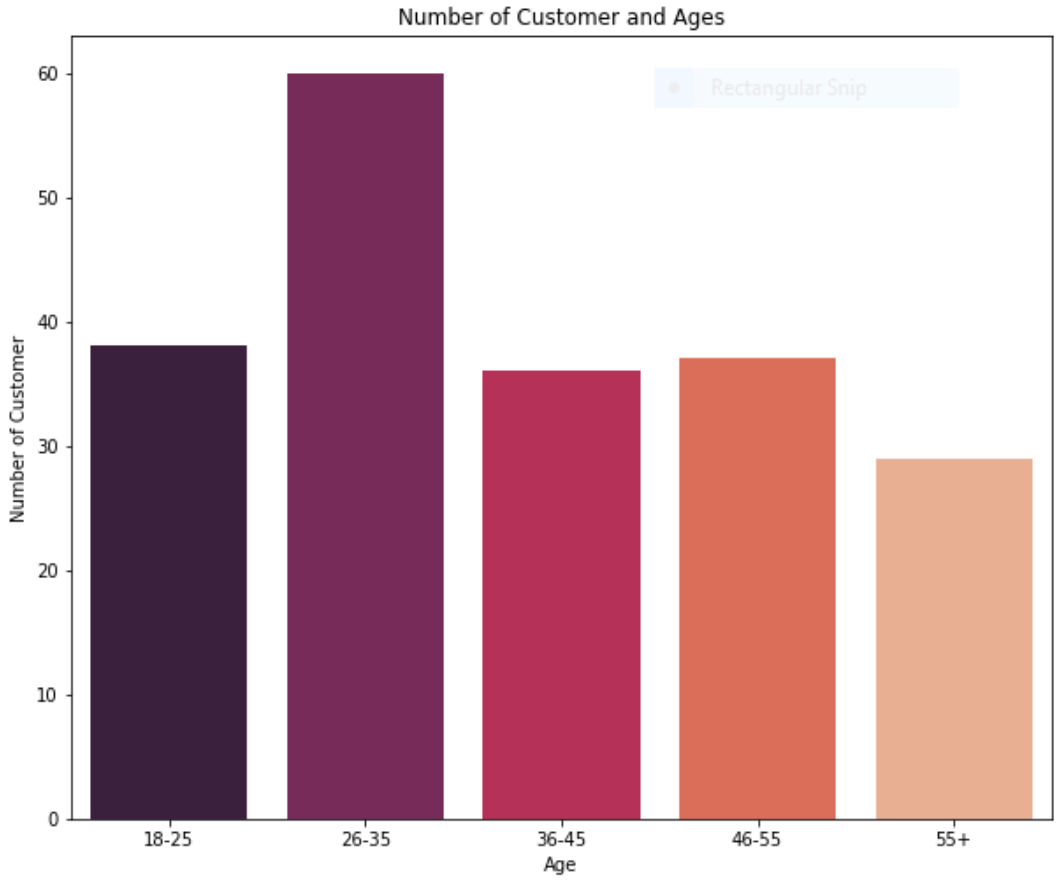
In [9]:
```
# Gender Distribution
# default theme is darkgrid,which can be changeed using sns.set_style
# to find the count of unique classes in categorical data, we use count plot (modified version of bar plot)
plt.figure(figsize=(5,5)) # width,height in inches
sns.countplot(data =df , x = 'Gender' )
plt.show()
# To create horizontal plot we map the categorical variable to the y instead of x, i.e just specify the y
```



We break the given age values into various categories and we find out how many customer lie in each category using bar plot.

In [10]:
```
age_18_25 = df.Age[(df.Age >= 18) & (df.Age <= 25)]
age_26_35 = df.Age[(df.Age >= 26) & (df.Age <= 35)]
age_36_45 = df.Age[(df.Age >= 36) & (df.Age <= 45)]
age_46_55 = df.Age[(df.Age >= 46) & (df.Age <= 55)]
age_55above = df.Age[df.Age >= 56]

x = ["18-25","26-35","36-45","46-55","55+"]
y = [len(age_18_25.values),len(age_26_35.values),len(age_36_45.values),len(age_46_55.values),len(age_55above.values)]
```

In [11]:
```
plt.figure(figsize=(10,8))
sns.barplot(x=x, y=y, palette="rocket")
plt.title("Number of Customer and Ages")
plt.xlabel("Age")
plt.ylabel("Number of Customer")
plt.show()
```

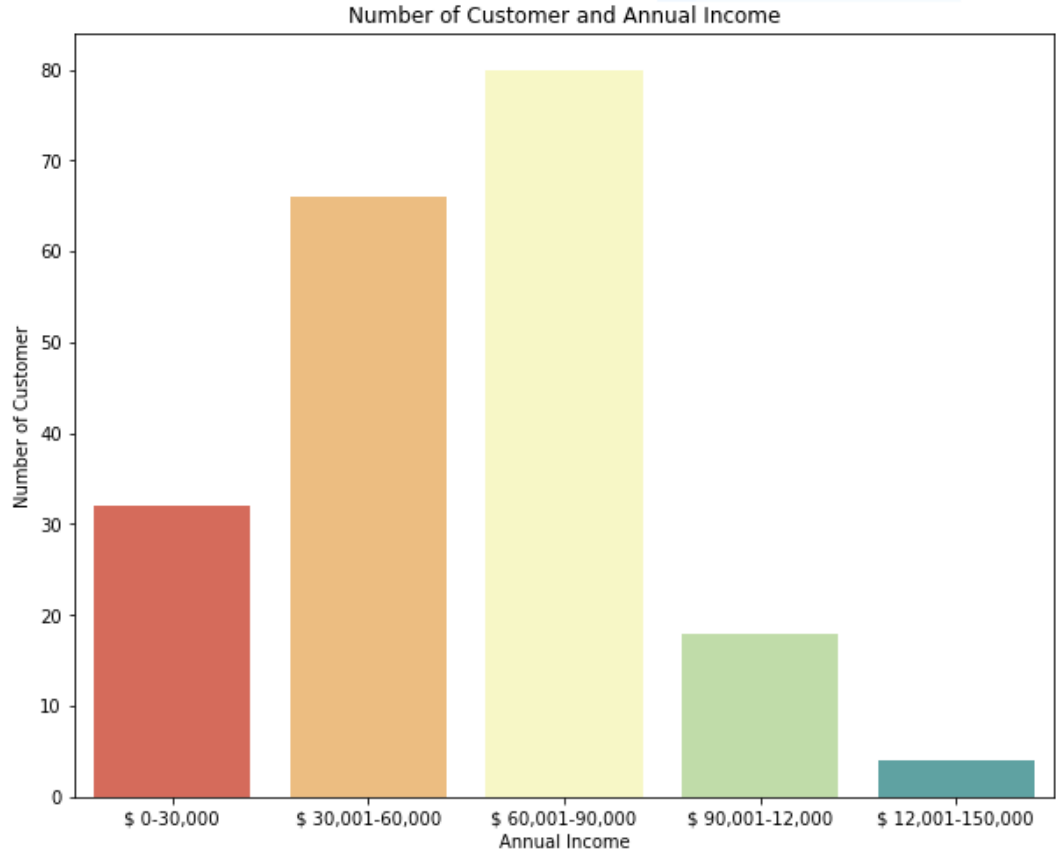


The same is done with the Annual Income column and Spending Score column.

In [12]:
```
#violin plot (combination of density plot and boxplot)(But has more noise than boxplot)(spending score distribution over gender)
```

In [13]:
```
# Age distribution bar plot
ai0_30 =df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual Income (k$)"] <= 30)]
ai31_60 =df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 31) & (df["Annual Income (k$)"] <= 60)]
ai61_90 =df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 61) & (df["Annual Income (k$)"] <= 90)]
ai91_120 =df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 91) & (df["Annual Income (k$)"] <= 120)]
ai121_150 =df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 121) & (df["Annual Income (k$)"] <= 150)]

aix = ["$ 0-30,000", "$ 30,001-60,000", "$ 60,001-90,000", "$ 90,001-12,000", "$ 12,001-150,000"]
aiy = [len(ai0_30.values), len(ai31_60.values), len(ai61_90.values), len(ai91_120.values), len(ai121_150.values)]
```

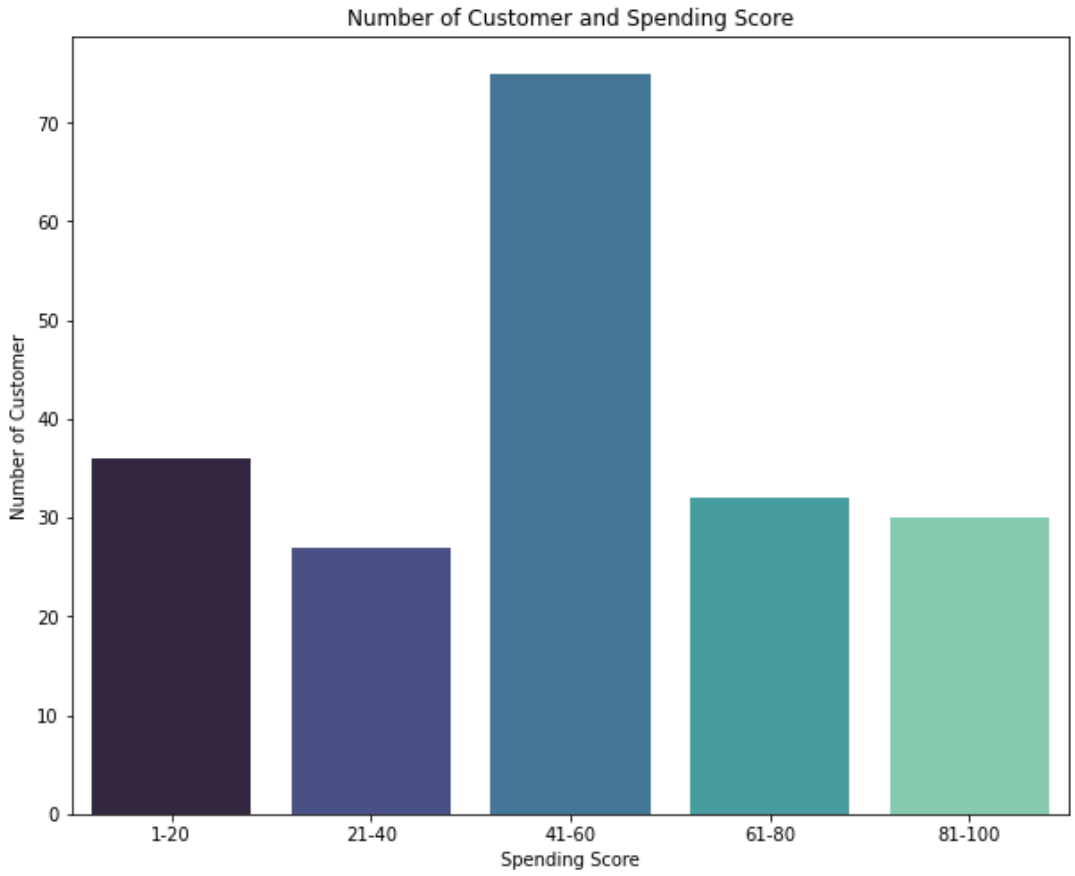
In [14]:
```
plt.figure(figsize=(10,8))
sns.barplot(x=aix, y=aiy, palette="Spectral")
plt.title("Number of Customer and Annual Income")
plt.xlabel("Annual Income")
plt.ylabel("Number of Customer")
plt.show()
```



Spending Score

In [15]:
```
# No of customers vs Spending Score
# age distribution bar  plot
ss_1_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Spending Score (1-100)"] <= 20)]
ss_21_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) & (df["Spending Score (1-100)"] <= 40)]
ss_41_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) & (df["Spending Score (1-100)"] <= 60)]
ss_61_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) & (df["Spending Score (1-100)"] <= 80)]
ss_81_100 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 81) & (df["Spending Score (1-100)"] <= 100)]

ssx = ["1-20","21-40","41-60","61-80","81-100"]
ssy = [len(ss_1_20.values),len(ss_21_40.values),len(ss_41_60.values),len(ss_61_80.values),len(ss_81_100.values)]
```

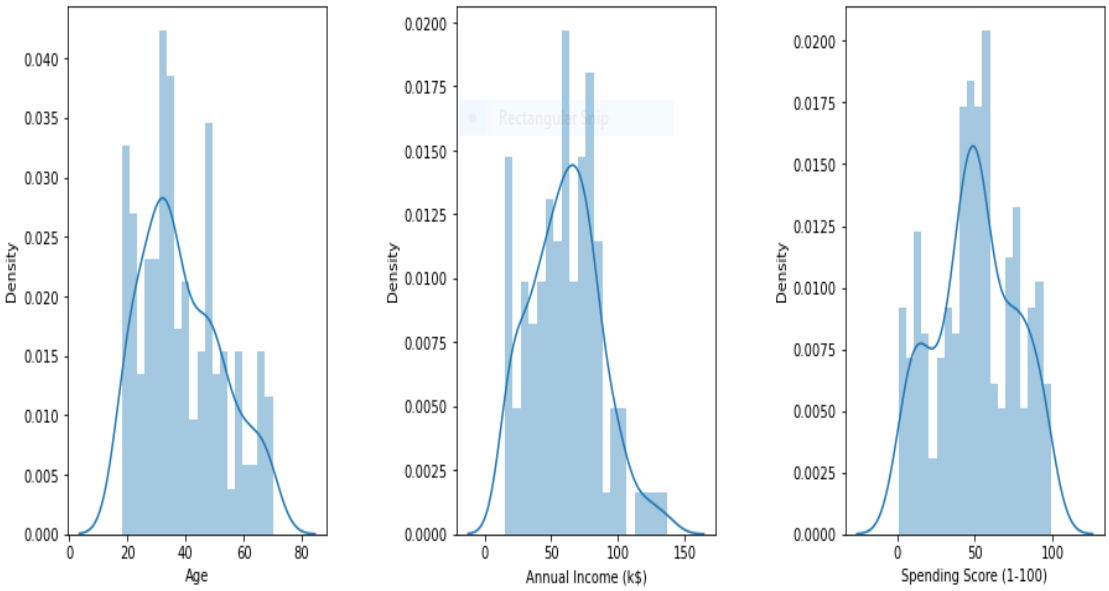
In [16]:
```
plt.figure(figsize=(10,8))
sns.barplot(x=ssx, y=ssy, palette="mako")
plt.title("Number of Customer and Spending Score")
plt.xlabel("Spending Score")
plt.ylabel("Number of Customer")
plt.show()
```



After this we use **distplot()** function over the columns which provide visual distribution that combines the matplotlib **dist()** function with the seaborn **kdeplot()** function(kernel density estimate) . and **regplot ()** function (used to plot data and a linear regression model fit).

Since this functions is now depreciated an alternative method would be to used displot or histplot function which provide similar functionality and flexibility.

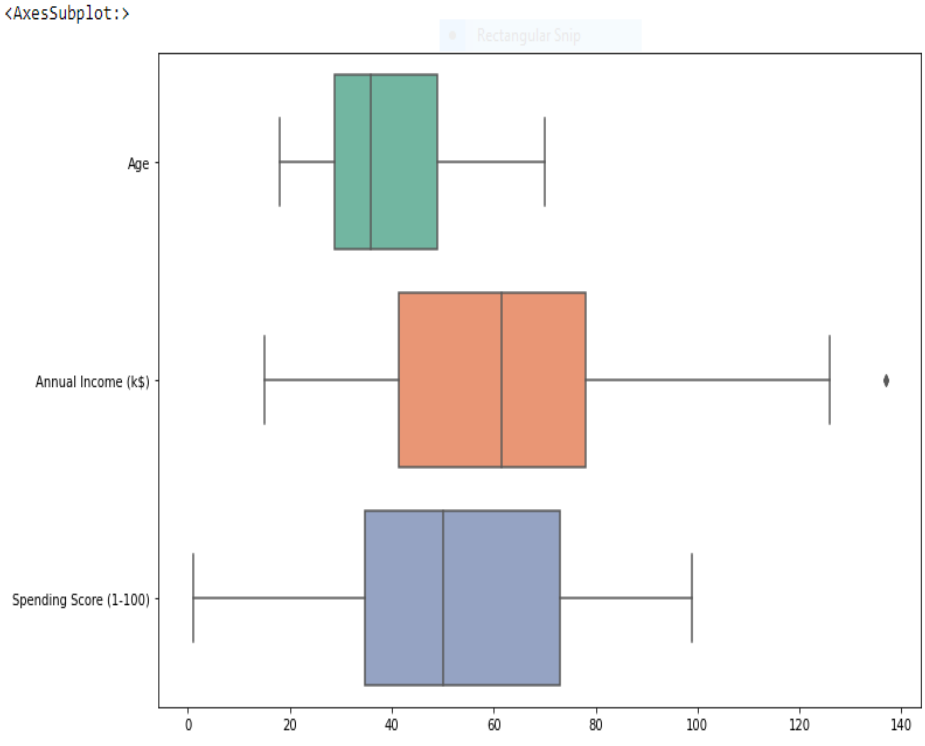
```
In [19]: # density plot for age,annual income , spending score
         plt.figure(1, figsize=(15,6))
         n=0
         for i in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
             n+=1
             plt.subplot(1, 3, n)
             plt.subplots_adjust(hspace =0.5, wspace=0.5)
             sns.distplot(df[i], bins = 20)
             #plt.title("Distplot of { }" , format(i))
         plt.show()
```



After this we used the boxplot function to get the distribution of the data points across quartiles of each column and observing this we see other that data set is nearly freeof outliers which should be points which are alying beyond 1.5 times of the first and third quartile .

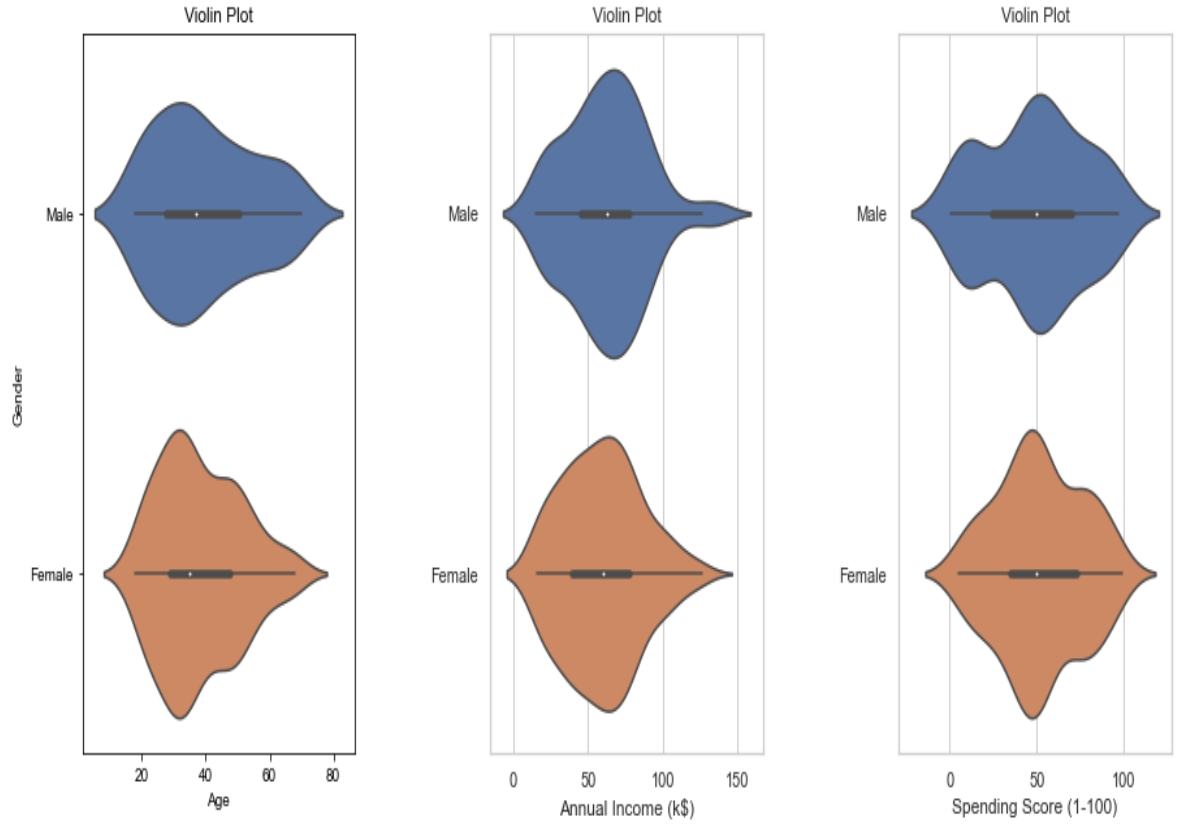
```
In [20]: fig, ax = plt.subplots()
         fig.set_size_inches(11.7, 8.27)
         sns.boxplot(data=df, orient="h", palette="Set2", ax=ax) #just ax = sns(....)yeilds a small figure size
```

Out[20]: <AxesSubplot:>



After this we used a violin plot to find the distribution of Age , Annual Income and Spending Score across classes Male and Female of the categorical variable Gender .

```
In [22]: #violin plot - A combination of density plot and box plot. But violin plot tends to be more noisy then boxplot

         plt.figure(1,figsize=(15,7))
         n=0
         for cols in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
             n+=1
             plt.subplot(1, 3, n)
             sns.set(style="whitegrid")
             plt.subplots_adjust(hspace =0.5, wspace=0.5)
             sns.violinplot(x = cols , y = 'Gender', data = df )
             plt.ylabel('Gender' if n == 1 else '')
             plt.title('Violin Plot' )
         plt.show()
```



Now we import the K-Means module from the sklearn.cluster module and come to the 2D implementation of the KMean clustering algorithm . for that we need to choose any two attributes to cluster.

And here I have choosing the Annual income and Spending Score .others can also be choosing but I got the best result with this tools.

Its all experimentations.

```
In [23]: # Annual Income (k$)  Spending Score (1-100)
         x1=df.loc[:, ["Age", "Spending Score (1-100)"]].values
         # from sklearn.cluster import KMeans
         # sc = StandardScaler()
         # x = sc.fit_transform(x)
```

```
In [28]: from sklearn.cluster import KMeans
         from sklearn.cluster import KMeans
         wcss = []
         for i in range(1,11):
             km=KMeans(n_clusters=i)
             km.fit(x1)
             wcss.append(km.inertia_)
```

So one of the prerequisite of the kmean clustering method is to find out k or the number of clusters that we need to give as a parameter to the model. For these we can either use domain specific intuition or if one as workdown similar project that value can also be used.but the most versatile method for this is the Elbow method . which inovolves running a loop covering all the key values and

Plotting a graph of within clusters sum of squares against that particular k value . the k at which the graph forms an elbow and appears to be steadily heading towards the zero is a recommended value of k that we should be using. So when we try the elbow method for Annual Income and Spending Score

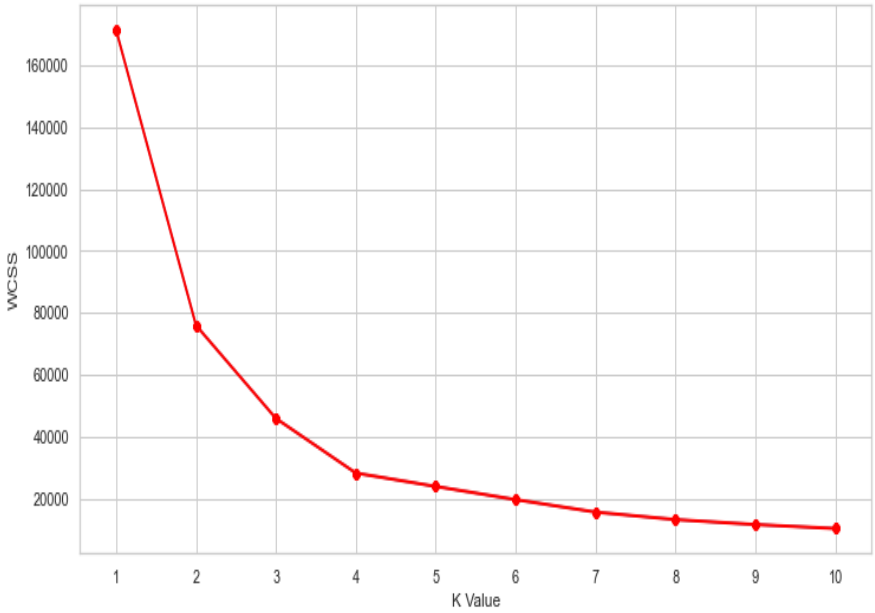
We get k=5 as we see in the graph .we apply the K-means clustering algorithm and save the obtain cluster value in the separate attribute called labels in the data set itself.

The Elbow Method - Calculate the Within Cluster Sum of Squared Errors (WSS) for different values of k, and choose the k for which WSS first starts to diminish. In the plot of WSS-versus k, this is visible as an elbow.

The steps can be summarized in the below steps:

1. Compute K-Means clustering for different values of K by varying K from 1 to 10 clusters.
2. For each K, calculate the total within-cluster sum of square (WCSS).
3. Plot the curve of WCSS vs the number of clusters K.
4. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters

In [30]:
```
#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 optimal K value is found to be 5 using the elbow method.

Finally I made a 3D 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 colours as shown in the 3D plot.

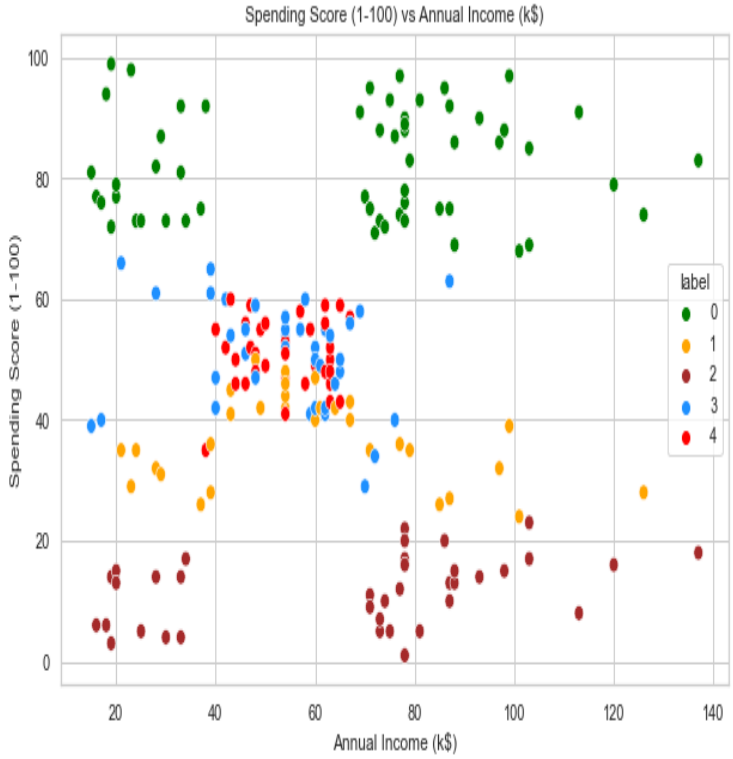
In [33]:
```
#Taking 5 clusters
km1=KMeans(n_clusters=5)
#Fitting the input data
km1.fit(x1)
#predicting the labels of the input data
y=km1.predict(x1)
#adding the labels to a column named label
df["label"] = y
#The new dataframe with the clustering done
df.head()
```

Out[33]:

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

And then we plot the clusters using the scatter plot function of the matplotlib.and we observe that the distinct clusters are formed using the given colors.

```
In [35]: #Scatterplot of the clusters
plt.figure(figsize=(10,6))
sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',hue="label",
                palette=['green','orange','brown','dodgerblue','red'], legend='full',data = df  ,s = 60 )
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```



We can clearly see that 5 different clusters have been formed from the data. The red cluster is the customers with the least income and least spending score, similarly, the blue cluster is the customers with the most income and most spending score.

k-Means Clustering on the basis of 3D data

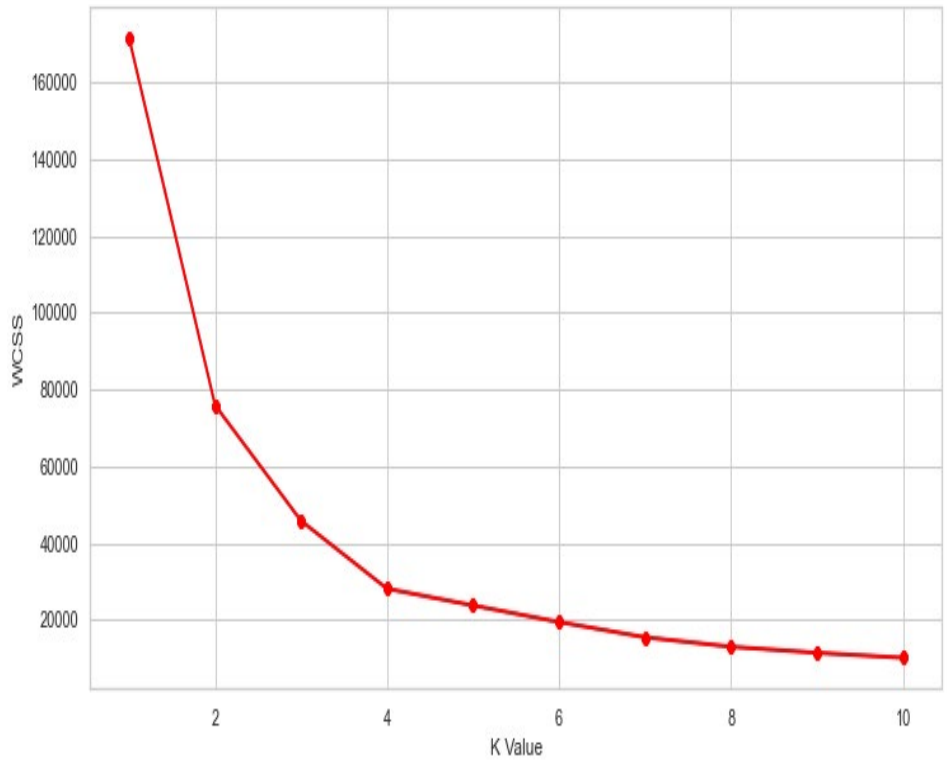
Now, we shall be working on 3 types of data. Apart from the spending score and annual income of customers, we shall also take in the age of the customers.

```
In [38]: #3 Dimensional clustering

         x4=df.loc[:, ["Age", "Spending Score (1-100)"]].values
         wcss = []
         for i in range(1,11):
             km = KMeans(n_clusters=i)
             km.fit(x4)
             wcss.append(km.inertia_)
         # elbow method
         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.ylabel("WCSS")
         plt.show()
```

The WCSS curve.

Next I plotted Within Cluster Sum Of Squares (WCSS) against the the number of clusters (K Value) to figure out the optimal number of clusters value. WCSS measures sum of distances of observations from their cluster centroids which is given by the below formula

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



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

In [41]:
```
#We choose the k for which WSS starts to diminish
km2 = KMeans(n_clusters=5)
y2 = km.fit_predict(x4)
df["label"] = y2
#The data with labels
df.head()
```

Out[41]:

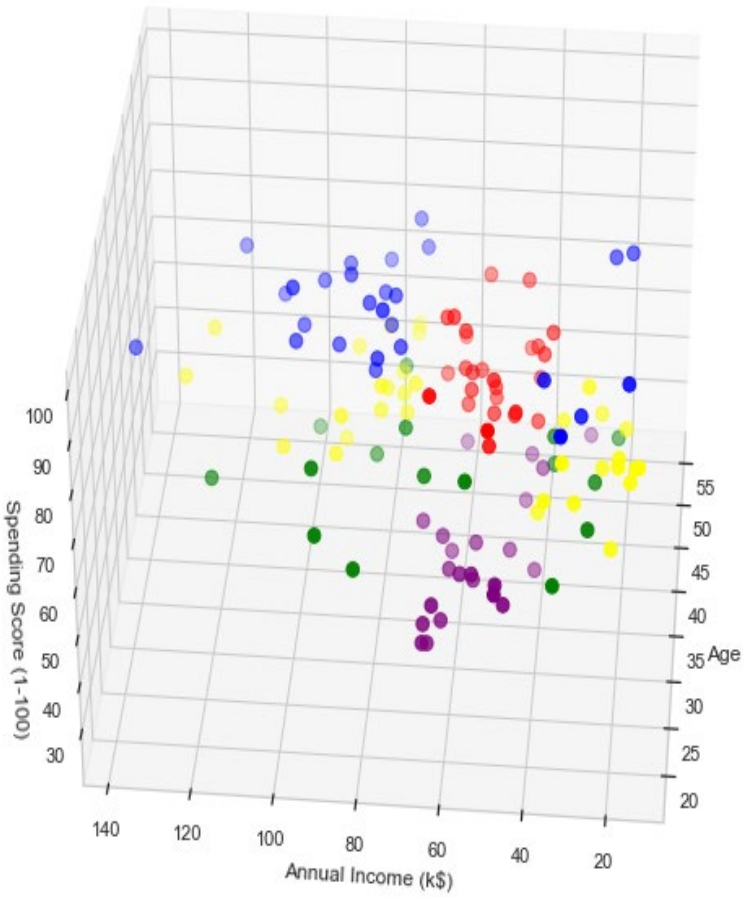
	Gender	Age	Annual Income (k$)	Spending Score (1-100)	label
0	Male	19	15	39	8
1	Male	21	15	81	4
2	Female	20	16	6	6
3	Female	23	16	77	4
4	Female	31	17	40	8

Similarly we can also apply a 3D version of k means by taking Age,Annual income and Spending score and cluster them in the same way as we have done overhere.

Now we plot it.

In [47]:
```
#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(df.Age[df.label == 0], df["Annual Income (k$)"]
           [df.label == 0], df["Spending Score (1-100)"][df.label == 0], c='purple', s=60)
ax.scatter(df.Age[df.label == 1], df["Annual Income (k$)"]
           [df.label == 1], df["Spending Score (1-100)"][df.label == 1], c='red', s=60)
ax.scatter(df.Age[df.label == 2], df["Annual Income (k$)"]
           [df.label == 2], df["Spending Score (1-100)"][df.label == 2], c='blue', s=60)
ax.scatter(df.Age[df.label == 3], df["Annual Income (k$)"]
           [df.label == 3], df["Spending Score (1-100)"][df.label == 3], c='green', s=60)
ax.scatter(df.Age[df.label == 4], df["Annual Income (k$)"]
           [df.label == 4], df["Spending Score (1-100)"][df.label == 4], c='yellow', s=60)
ax.view_init(35, 185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```

The output:


Spending Score (1-100)
100
90
80
70
60
50
40
30
140
120
100
80
60
40
20
Annual Income (k$)
55
50
45
40
35
30
25
20
Age

CHAPTER 6 : CONCLUSION

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 application of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.

So, we used K-Means clustering to understand customer data. K-Means is a good clustering algorithm. Almost all the clusters have similar density. It is also fast and efficient in terms of computational cost.

CHAPTER 7 : REFERENCES

- Analytics Community | Analytics Discussions | Big Data Discussion (analyticsvidhya.com)
- K-Means clustering with Mall Customer Segmentation - Analytics Vidhya
- Machine Learning and Data Science - GeeksforGeeks
- **Mall_Customers.csv - Jupyter Text Editor**
- Python RFM (Recency, Frequency, Monetary) Analysis for Customer Segmentation - DataCamp
- K-Means Clustering Algorithm - Javatpoint
- Clustering in Machine Learning - Nixus (nixustechnologies.com)