

AN INTERNSHIP PROJECT REPORT

on

CUSTOMER SEGMENTATION

Submitted in partial fulfillment of the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

**LALITHA HARINI CH
(20021A0503)**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING KAKINADA

Jawaharlal Nehru Technological University Kakinada

Kakinada-533003, Andhra Pradesh, INDIA

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY COLLEGE OF ENGINEERING KAKINADA
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA
KAKINADA – 533003, ANDHRA PRADESH, INDIA**



CERTIFICATE

This is to certify that this project report entitled “**CUSTOMER SEGMENTATION**” is a bonafide record of the work being submitted by **LALITHA HARINI CH** bearing the roll number **20021A0503**, in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in **COMPUTER SCIENCE AND ENGINEERING** to the **UCEK(A), JNTUK**, Kakinada, Andhra Pradesh, India. It has been found satisfactory and hereby found satisfactory and hereby approved for submission.

Signature of Head of the Department

Dr. O. Srinivasa Rao
Professor & HOD
Department of CSE
UCEK (A)
JNTU KAKINADA




Certificate of Internship

This is to certify that

Chandu Lalitha Harini

is hereby awarded this certificate for successfully completing 2 months Internship program in **Data Science and Machine Learning** between 02/05/2023 & 30/06/2023. During the time of Internship she has worked with commitment and successfully completed the project. We wish her all the very best for future endeavors.


Program Head



Certificate issued on 30/06/2023

AICTE INTERNSHIP_1680690356642d4cb4eab80

Certificate id : HDLC/IT/MY/2247

ABSTRACT

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. Customer segmentation has the potential to allow marketers to address each customer in the most effective way. Using the large amount of data available on customers and potential customers, a customer segmentation analysis allows marketers to identify discrete groups of customers with a high degree of accuracy based on demographic, behavioral and other indicators. To scale efficiently and effectively, expansion stage companies need to focus their efforts not on a broad universe of potential customers, but rather on a specific subset of customers who are most similar to their best current customers.

The key to doing so is through customer segmentation. The segmentation is based on customers having similar 'needs'(so that a single whole product can satisfy them) and 'buying characteristics (responses to messaging, marketing channels, and sales channels, that a single go- to-market approach can be used to sell to them competitively and economically). In this project we will explore a data set on customers to try to see if there are any discernible segments and patterns. Customer segmentation is useful in understanding what demographic and psychographic sub-populations there are within your customers in a business case. By understanding this, we can better understand how to market and serve them. This project uses the packages such as numpy, pandas, seaborn and matplotlib and the tools such as jupyter notebook to analyze and segment each individual customer into their respective segment based on three important attributes which are "gender", "income" and "expenditure"

CONTENTS		Page No.
Chapter 1	INTRODUCTION	2
Chapter 2	LITERATURE SURVEY	4
Chapter 3	SYSTEM ANALYSIS	6-23
3.1	Aim	
3.2	Existing System	
3.3	Proposed System	
3.4	Hardware Requirements	
3.5	Software Requirements	
Chapter 4	SYSTEM DESIGN	25-26
Chapter 5	IMPLEMENTATION	28-36
5.1	Module Description	
5.2	Dataset Taken	
5.3	Code	
Chapter 6	CONCLUSION	38
Chapter 7	FUTURE ENHANCMENT	40
Chapter 8	BIBLIOGRAPHY	42

CHAPTER I

INTRODUCTION

INTRODUCTION

In the contemporary day and age, the importance of treating customers as the principal asset of an organization is increasing in value. Organizations are rapidly investing in developing strategies for better customer acquisition, maintenance and development. The concept of business intelligence has a crucial role to play in making it possible for organizations to use technical expertise for acquiring better customer insight for outreach programs. In this scenario, the concept of CRM garners much attention since it is a comprehensive process of acquiring and retaining customers, using business intelligence, to maximize the customer value for a business enterprise.

One of the two most important objectives of CRM is customer development through customer insight. This objective of CRM entails the usage of an analytical approach in order to correctly assess customer information and analysis of the value of customers for better customer insight. Keeping up with the changing times, organizations are modifying their business flow models by employing systems engineering as well as change management and designing information technology (IT) solutions that aid them in acquiring new customers, help retain the present customer base and boost the customers lifelong value.

Due to the diverse range of products and services available in the market as well as the intense competition among organizations, customer relationship management has come to play a significant role in the identification and analysis of a company's best customers and the adoption of best marketing strategies to achieve and sustain competitive advantage. One of the most useful techniques in business analytics for the analysis of consumer behavior and categorization is customer segmentation. By using clustering techniques, customers with similar means, end and behavior are grouped together into homogeneous clusters.

CHAPTER II

LITERATURE SURVEY

LITERATURE SURVEY

Research dealing with shopping malls' and/or hypermarkets' attributes, especially in the Indian context, is very less in number. Not many studies have empirically analyzed the influence of an assortment of attributes on buying behavior in shopping arcades and malls and customers shopping experiences. Mostly the researches undertaken so far have been taken from the foreign experiences, as they have come of age in the US, UK and European markets.

Brunner and Mason (1968) investigated the importance of driving time upon the preferences of consumers towards regional shopping centers. They expressed that although it is recognized that population, purchasing power, population density, newspaper circulation, and other factors are influential in determining the shopping habits of consumers, a factor which is generally overlooked is the driving time required to reach the center. In this study, it was established that the driving time required to reach a center is highly influential in determining consumer shopping center preferences. The most consistent and significant driving time dimension in delineating shopping center trade areas was found at the 15-minute driving points, as three fourths of each center's shoppers resided within this range.

Huff (1964 and 1966) concluded that the comparative size of the centers and the convenience of access were the primary characteristics that consumers sought when choosing a shopping center to visit.

Cox and Cooke (1970) determined customer preference for shopping centers and the importance of driving time. The authors concluded that location and attractiveness are important determinants of consumer shopping center preferences

Mehrabian and Russell (1974) noted that the response that store atmosphere elicits from consumers, varies along three dimensions of pleasantness, arousal and dominance

CHAPTER III

SYSTEM ANALYSIS

SYSTEM ANALYSIS

3.1 AIM

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.

3.2 EXISTING SYSTEM

The existing system contains the following drawbacks:

- All the segmentations are search based
- Difficult to gather the data and segment them accordingly
- The results are not really accurate as the clustering is not close enough to determine accurate centroids

3.3 PROPOSED SYSTEM

Our proposed system has the following features:

- Develop the system to get easy visualization techniques
- Increase the data set to accommodate many data points so that results will be more accurate
- Segment the products directly according to the customer group
- Use different methods to collect the customer data instead of physical forms

3.4 HARDWARE REQUIREMENTS:

- Hard disk
- System (8GB RAM and 1TB Hand DISK)
- Forms (To collect data from the customer io malls)

3.5 SOFTWARE REQUIREMENTS:

- Anaconda
- Jupyter
- Kaggle
- Operating system (Windows 10)

ALGORITHM USED

The most common ways in which businesses segment their customer base are: Demographic segmentation: Clustering demographic information such as gender, age, familial and marital status, income, education, and occupation.

Demographic clustering is distribution-based. It provides fast and natural clustering of very large databases. Clusters are characterized by the value distributions of their members. It automatically determines the number of clusters to be generated.

Typically, demographic data contains many categorical variables. The mining function works well with data sets that consist of this type of variables.

You can also use numerical variables. The Demographic Clustering algorithm treats numerical variables by assigning similarities according to the numeric difference of the values

Clustering is an iterative process over the input data. Each input record is read in succession. The similarity of each record with each of the currently existing clusters is calculated. If the biggest calculated similarity is above a given threshold, the record is added to the relevant cluster. This cluster's characteristics change accordingly. If the calculated similarity is not above the threshold, or if there is no cluster (which is initially

the case) a new cluster is created that contains the record alone. You can specify the maximum number of clusters, as well as the similarity threshold.

Demographic Clustering uses the statistical Condorcet criterion to manage the assignment of records to clusters and the creation of new clusters. The Condorcet criterion evaluates how homogeneous each discovered cluster is (in that the records it contains are similar) and how heterogeneous the discovered clusters are among each other. The iterative process of discovering clusters stops after two or more passes over the input data if the improvement of the clustering result according to the Condorcet criterion does not justify a new pass

Geographical segmentation: It differs depending on the scope of the company. For localized businesses, this info might pertain to specific towns or counties. For larger companies, it might mean a customer's city, state, or even country of residence

Geographic segmentation is the simplest type of market segmentation. It categorizes customers based on geographic borders

Geographic Market Segmentation Examples

- ZIP code
- City
- Country
- Radius around a certain location
- Climate
- Urban or rural

Geographic segmentation can refer to a defined geographic boundary (such as a city or ZIP code) or type of area (such as the size of city or type of climate).

An example of geographic segmentation may be the luxury car company choosing to target customers who live in warm climates where vehicles don't need to be equipped for snowy.

The marketing platform might focus their marketing efforts around urban, city centers weather. Where their target customer is likely to work.

We can get details for geographic segmentation and find out where your audience lives using Alexa's Site Overview tool. Enter your site URL, and the report shows you where your website visitors are located across the world.

Psychographic segmentation: Psychographic information such as social class, lifestyle, and personality traits

Psychographic segmentation categorizes audiences and customers by factors that relate to their personalities and characteristics.

Psychographic Market Segmentation Examples

- Personality traits
- Values
- Attitudes
- Interests
- Lifestyles
- Psychological influences
- Subconscious and conscious beliefs
- Motivations
- Priorities

Psychographic segmentation factors are slightly more difficult to identify than demographics because they are subjective. They are not data-focused and require research to uncover and understand.

For example, the luxury car brand may choose to focus on customers who value quality and status. While the B2B enterprise marketing platform may target marketing managers who are motivated to increase productivity and show value to their executive team.

When your obvious groupings of target segments seem to have radically different needs and responses to your offerings and messaging, this is a major indicator it is a good time to look at psychographic segmentation. This method is a powerful way to market the same product to individuals who otherwise seem very heterogeneous. Many expert marketers say this approach will ultimately yield the greatest payoff, in many ways: purchase amount and frequency, lifetime value, loyalty, and more.

Behavioral segmentation: It collects behavioural data, such as spending and consumption habits, product/service usage, and desired benefits.

While demographic and psychographic segmentation focus on who a customer is behavioural focuses on how the customer acts.

Behavioral Market Segmentation Examples

- Purchasing habits
- Spending habits
- User status
- Brand interactions

Behavioral segmentation requires you to know about your customer's actions. These activities may relate to how a customer interacts with your brand or to other activities that happen away from your brand.

A B2C example in this segment may be the luxury car brand choosing to target customers who have purchased a high-end vehicle in the past three years. The B2B marketing platform may focus on leads who have signed up for one of their free webinars

Behavioral segmentation isn't about just recognizing that people have different habits, it's about optimizing marketing campaigns to match these behavioral patterns with a particular message.

Behavioral segmentation is the process of sorting and grouping customers based on the behaviors they exhibit. These behaviors include the types of products and content they consume, and the cadence of their interactions with an app, website, or business.

Acquisition, engagement, and retention are all important factors to keep in mind when analyzing customer behavior. Understanding the following ways your users can interact with your product will help you accomplish a sustainable and constructive behavioral segmentation strategy.

CLUSTERING

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

Clustering analysis can be done on the basis of features where we try to find subgroups of samples based on features or on the basis of samples where we try to find subgroups of features based on samples. We'll cover here clustering based on features. Clustering is used in market segmentation; where we try to find customers that are similar to each other whether in terms of behaviors or attributes, image segmentation/compression; where we try to group similar regions together, document clustering based on topics, etc.

Unlike supervised learning, clustering is considered an unsupervised learning method since we don't have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance. We only want to try to investigate the structure of the data by grouping the data points into distinct subgroups.

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.

Why Clustering?

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

Clustering Methods:

- Density-Based Methods: These methods consider the clusters as the dense region having some similarity and different from the lower dense region of the space. These methods have good accuracy and ability to merge two clusters. Example DBSCAN
- Hierarchical Based Methods: The clusters formed in this method forms a tree-type structure based on the hierarchy. New clusters are formed using the previously formed one.

It is divided into two category

- a. Agglomerative
- b. Divisive

- 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
- 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 operation done on these grids are fast and independent of the number of data objects

It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs.

For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection).

Clustering algorithms can be applied in many fields, for instance:

- Marketing: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records
- Biology: classification of plants and animals given their features;
- Libraries: book ordering;
- Insurance: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- City-planning: identifying groups of houses according to their house type, value and geographical location;
- Earthquake studies: clustering observed earthquake epicenters to identify dangerous zones;
- WWW: document classification; clustering weblog data to discover groups of similar access patterns.

Requirements

The main requirements that a clustering algorithm should satisfy are:

- scalability;
- dealing with different types of attributes;
- discovering clusters with arbitrary shape;
- minimal requirements for domain knowledge to determine input parameters;
- ability to deal with noise and outliers;
- insensitivity to order of input records.

Problems

There are a number of problems with clustering. Among them:

- current clustering techniques do not address all the requirements adequately (and concurrently);
- dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
- the effectiveness of the method depends on the definition of "distance" (for distance-based clustering);
- if an obvious distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;
- the result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

Clustering algorithms may be classified as listed below:

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering
- Probabilistic Clustering

In the first case data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. A simple example of that is shown in the figure below, where the separation of points is achieved by a straight line.

On the contrary the second type, the overlapping clustering, uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.

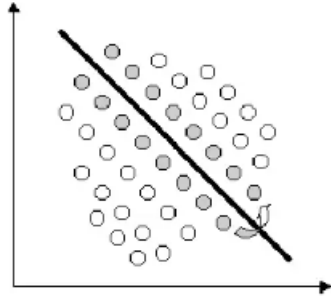


Fig 3.1 Working of a clustering algorithm

Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted.

Finally, the last kind of clustering use a completely probabilistic approach.

In this tutorial we propose four of the most used clustering algorithms:

- K-means
- Fuzzy C-means
- Hierarchical clustering
- Mixture of Gaussians

Each of these algorithms belongs to one of the clustering types listed above. So that K-Means is an exclusive clustering algorithm, Fuzzy C-Means is an overlapping clustering algorithm, Hierarchical clustering is obvious and lastly Mixture of Gaussian is a probabilistic clustering algorithm. We will discuss about each clustering method in the following paragraphs.

Distance Measure

An important component of a clustering algorithm is the distance measure between data points. If the components of the data instance vectors are all in the same physical unit then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading.

Figure shown below illustrates this with an example of the width and height measurements of an object. Despite both measurements being taken in the same

physical units, an informed decision has to be made as to the relative scaling. As the figure shows, different scalings can lead to different clusterings.

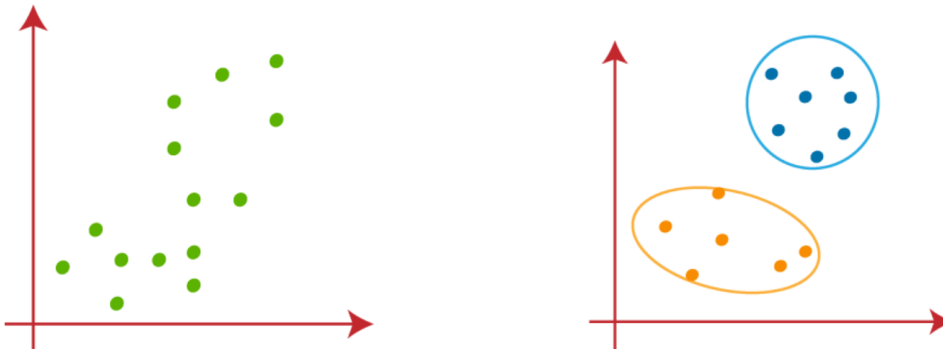


Fig 3.2 Formation of clusters

Notice however that this is not only a graphic issue: the problem arises from the mathematical formula used to combine the distances between the single components of the data feature vectors into a unique distance measure that can be used for clustering purposes different formulas leads to different clusterings.

Measuring Algorithm Performance

One of the most important considerations regarding the ML model is assessing its performance, or you can say the model's quality. In the case of supervised learning algorithms, evaluating the quality of our model is easy because we already have labels for every example.

On the other hand, in the case of unsupervised learning algorithms, we are not that much blessed because we deal with unlabeled data. But still, we have some metrics that give the practitioner insight into the happening of change in clusters depending on the algorithm. What are the criteria for comparing clustering algorithms

Now a good clustering algorithm aims to create clusters whose:

- The intra-cluster similarity is high (The data that is present inside the cluster is similar to one another)
- The inter-cluster similarity is less (Each cluster holds information that isn't similar to the other)

Before we deep dive into such metrics, we must understand that these metrics only Evaluates the comparative performance of models against each other rather than measuring the validity of the model's prediction.

You still don't know which cluster is which class, and if they make any sense at all. In this case, you can validate your results by simple sampling from the clusters and looking at the quality of classification. If the questions are split reasonably, you can register a label for every cluster and either label the whole dataset, train a supervised model, or you can continue to use the k-means cluster, keeping the information about which cluster corresponds to which class.

Applications of Clustering

Customer Segmentation: Subdivision of customers into groups/segments such that each customer segment consists of customers with similar market characteristics - pricing, loyalty, spending behaviour etc. Some of the segmentation variables could be, e.g., the number of items bought on sale, avg transaction value, the total number of transactions.

Creating News Feeds: K-Means can be used to cluster articles by their similarity — it can separate documents into disjoint clusters.

Cloud Computing Environment: Clustered storage to increase performance, capacity, or reliability

— clustering distributes workloads to each server, manages the transfer of workloads between servers, and provides access to all files from any server regardless of the physical location of the data.

Environmental risks: K-means can be used to analyse environmental risk in an area — environmental risk zoning of a chemical industrial area.

Pattern Recognition in images: For example, to automatically detect infected fruits or for segmentation of blood cells for leukaemia detection.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their understanding of what constitutes a cluster and how to efficiently find them.

K-Means Clustering algorithm

K-means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group.

It tries to make the intra-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

The way k-means algorithm works is as follows:

1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
 - Compute the sum of the squared distance between data points and all centroids.
 - Assign each data point to the closest cluster (centroid).
 - Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach means follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a breakdown of how we can solve it mathematically (feel free to skip it).

The objective function is:

$$J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|x^i - \mu_k\|^2 \quad (1)$$

where $w_{ik}=1$ for data point x_i if it belongs to cluster k ; otherwise, $w_{ik}=0$. Also, μ_k is the centroid of x_i 's cluster.

It's a minimization problem of two parts. We first minimize J w.r.t. w_{ik} and treat μ_k fixed.

Then we minimize J w.r.t. μ_k and treat w_{ik} fixed. Technically speaking, we differentiate J w.r.t. w_{ik} first and update cluster assignments (E-step). Then we differentiate J w.r.t. μ_k and recompute the centroids after the cluster assignments from previous step (M-step). Therefore, E-step is:

$$\begin{aligned}\frac{\partial J}{\partial w_{ik}} &= \sum_{i=1}^m \sum_{k=1}^K \|x^i - \mu_k\|^2 \\ \Rightarrow w_{ik} &= \begin{cases} 1 & \text{if } k = \operatorname{argmin}_j \|x^i - \mu_j\|^2 \\ 0 & \text{otherwise.} \end{cases}\end{aligned}$$

In other words, assign the data point x_i to the closest cluster judged by its sum of squared distance from cluster's centroid.
And M-step is:

$$\begin{aligned}\frac{\partial J}{\partial \mu_k} &= 2 \sum_{i=1}^m w_{ik} (x^i - \mu_k) = 0 \\ \Rightarrow \mu_k &= \frac{\sum_{i=1}^m w_{ik} x^i}{\sum_{i=1}^m w_{ik}}\end{aligned}$$

Which translates to recomputing the centroid of each cluster to reflect the new assignments

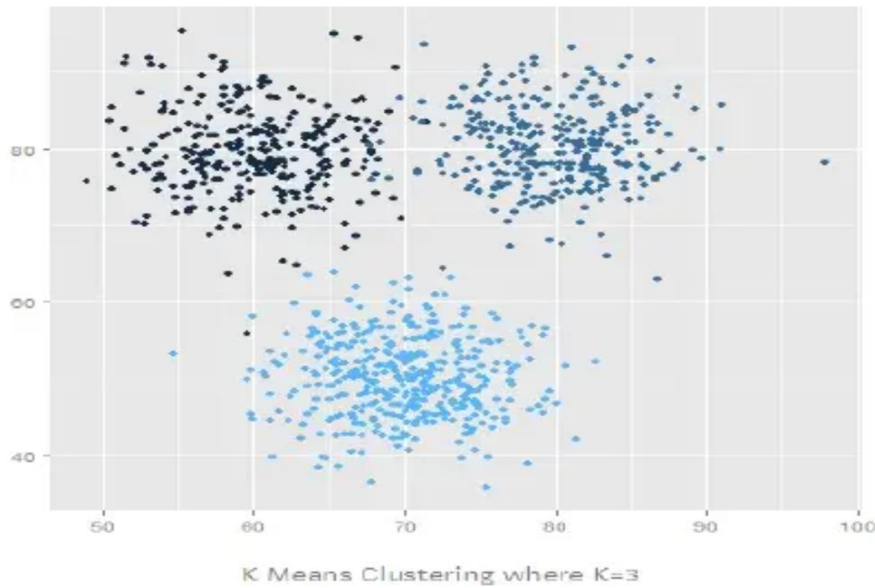


Fig 3.3 K Means Clustering where K=3

K-means is an unsupervised clustering algorithm designed to partition un-labelled data into a certain number (thats the "K") of distinct groupings. In other words, k-means finds observations that share important characteristics and classifies them together into clusters. A good clustering solution is one that finds clusters such that the observations within each cluster are more similar than the clusters themselves.

There are countless examples of where this automated grouping of data can be extremely useful. For example, consider the case of creating an online advertising campaign for a brand-new range of products being released to the market. While we could display a single generic advertisement to the entire population, a far better approach would be to divide the population into clusters of people who hold shared characteristics and interests displaying customized advertisements to each group.

K-means is an algorithm that finds these groupings in big datasets where it is not feasible to be done by hand. The intuition behind the algorithm is actually pretty straight forward. To begin, we choose a value for k (the number of clusters) and randomly choose an initial centroid (center coordinates) for each cluster.

We then apply a twostep process:

1. Assignment step - Assign each observation to its nearest center.
2. Update step — Update the centroids as being the center of their respective observation.

We repeat these two steps over and over until there is no further change in the clusters. At this point the algorithm has converged and we may retrieve our final clusterings

One final key aspect of k-means returns to this concept of convergence. We previously mentioned that the k-means algorithm doesn't necessarily converge to the global minima and instead may converge to a local-minima.

In fact, depending on which values we choose for our initial centroids we may obtain differing results.

As we are only interested in the best clustering solution for a given choice of k , a common solution to this problem is to run k-means multiple times, each time with different randomised initial centroids, and use only the best solution. In other words, always run k-means multiple times to ensure we find a solution close to the global minima.

Advantages

- 1) Fast, robust and easier to understand.
- 2) Relatively efficient: $O(tknd)$, where n is # objects, k is # clusters, d is # dimension of each object, and t is # iterations. Normally, $k, t, d \ll n$.
- 3) Gives best result when data set are distinct or well separated from each other.

Disadvantages

- 1) The learning algorithm requires apriori specification of the number of cluster centers.
- 2) The use of Exclusive Assignment - If there are two highly overlapping data then k-means will not be able to resolve that there are two clusters.
- 3) The learning algorithm is not invariant to non-linear transformations i.e. with different representation of data we get.
- 4) Euclidean distance measures can unequally weight underlying factors.
- 5) The learning algorithm provides the local optima of the squared error function.
- 6) Randomly choosing of the cluster center cannot lead us to the fruitful result. Pl. refer Fig.
- 7) Applicable only when mean is defined i.e. fails for categorical data.
- 8) Unable to handle noisy data and outliers
- 9) Algorithm fails for non-linear data set. The main drawback of this technique is related to ambiguity about the K number of points that should be initialized. To overcome this issue, the performance of the algorithm is calculated for different numbers of centroids.

Conclusion

K-means is one of the most common and intuitive clustering algorithms in Machine Learning. The name 'k-means' almost explains the theory itself.

- 'K' number of data points is initialized.
- The mean of the corresponding features of the nearest data points is calculated And set as a new coordinate of the pre-initialize.

EVALUATION METHODS:

Contrary to supervised learning where we have the ground truth to evaluate the model's performance, clustering analysis doesn't have a solid evaluation metric that we can use to evaluate the outcome of different clustering algorithms. Moreover, since k-means requires k as an input and doesn't learn it from data, there is no right answer in terms of the number of clusters that we should have in any problem. Sometimes domain knowledge and intuition may help but usually that is not the case. In the cluster-predict methodology, we can evaluate how well the models are performing based on different K clusters since clusters are used in the downstream modeling.

We'll cover metric that may give us some intuition about k:

- Elbow method
- Quick method

ELBOW METHOD:

Elbow method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. We pick k at the spot where SSE starts to flatten out and forming an elbow. We'll use the geyser dataset and evaluate SSE for different values of k and see where the curve might form an elbow and flatten out.

Then, plot a line chart of the SSE for each value of k. If the line chart looks like an arm, then the "elbow" on the arm is the value of k that is the best. The idea is that we want a small SSE, but that the SSE tends to decrease toward 0 as we increase k (the SSE is 0 when k is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the center of its cluster). So our goal is to choose a small value of k that still has a low SSE, and the elbow usually represents where we start to have diminishing returns by increasing k.

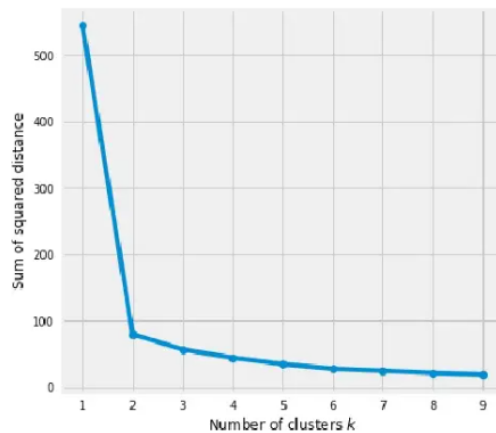


Fig 3.4 Example of the Elbow method

The graph above shows that $k=2$ is not a bad choice. Sometimes it's still hard to figure out a good number of clusters to use because the curve is monotonically decreasing and may not show any elbow or has an obvious point where the curve starts flattening out.

Quick Method

The same functionality above can be achieved with the associated quick method `k-elbow` visualizer. This method will build the `K-elbow Visualizer Object` with the associated arguments, fit it, then (optionally) immediately show the visualization.

The `K-Elbow Visualizer` implements the "elbow" method of selecting the optimal number of clusters for K-means clustering. K-means is a simple unsupervised machine learning algorithm that groups data into a specified number (k) of clusters. Because the user must specify in advance what k to choose, the algorithm is somewhat naive - it assigns all members to k clusters even if that is not the right k for the dataset.

The elbow method runs k-means clustering on the dataset for a range of values for k and then for each value of k computes an average score for all clusters. By default, the distortion score is computed, the sum of square distances from each point to its assigned center. Other metrics can also be used such as the silhouette score, the mean silhouette coefficient for all samples or the score, which computes the ratio of dispersion between and within clusters.

CHAPTER IV

SYSTEM DESIGN

SYSTEM DESIGN

SYSTEM ARCHITECTURE

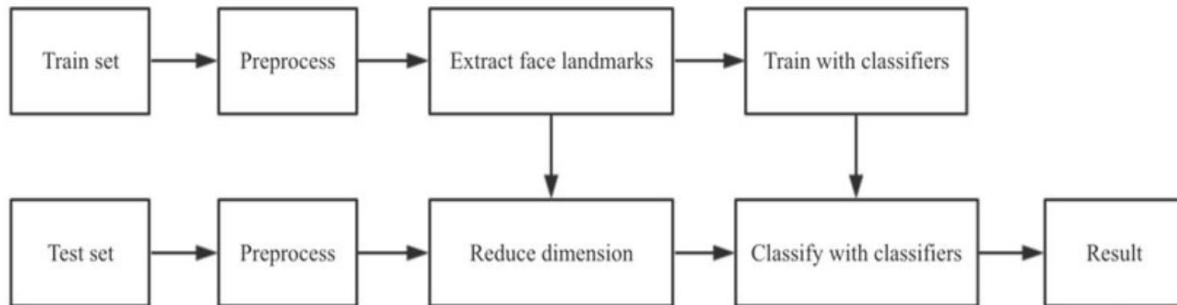


Fig 5.1 Sytem architecture of a machine learning algorithm and how it flows

The machine learning architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out in the transformation of raw data into training data sets capable for enabling the decision making of a system.

DATAFLOW DIAGRAM

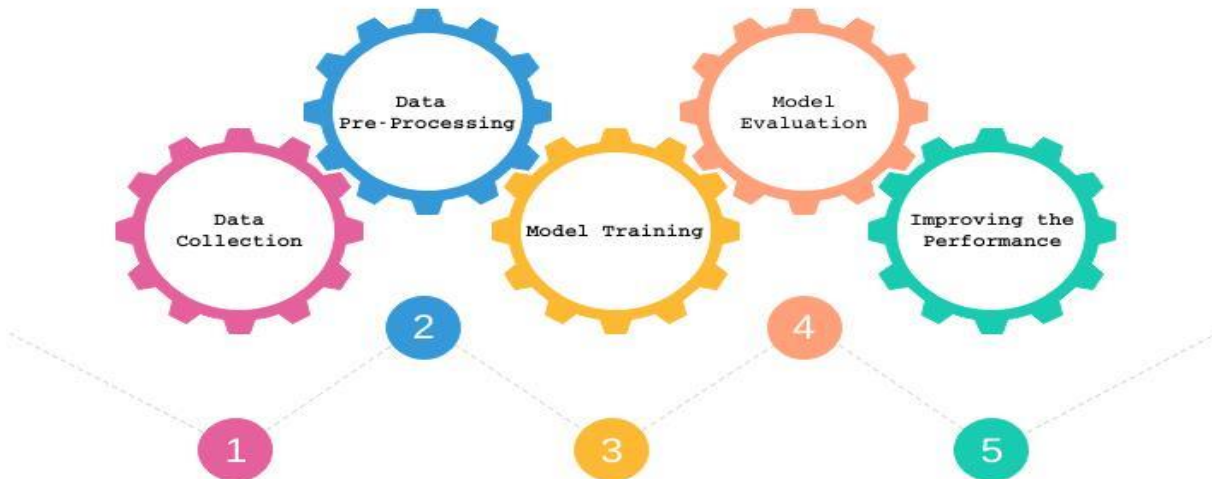
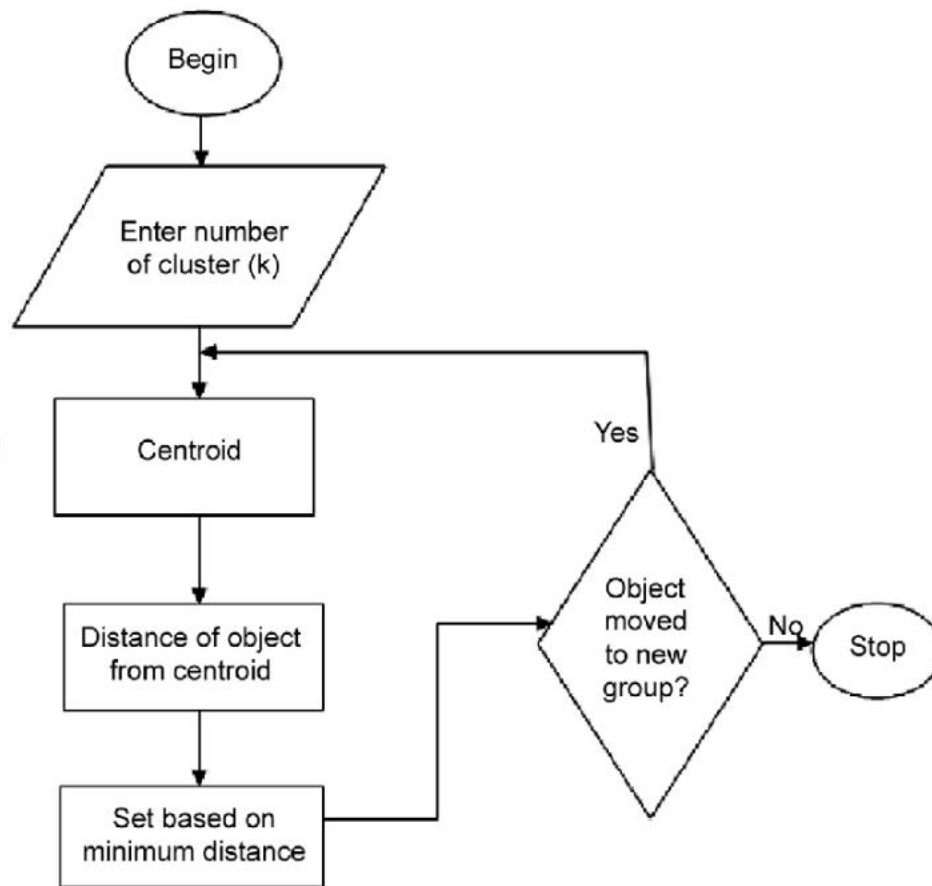


Fig 5.2 The dataflow diagram of a how a machine learning algorithm works

CLUSTERING ALGORITHM



The algorithm splits a given dataset into different clusters on the bases of data density. It works on each data point and check the proximity of the given data point from all the cluster centers. k-means then allocates that data point to the cluster whose cluster center (centroid) is closest to it.

CHAPTER V

IMPLEMENTATION

IMPLEMENTATION

5.1 MODULE DESCRIPTION

NO. OF MODULES

- Administrator
- Customer

MODULE DESCRIPTION

- Administrator: Administrator is the controller of the survey link. Admin will perform all the controlling operations of the model. Admin designs an algorithm to get expected expenditure score of customer after customer fills his details in the form.
- Customer: The customer is for whom the output is targeted. They give their details such as their age, income, gender etc by filling in the survey form. User cannot make changes to the model, but can only use the already trained model.

INPUT AND OUTPUT

The following are some of the inputs and outputs:

INPUTS:

- Admin trains the model
- Admin trains the datasets
- Admin tests the model
- Admin adds various categories
- Customer enters his details
- Details which user enters are his age, income, gender.

OUTPUTS:

- Admin will get the corresponding expenditure score of a customer
- Admin will be able to cluster different categories of customers
- Customers may buy their desired products quite effectively and quickly
- Different products can be sold to different categories of customers with a guaranteed customer satisfaction

CODING

5.2 DATASET TAKEN

The dataset consists of Annual income of 1 lakh customers and their total expenditure score (in \$) for a period of one year. This dataset is taken from Kaggle which contains various types of datasets. Let us explore the data using numpy and pandas libraries in python

This dataset contains the basic information (ID, age, gender, income, spending score) about the customers

	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

Steps in the analysis

1. Importing Libraries.
2. Data Exploration and Visualization.
3. Clustering using K-Means.
4. Selecting variable combinations
5. Selecting Clusters.
6. Plotting the Cluster Boundary and Clusters.
7. 3D Plot of Clusters.

CODING

```
In [3]: ▶ import pandas as pd
```

```
In [4]: ▶ df=pd.read_csv("cust.csv")
```

```
In [5]: ▶ df.head()
```

Out[5]:

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

```
In [6]: ▶ df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   CustomerID                            200 non-null   int64
1   Gender                                200 non-null   object
2   Age                                    200 non-null   int64
3   Annual Income (k$)                    200 non-null   int64
4   Spending Score (1-100)                 200 non-null   int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

```
In [7]: ▶ df.shape
```

Out[7]: (200, 5)

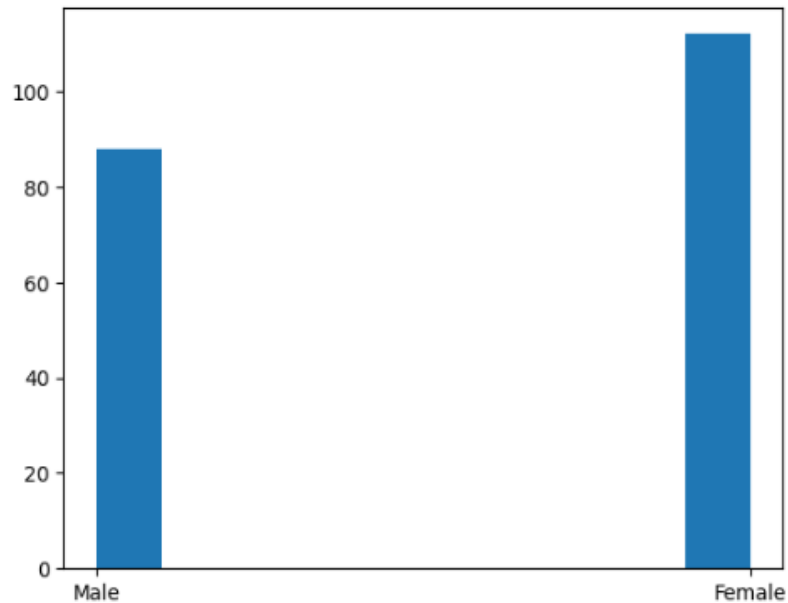
preprocessing the data

```
In [8]: ▶ df.isnull().sum()
```

```
Out[8]: CustomerID      0
Gender      0
Age         0
Annual Income (k$)    0
Spending Score (1-100) 0
dtype: int64
```

```
In [9]: import matplotlib.pyplot as plt
```

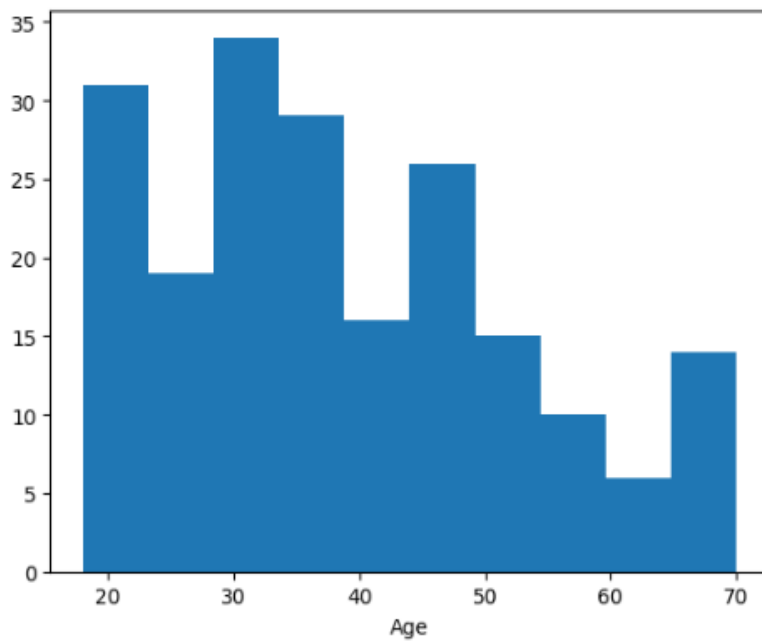
```
In [10]: plt.hist(df['Gender'])  
plt.show()
```



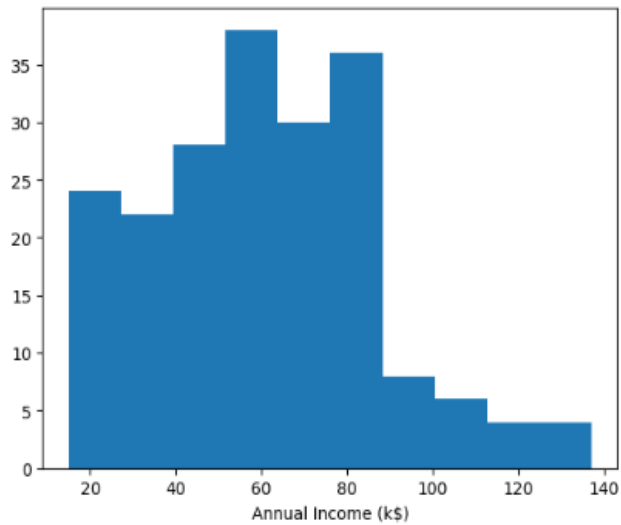
1st observation: female are more likely to shop than male

analyze behaviours based on Age

```
In [11]: plt.hist(df['Age'])  
plt.xlabel("Age")  
plt.show()
```

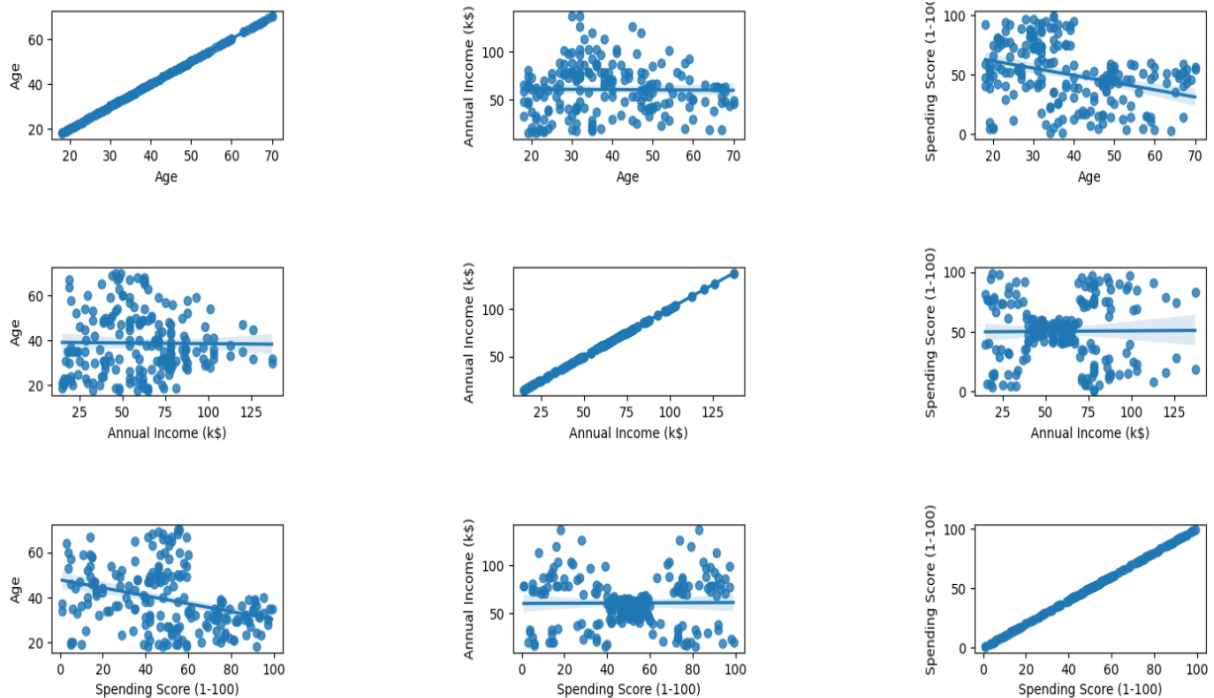


```
In [12]: plt.hist(df['Annual Income (k$)'])
plt.xlabel("Annual Income (k$)")
plt.show()
```



```
In [13]: import seaborn as sns
```

```
In [14]: plt.figure(1, figsize = (17 , 8))
n = 0
for x in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
    for y in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
        n += 1
        plt.subplot(3,3,n)
        plt.subplots_adjust(hspace=1, wspace=1)
        sns.regplot(x=x, y=y, data=df)
        plt.ylabel(y)
plt.show()
```



Observations:

1. with increase in age there is no change in annual income (subplot-[1,2]).
2. with increase in age there is decrease in spending score (subplot-[1,3]).
3. observing subplot-[2,3] conclusions can be,
 - annual income < 37.5(k\$) and > ~70(k\$) has either category of more spending and less spending customers.
 - people of annual income in between [37.5(k\$), ~70(k\$)] exhibit very similar behaviour in spending money.

```
In [15]: from sklearn.cluster import KMeans
```

K-means clustering using 2 dimensions

age,spending score be the dimensions taken under consideration to form clusters.

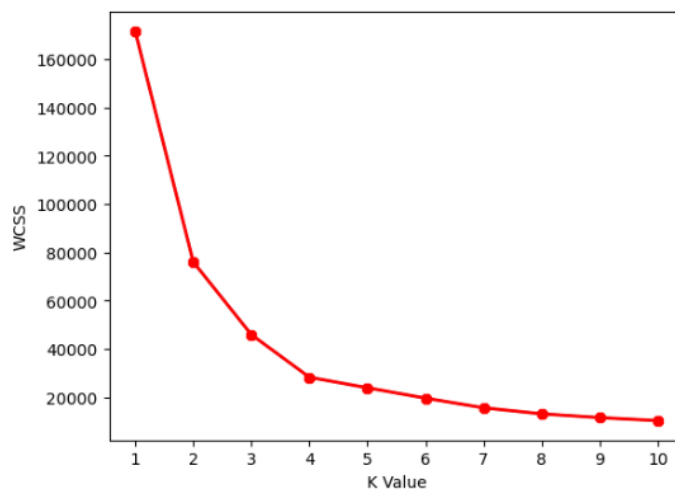
K value is not defined. Lets calculate WSS for all values of k<=10. WSS stands for Within Cluster Sum of Squared Errors.

```
In [16]: X = df[['Age','Spending Score (1-100)']].iloc[:, :].values
```

```
In [17]: wcss=[]
for i in range(1,11):
    km=KMeans(n_clusters=i)
    km.fit(X)
    wcss.append(km.inertia_)
```

```
In [18]: import numpy as np
```

```
In [19]: 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()
```



Observations:

k<4 gives underfitting
k>4 gives overfitting
Lets choose k=4 to avoid above extremes.

```
In [20]: km1=KMeans(n_clusters=4)
#Fitting the input data
km1.fit(X)
y=km1.predict(X)
#adding the labels to a column named label
df["label"] = y
#The new dataframe with the clustering done
df.head()
```

```
C:\Users\Lalitha Harini\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Lalitha Harini\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[20]:

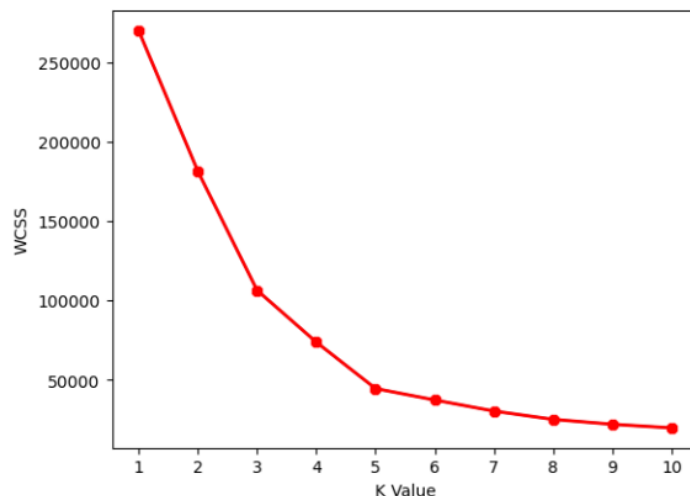
	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)	label
0	1	Male	19	15	39	3
1	2	Male	21	15	81	1
2	3	Female	20	16	6	2
3	4	Female	23	16	77	1
4	5	Female	31	17	40	3

In []:

Lets do the same with another pair: Annual income,spending score be the dimensions taken under consideration to from clusters

```
In [21]: X = df[['Annual Income (k$)','Spending Score (1-100)']].iloc[:, :].values
wcss=[]
for i in range(1,11):
    km=KMeans(n_clusters=i)
    km.fit(X)
    wcss.append(km.inertia_)
```

```
In [23]: 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()
```



Observations:

k<5 gives underfitting. k>5 gives overfitting. Lets choose k=5 to avoid above extremes.

```
In [24]: km1=KMeans(n_clusters=5)
#Fitting the input data
km1.fit(X)
y=km1.predict(X)
#adding the labels to a column named label
df["label1"] = y
#The new dataframe with the clustering done
df.head()

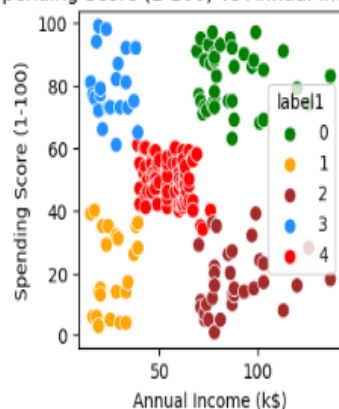
C:\Users\Lalitha Harini\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
  warnings.warn(
C:\Users\Lalitha Harini\anaconda3\lib\site-packages\sklearn\cluster\_kmeans.py:1382: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[24]:

	CustomerID	Gender	Age	Annual Income (k\$)	Spending Score (1-100)	label	label1
0	1	Male	19	15	39	3	1
1	2	Male	21	15	81	1	3
2	3	Female	20	16	6	0	1
3	4	Female	23	16	77	1	3
4	5	Female	31	17	40	3	1

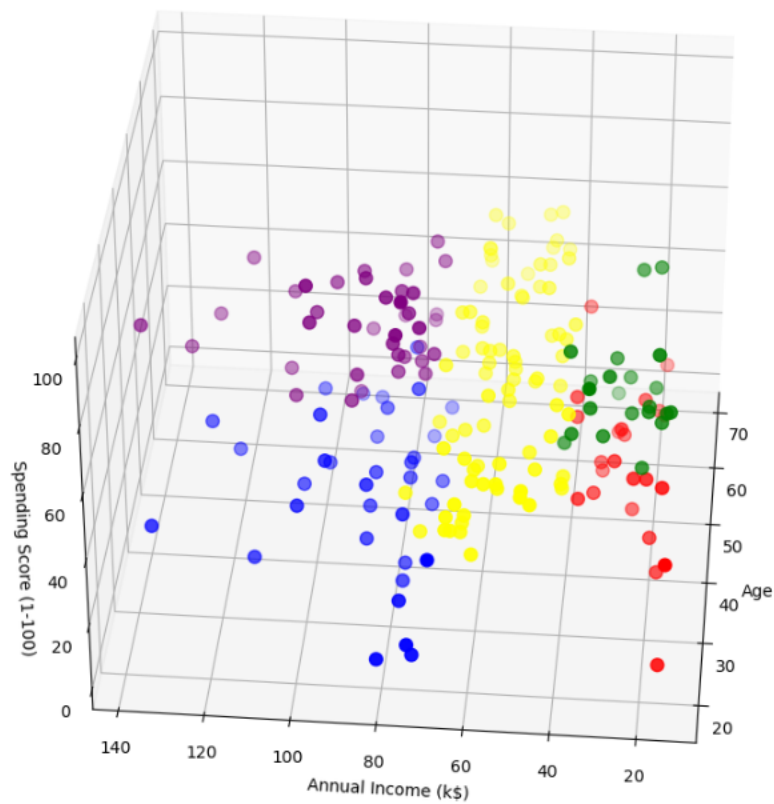
```
In [28]: #Scatterplot of the clusters
plt.figure(figsize=(3,3))
sns.scatterplot(x = 'Annual Income (k$)',y = 'Spending Score (1-100)',hue="label1",
               palette=['green','orange','brown','dodgerblue','red'], legend='full',data = df ,s = 60 )
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Spending Score (1-100) vs Annual Income (k$)')
plt.show()
```

Spending Score (1-100) vs Annual Income (k\$)



Lets form 3D cluster plot

```
[32]: fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df.Age[df.label1 == 0], df["Annual Income (k$)"][df.label1 == 0], df["Spending Score (1-100)"][df.label1 == 0],
ax.scatter(df.Age[df.label1 == 1], df["Annual Income (k$)"][df.label1 == 1], df["Spending Score (1-100)"][df.label1 == 1],
ax.scatter(df.Age[df.label1 == 2], df["Annual Income (k$)"][df.label1 == 2], df["Spending Score (1-100)"][df.label1 == 2],
ax.scatter(df.Age[df.label1 == 3], df["Annual Income (k$)"][df.label1 == 3], df["Spending Score (1-100)"][df.label1 == 3],
ax.scatter(df.Age[df.label1 == 4], df["Annual Income (k$)"][df.label1 == 4], df["Spending Score (1-100)"][df.label1 == 4],
ax.view_init(35, 185)
plt.xlabel("Age")
plt.ylabel("Annual Income (k$)")
ax.set_zlabel('Spending Score (1-100)')
plt.show()
```



CHAPTER VI

CONCLUSION

CONCLUSION

Our project classifies various customers into different clusters so that different marketing strategies can be employed to different clusters attain maximum profit. Due to increasing commercialization, consumer data is increasing exponentially. When dealing with this large magnitude of data, organizations need to make use of more efficient clustering algorithms for customer segmentation. These clustering models need to possess the capability to process this enormous data effectively. Each of the above discussed clustering algorithms come with their own set of merits and demerits. The computational speed of K-Means clustering algorithm is relatively better as compared to the hierarchical clustering algorithms as the latter require the calculation of the full proximity matrix after each iteration. K-Means clustering gives better performance for a large number of observations while hierarchical clustering has the ability to handle fewer data points.

The major hindrance produces itself in the form of selecting the numbers of clusters „K" for the K-Means process, which have to be provided as an input to this non-hierarchical clustering algorithm. This limitation does not exist in the case of hierarchical clustering since it does not require any cluster centers as input. It depends on the user to choose the cluster groups as well as their number. Hierarchical clustering also gives better results as compared to K-Means when a random dataset is used. The output or results obtained when using hierarchical clustering are in the form of dendrograms but the output of K-Means consists of flat structured clusters which may be difficult to analyze. As the value of k increases, the quality(accuracy) of hierarchical clustering improves when compared to K-Means clustering. As such, partitioning algorithms like K-Means are suitable for large datasets while hierarchical clustering algorithms are more suitable for small datasets.

CHAPTER VII

FUTURE ENHANCEMENT

FUTURE ENHANCEMENT

We have done this project with as minimum flaws as possible and can further be enhanced by including major identification of statistics of people and improving the accuracy of the output.

All the census data also can be collected to train the dataset even more to get more accurate outputs.

Based on the social data accumulated, we can conclude that mall customer segmentation system can be used in a wide range of applications across a variety of domains including:

- Identifying interests of people while they are buying items from a mall
- Grouping people efficiently
- Identifying many more clusters to segment the products to improve the sales of the product

In this project we have implemented k-means algorithm, it can be further enhanced by using few complex algorithms such as conventional neural networks algorithms.

All the census data also can be collected to train the dataset even more to get more accurate outputs.

There is no need for privacy invasion of users like other applications like amazon etc.

CHAPTER VII

BIBLIOGRAPHY

BIBLIOGRAPHY

Here are some references for customer segmentation that you may find useful:

- Kumar, V., & Reinartz, W. (2016). Customer Relationship Management: Concept, Strategy and Tools. Springer.
- Wedel, M., & Kamakura, W. A. (2012). Market Segmentation: Conceptual and Methodological Foundations (2nd ed.). Springer.
- Wind, Y., & Green, P. E. (1973). On the use of segmentation in marketing: Efficiency, Effectiveness and strategic concern. *Journal of Marketing Research*, 10(4), 394-405.
- Kotler, P., & Armstrong, G. (2016). Principles of Marketing (16th ed.). Pearson.
- Dibb, S., Simkin, L., Pride, W. M., & Ferrell, O. C. (2012). Marketing: Concepts and Strategies. Cengage Learning.

Make sure to check the specific chapters or sections related to customer segmentation in these books for more in-depth information.