**PROJECT TITLE - CUSTOMER SEGMENTATION**

**TEAM MEMBER**

**210621104011 – DHARUNRAJ N**

**APPLIED DATA SCIENCE**

**Phase-4 Development Part 2**

# 1.Introduction

The main goal of this project is to use advanced techniques in data science to group customers based on how they behave, what they like, and information about them like their age or location. This grouping strategy is really important because it helps businesses create special marketing plans for each group of customers. This can make customers happier and more likely to keep coming back to the business. The project has several important steps like getting the data, cleaning it up, making new useful information from the data, using different ways to group the customers, showing the data in pictures, and understanding what the groups mean.

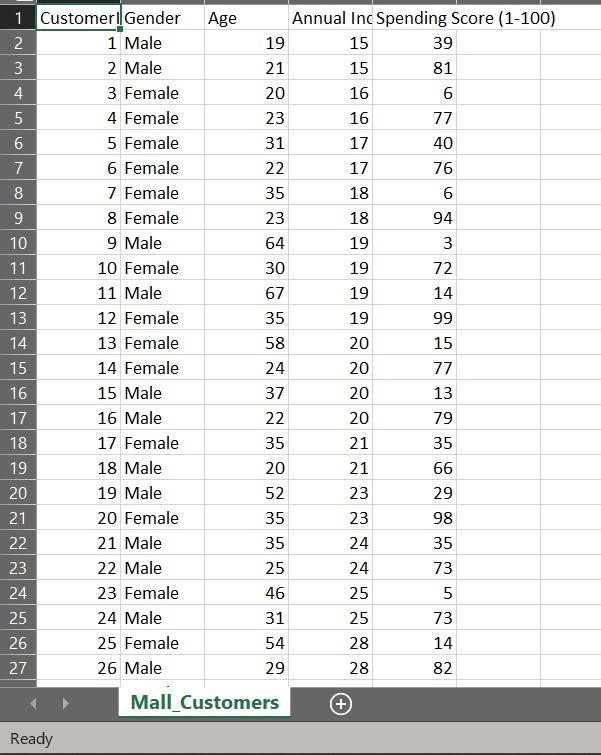
## 2.Dataset

The data set is collected from Kaggle which is a leading collaborative data science platform.

Dataset Link :<https://www.kaggle.com/datasets/akram24/mall-customers>

All the data are stored in the Mall\_Customers.csv (Comma Separated Values) format, which is used to store the data efficiently.

Using pandas, we can use the .csv format for Data Processing and Manipulation



## The columns that are present are

1. CustomerID
2. Gender
3. Age
4. Annual Income
5. Spending Score

CustomerID only contains values that are useless, so we will not be using it.

We will use all the other columns except CustomerID for Exploratory Data Analysis (EDA)

For clustering, we will be using Annual Income and Spending Score column.

**3. Feature Engineering**

Feature engineering refers to manipulation — addition, deletion, combination, mutation — of your data set to improve machine learning model training, leading to better performance and greater accuracy

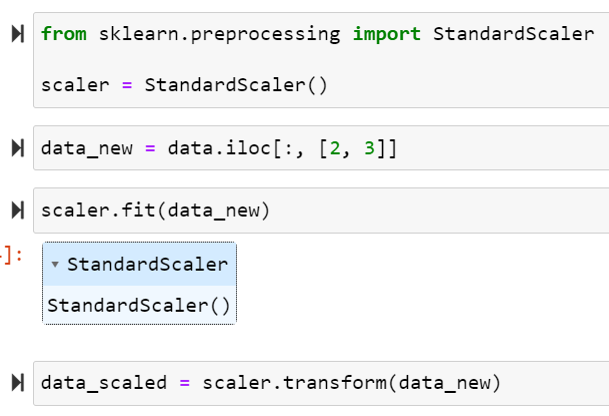
**Removing Useless Column**



We will remove the “CustomerID” column from our data. Because the values of this column are useless.

**Feature Scaling**

Feature scaling is essential for machine learning algorithms that calculate distances between data. If not scaled, the feature with a higher value range starts dominating when calculating distances. KNN which uses Euclidean distance is one such algorithm which essentially require scaling



**4.Model Training**

K-means clustering is used when you have unlabelled data (i.e., data without defined categories or groups). It's capable of classifying unlabelled data into a predetermined number of clusters based on similarities (k)

**Code**

**# Importing Libraries**

import pandas as pd

from sklearn.cluster import KMeans

import warnings

warnings.filterwarnings("ignore")

import matplotlib.pyplot as plt

import seaborn as sns

**# Loading the data**

data = pd.read\_csv("Mall\_Customers.csv")

data.head()

**# Preprocessing**

data.drop(["CustomerID"], axis=1, inplace=True)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

data\_new = data.iloc[:, [2, 3]]

scaler.fit(data\_new)

data\_scaled = scaler.transform(data\_new)

**# Model Training**

kmeans = KMeans(5)

clusters = kmeans.fit\_predict(data\_new)

data\_new["Clusters"] = clusters

clusters = kmeans.cluster\_centers\_

**5.Model Evaluation**

Model evaluation is a process that uses metrics to understand a machine learning model's performance. Some of the commonly used metrics for clustering are WCSS (Within Clusters Sum of Squares) and Silhouette Score

**WCSS**

It measures the distance between each observation and the centroid and calculates the squared difference between the two. The idea is to minimize the sum.

**Code**

wcss = []

for i in range(1, 10):

kmeans = KMeans(n\_clusters=i)

kmeans.fit(data\_scaled)

wcss.append(kmeans.inertia\_)

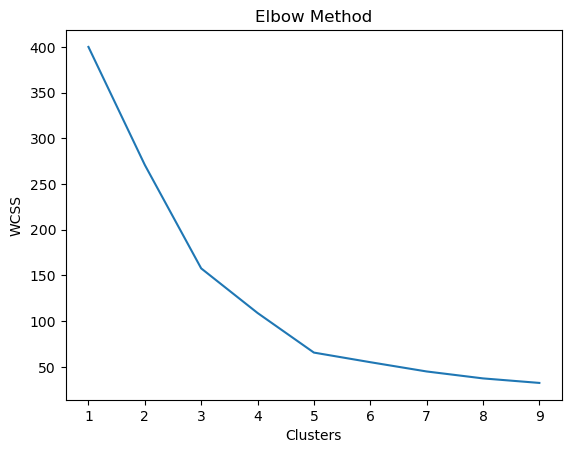
plt.plot(range(1,10),wcss)

plt.xlabel("Clusters")

plt.ylabel("WCSS")

plt.title("Elbow Method")

plt.show()

****

Using the WCSS we can choose no of clusters as either 3 or 5.

We use silhouette score to further find the optimal no of clusters.

**Silhouette Score**

The silhouette score is a metric that measures how well a data point fits into its assigned cluster and how distinct it is from other clusters.

The silhouette score ranges from -1 to 1. A value of 1 means that clusters are well apart from each other and clearly distinguished.

A high value indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters.

Negative values generally indicate that a sample has been assigned to the wrong cluster.

**Code**

from sklearn.metrics import silhouette\_score

silhouette\_avg=[]

for i in range(2,10):

temp\_kmeans=KMeans(n\_clusters=i)

temp\_kmeans.fit(data\_scaled)

cluster\_label=temp\_kmeans.labels\_

silhouette\_avg.append(silhouette\_score(data\_scaled,cluster\_label))

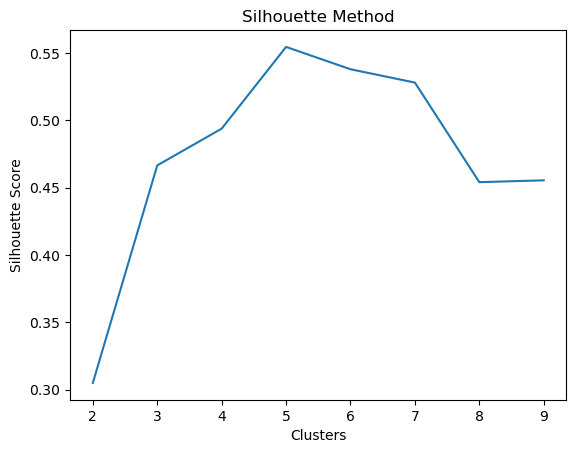
plt.plot(range(2,10),silhouette\_avg)

plt.xlabel("Clusters")

plt.ylabel("Silhouette Score")

plt.title("Silhouette Method")

plt.show()



From this metrics we choose 5 as the optimal number of clusters