**CUSTOMER SEGMENTATION USING DATA SCIENCE**

## **Project Submission Part 5: Project Documentation & Submission**

**Abstract:**

Nowadays Customer segmentation became very popular method for dividing company’s customers for retaining customers and making profit out of them, in the following study customers of different of organizations are classified on the basis of their behavioral characteristics such as spending and income, by taking behavioral aspects into consideration makes these methods efficient compared to others. For this classification a machine algorithm named as K-means clustering algorithm is used and based on the behavioral characteristic’s customers are classified. Formed clusters help the company to target individual customers and advertise the content to them through marketing campaigns and social media sites which they are really interested in.

**Keywords:** Machine learning, Customer segmentation, K-means algorithm.

**Introduction:**

Today many of the businesses are going online and, in this case, online marketing is becoming essential to hold customers, but during this, considering all customers as same and targeting all of them with similar marketing strategy is not very efficient way rather it's also annoying the customers by neglecting his or her individuality, so customer segmentation is becoming very popular and also became the efficient solution for this existing problem. Customer segmentation is defined as dividing a company's customers on the basis of demographic (age, gender, marital status) and behavioral (types of products ordered, annual income) aspects. Since demographic characteristics does not emphasize on individuality of customer because same age groups may have different interests so behavioral aspects are a better approach for customer segmentation as its focus on individuality and we can do proper segmentation with the help of it.

**Phase1:**

## **Problem definition**:

Implement data science techniques to segment customers based on their behavior, preferences, and demographic attributes, enabling businesses to personalize marketing strategies and enhance customer satisfaction.

**Design thinking:**

* **Data Collection:** Collecting customer data, including attributes like purchase history, demographic information, and interaction behavior.
* **Data Preprocessing:** Cleaning and preprocessing the data, handling missing values, and converting categorical features into numerical representations.
* **Feature Engineering:** Creating additional features that capture customer behavior and preferences, such as total spending, frequency of purchases, etc.
* **Clustering Algorithm:** K-Means clustering to segment customers.
* **Visualization:** Visualizing the customer segments by plotting all the clusters and their centroids.
* **Interpretation**: Analyzing and interpreting the characteristics of each customer segment to derive actionable insights for marketing strategies.

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

* Unlike supervised learning algorithms, K-means clustering is an unsupervised machine learning algorithm. This algorithm is used when we have unlabeled data. Unlabeled data means input data without categories or groups provided. Our customer segmentation data is like this for this problem.
* The algorithm discovers groups (cluster) in the data, where the number of clusters is represented by the K value. The algorithm acts iteratively to assign each input data to one of K clusters, as per the features provided. All of this makes k-means quite suitable for the customer segmentation problem.
* Given a set of data points are grouped as per feature similarity. The output of the K-means clustering algorithm is:

1. The centroids values for K clusters,
2. Labels for each input data point.
3. At the end of implementation, we’re going to get output such as a group of clusters along with which customer belongs to which cluster.

**Phase 2:**

**Dataset:**

i)We got dataset from Kaggle.

ii)The dataset link: <https://www.kaggle.com/datasets/akram24/mall-customers>.

iii)It is a mall customer dataset. It contains 5 columns and 200 rows. It have details

about 200

iv)customers, they are customer ID, genre(gender), age, Annual income and their

spending score.

**Columns:**

1.Customer ID: A unique identifier for each customer.

2.Age: Age of the customer.

3.Gender: Customer's gender (male, female, other).

4.Income: Customer's annual income.

5.Spending score: Spending score, also known as customer spending score or

purchase score, is a numerical value that represents a customer's spending behavior

or propensity to make purchases. It is commonly used in customer segmentation as

one of the key variables to group customers based on their purchasing patterns and preferences.

**Libraries to be used:**

1.Numpy.

2.Pandas.

3.Matplotlib.

4.Seaborn.

5.sklearn.cluster import kmeans

**How to train and test data:**

1. **Data Preparation:**
   1. Collect and clean the customer data, ensuring it's in a format suitable for analysis.
   2. Handle missing values, outliers, and data inconsistencies.
   3. Normalize or standardize numeric features if necessary.
   4. Encode categorical variables if needed (e.g., one-hot encoding).
2. **Data Splitting:**
   1. Split your dataset into two subsets: one for training and one for testing.
   2. Common splitting ratios are 70-80% for training and 20-30% for testing. Adjust as needed based on our dataset size.
3. **Feature Selection or Engineering:**
   1. Choose relevant features for customer segmentation. This step may involve domain knowledge or feature selection techniques.
   2. Create new features if they could enhance segmentation (e.g., calculating RFM scores).
4. **Model Selection:**
   1. Decide on the segmentation algorithm(s) you want to use. We are going to use k-means clustering.
   2. Choose appropriate hyper parameters for the chosen algorithm(s).
5. **Training:**
   1. Use the training dataset to fit your segmentation model(s). We're using k-means clustering, we'd fit the k-means algorithm to our training data.
6. **Testing:**
   1. Apply the trained model to the testing dataset to segment the customers in the test set.
   2. Evaluate the performance of our segmentation model. Common evaluation metrics include silhouette score, Davies-Bouldin index, or domain-specific metrics.
7. **Visualization and Interpretation:**
   1. Visualize the segmented customer groups to gain insights.
   2. Interpret the characteristics of each segment and give them meaningful labels (e.g., "High-Value Customers," "Low-Activity Customers").
   3. We use scatter plots.
8. **Application:**

Apply the customer segments to your business strategies, such as targeted marketing, product recommendations, or customer retention efforts.

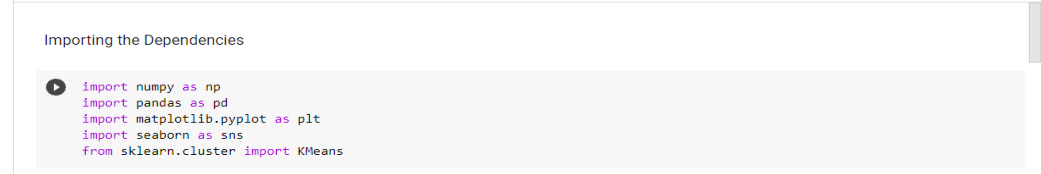
**Accuracy check:**

Accuracy in customer segmentation is typically assessed using metrics like silhouette score, Davies-Bouldin index, or other clustering evaluation metrics. Scatter plots themselves are not used to directly measure accuracy, but they can be a valuable tool for visualizing and interpreting our segmentation results.

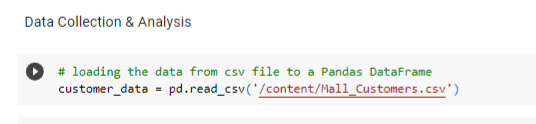
**Phase 3:**

**Importing the required libraries:**

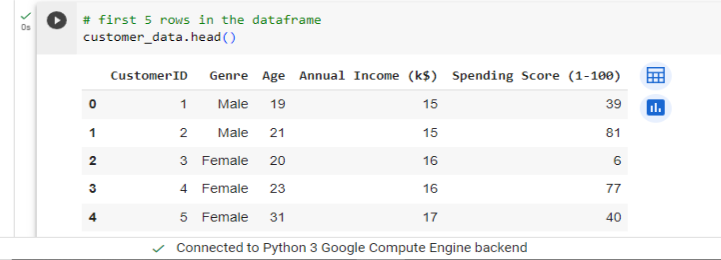
Let’s start the development part of customer segmentation using data science by importing the necessary Python libraries.



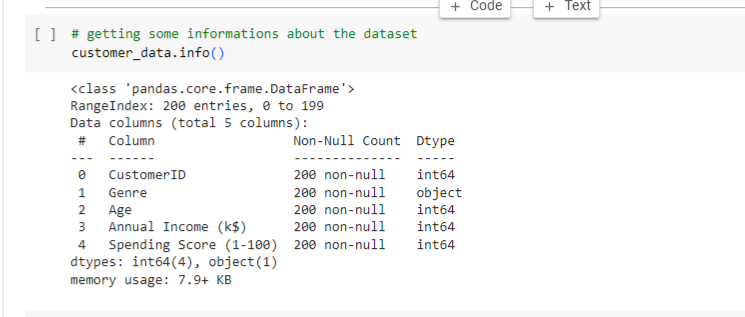
**Importing the dataset:**

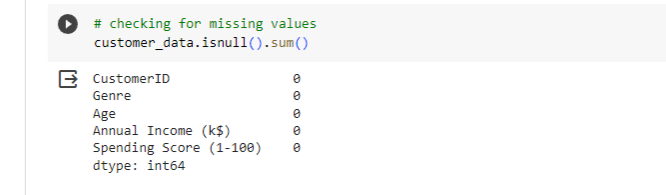


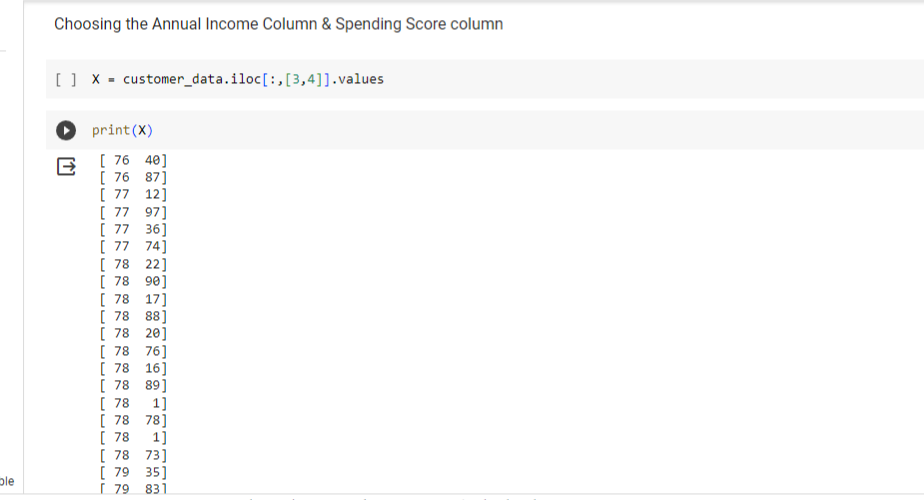
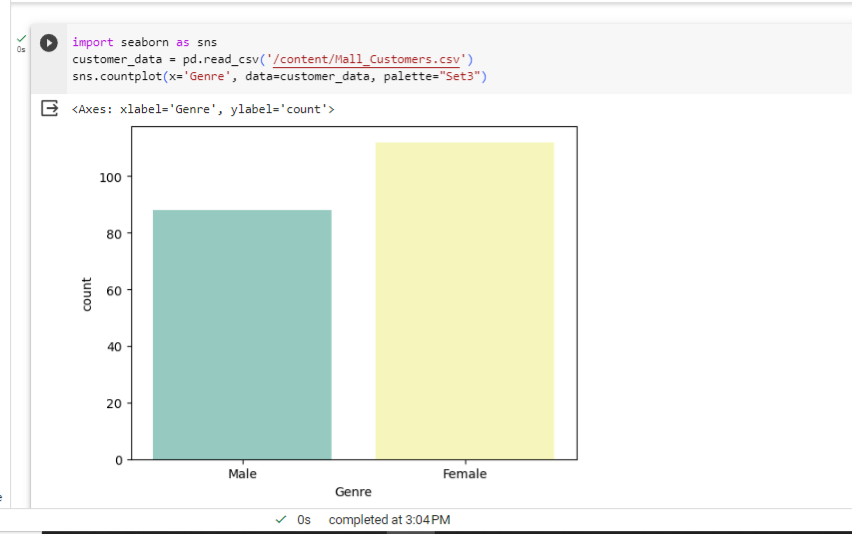
**Printing first five rows of the dataset:**



**Analyzing the dataset:**





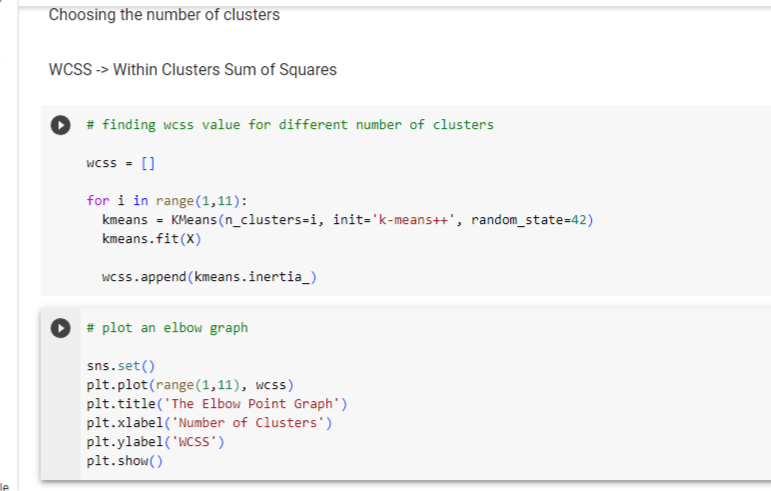
**Encoding the categorical features:**

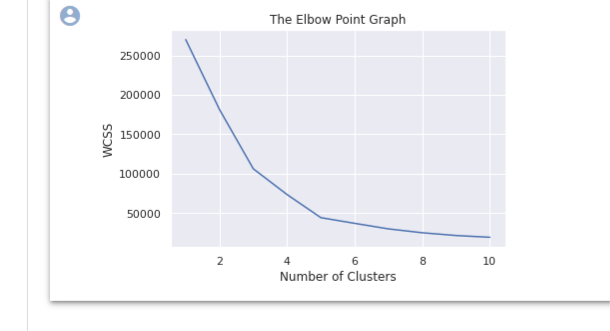
* Encoding categorical data using one- hot encoding (OneHotEncoder) is a process of representing categorical variables in a numerical format that can be used in various machine learning algorithms
* Encoding categorical data using OneHotEncoder is a process in data projects where categorical variables are transformed into a numerical format, specifically binary vectors.
* Each unique category or label within a categorical variable is converted into a separate binary column, and for each observation, the column corresponding to its category is marked with a “1,” while all other columns are set to “O.”
* This method is used to make categorical data compatible with machine learning algorithms that require numerical input.
* OneHotEncoder ensures that the categorical data is represented in a way that doesn’t introduce false ordinal relationships or numerical values, preventing biases in the model’s interpretation of the data



**Choosing the number of clusters:**

K Means Clustering Using the Elbow Method For each value of K, we are calculating WCSS (Within-Cluster Sum of Square). WCSS is the sum of the squared distance between each point and the centroid in a cluster. When we plot the WCSS with the K value, the plot looks like an Elbow

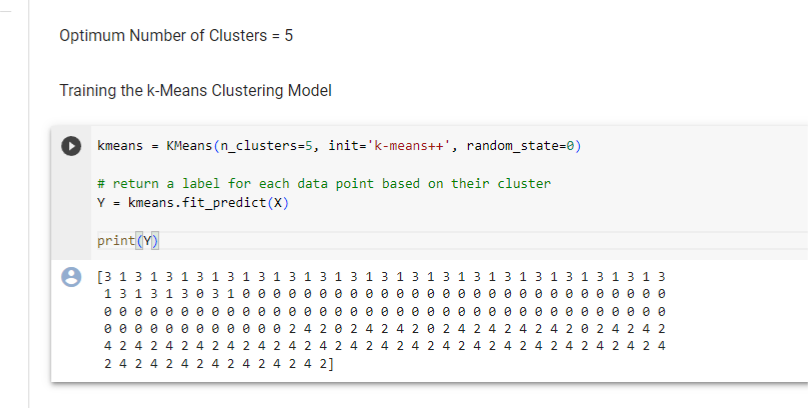


.

**Training the k-Means algorithm:**

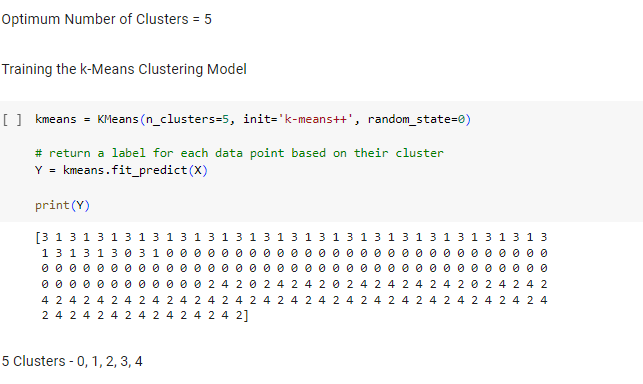
**1.Testing set:** The testing set is a smaller portion of the data, usually around 20-30% of the dataset. It is kept separate and is not used during the model training phase. Instead, it is used to evaluate the model’s performance by making predictions or performing analyses and comparing them to the actual, known outcomes.

**2.Training set**: This subset contains a majority of the data, typically around 70-80% of the dataset. It is used to train machine learning models or perform data analysis tasks. The model learns patterns, relationships, and trends within the data from this set.

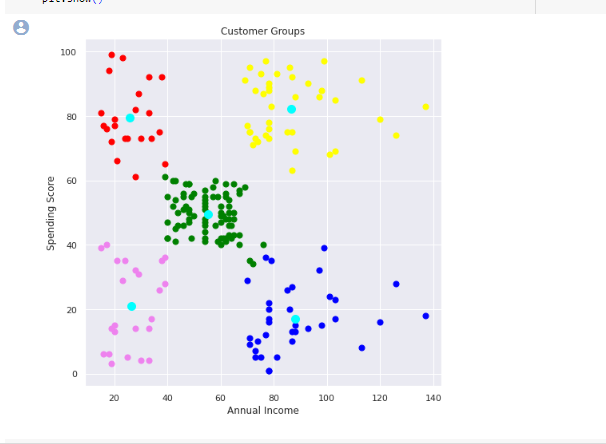


**Phase4:**

**Training the KMeans Algorithm:**



**Visualizing the clusters:**



**Algorithm used:**

K-means is a popular clustering algorithm used in machine learning and data analysis. It's a partitioning algorithm that divides a dataset into K distinct, non-overlapping subgroups or clusters. The goal is to group data points that are similar to each other into the same cluster while keeping dissimilar data points in different clusters. K-means is an unsupervised learning algorithm, meaning it doesn't rely on labeled data for training.

Here's an overview of how the K-means algorithm works:

* **Initialization:** Choose K initial cluster centroids. These centroids are typically randomly selected from the data points, or you can use other initialization methods.
* **Assignment:** For each data point in the dataset, calculate its distance (usually Euclidean distance) to all K centroids and assign the data point to the cluster with the nearest centroid. This step forms K clusters.
* **Update:** Calculate the new centroids for each cluster by taking the mean of all data points in that cluster. These centroids represent the center of the clusters.
* **Iteration:** Repeat the assignment and update steps until a stopping condition is met. Common stopping conditions include a maximum number of iterations, convergence (when the centroids no longer change significantly), or a predetermined threshold.

The algorithm aims to minimize the sum of squared distances between data points and their respective cluster centroids. This is often referred to as the "within-cluster sum of squares" (WCSS) or "inertia."

K-means is efficient and easy to understand, making it a popular choice for clustering tasks. However, it has some limitations:

* The algorithm's performance can be sensitive to the initial choice of cluster centroids, potentially leading to suboptimal results.
* It assumes that clusters are spherical, equally sized, and have similar densities, which may not hold in all datasets.
* The number of clusters (K) must be specified in advance, and choosing an appropriate value for K can be challenging. Various methods, such as the elbow method or silhouette analysis, can help with this.

Despite these limitations, K-means is a valuable tool for many clustering tasks and is widely used in various domains, including customer segmentation, image processing, and document clustering.

**Observations:**

• The age of the customers ranges from 18-70. This shows that the mall attracts

shops and things which suit all age group people.

• The average age of customers is 39.

• The average income of customers is 60 K$.

• The average spending score of customers is 50.

**Drawback of System:**

1. Marketing will become expensive.

2. Because of having less no. of customers in a segment problem of limited production occurs.

**Conclusions:**

Customer segmentation is performed on the company's customers data and with the help of K-means clustering machine learning algorithm customers are divided using features like total spending and annual income, this study also proves that the dividing customers on the basis of behavioral characteristics is a better solution for existing customer segmentation problem and K-means clustering algorithm is identified as a good choice for this approach.

**CODING:**

Importing the Dependencies

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

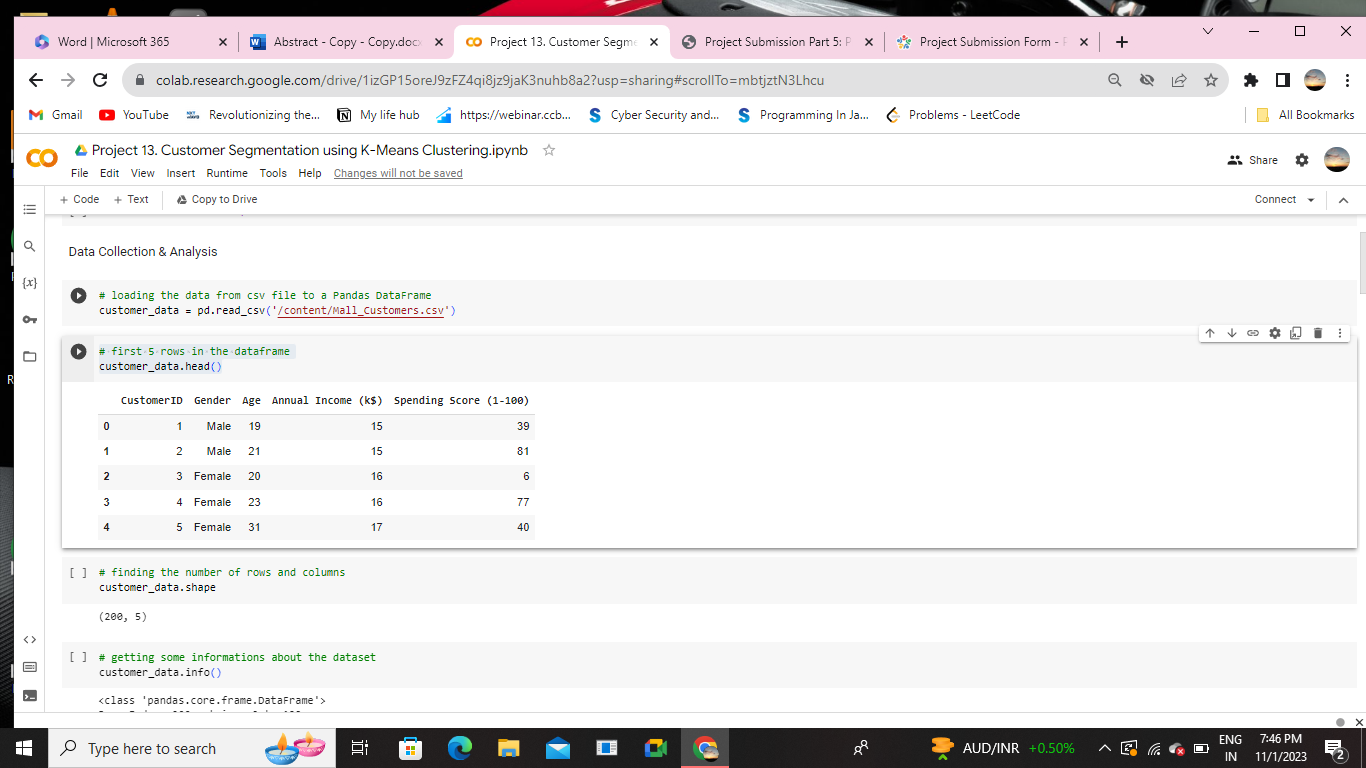
Data Collection & Analysis

# loading the data from csv file to a Pandas DataFrame

customer\_data = pd.read\_csv('/content/Mall\_Customers.csv')

# first 5 rows in the dataframe

customer\_data.head()



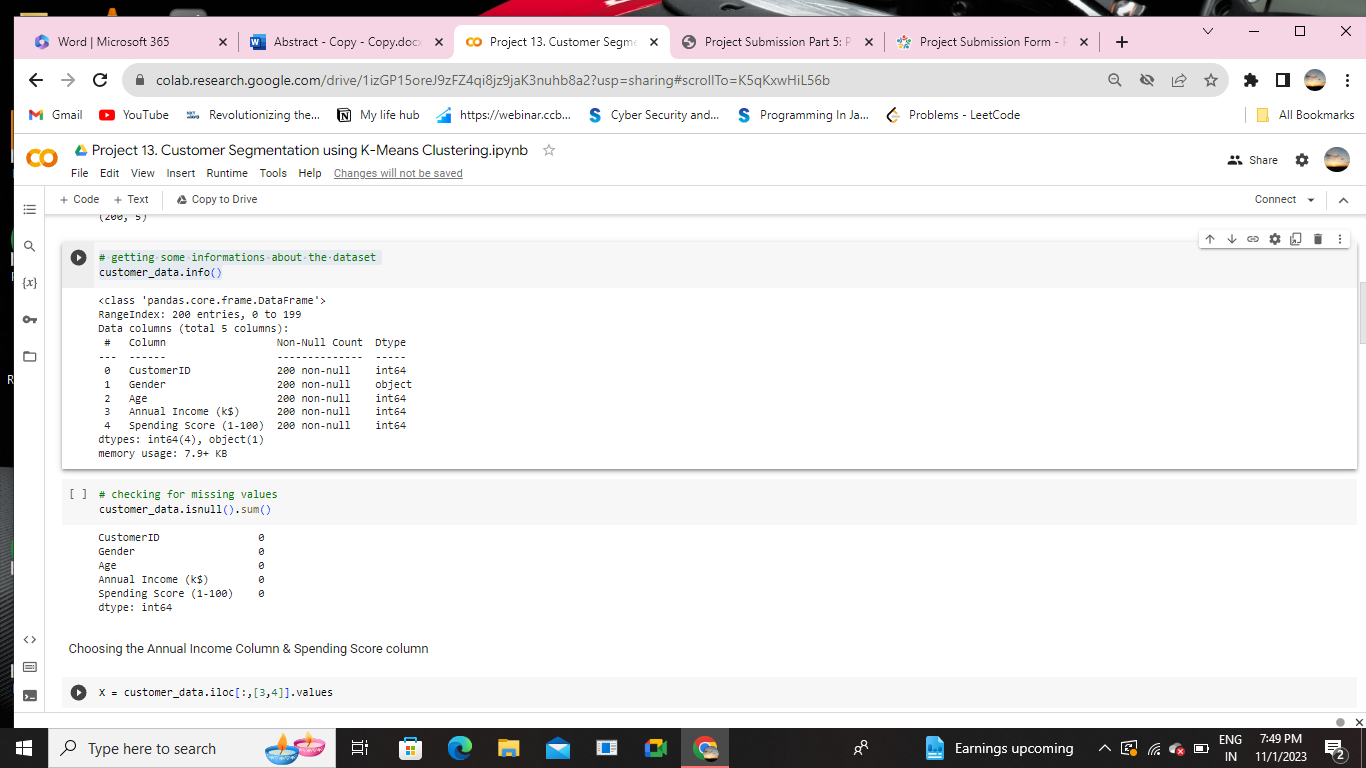
# finding the number of rows and columns

customer\_data.shape

(200, 5)

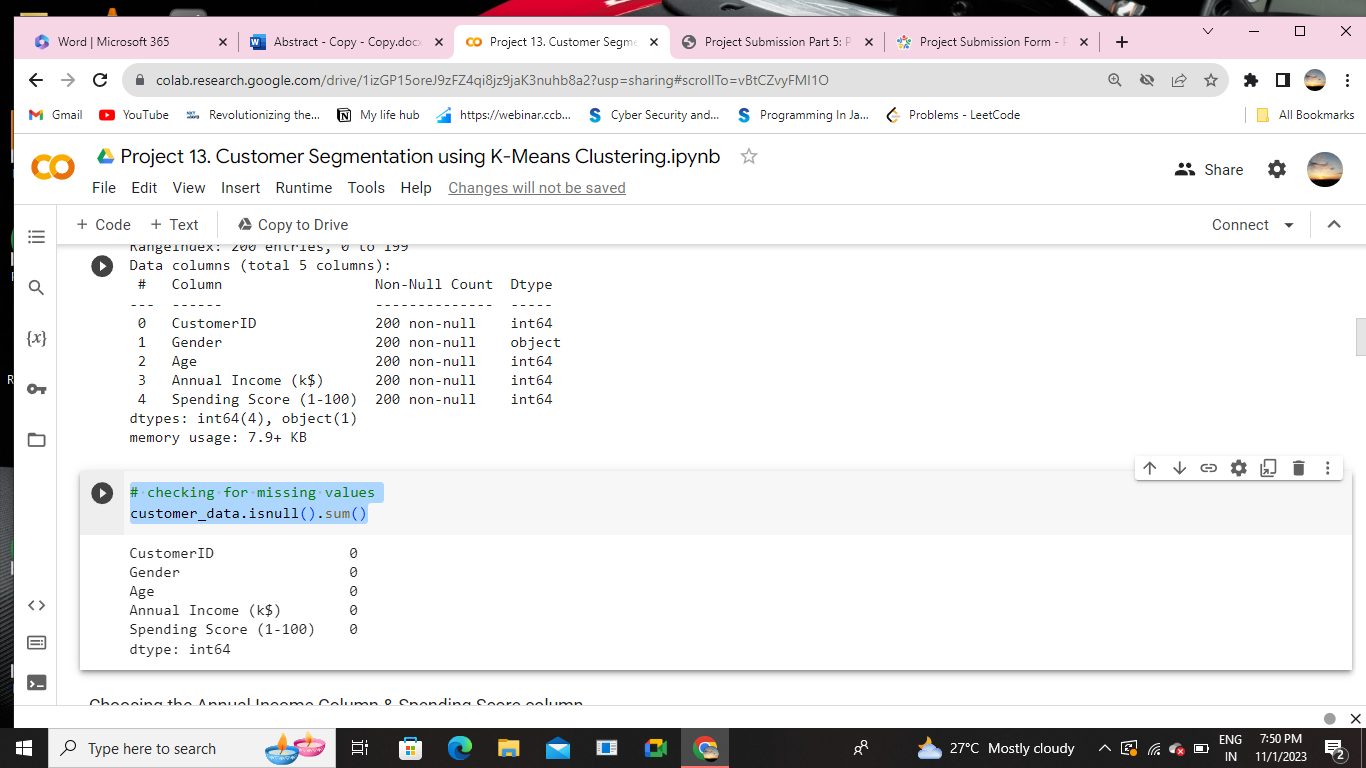
# getting some informations about the dataset

customer\_data.info()



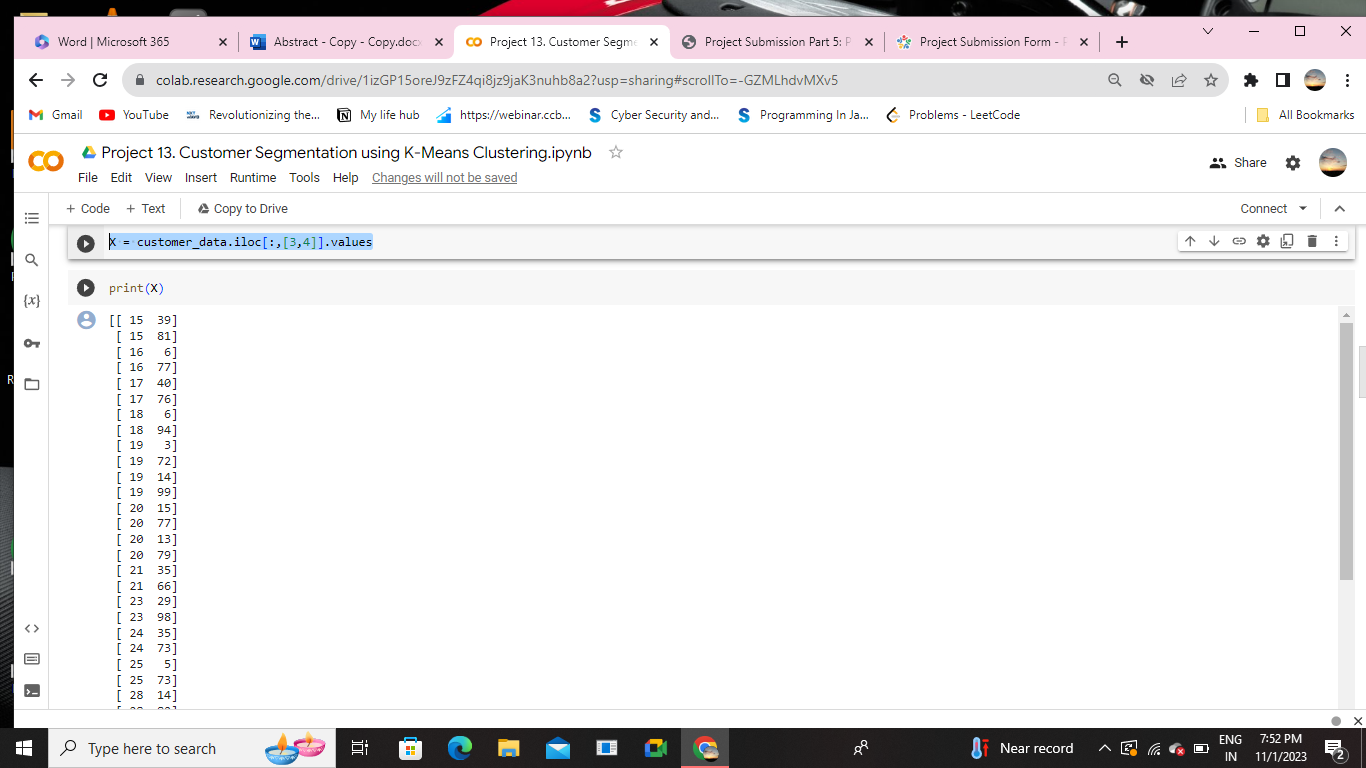
# checking for missing values

customer\_data.isnull().sum()



Choosing the Annual Income Column & Spending Score column

X = customer\_data.iloc[:,[3,4]].values



Choosing the number of clusters

WCSS -> Within Clusters Sum of Squares

# finding wcss value for different number of clusters

wcss = []

for i in range(1,11):

kmeans = KMeans(n\_clusters=i, init='k-means++', random\_state=42)

kmeans.fit(X)

wcss.append(kmeans.inertia\_)

# plot an elbow graph

sns.set()

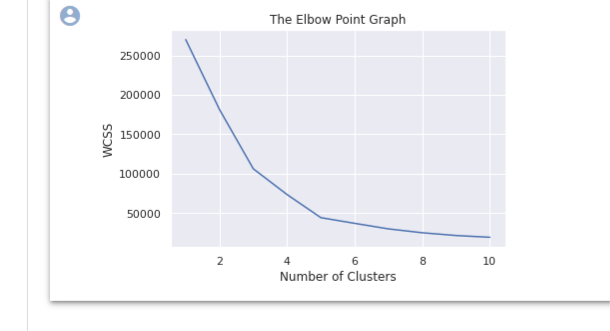
plt.plot(range(1,11), wcss)

plt.title('The Elbow Point Graph')

plt.xlabel('Number of Clusters')

plt.ylabel('WCSS')

plt.show()



Optimum Number of Clusters = 5

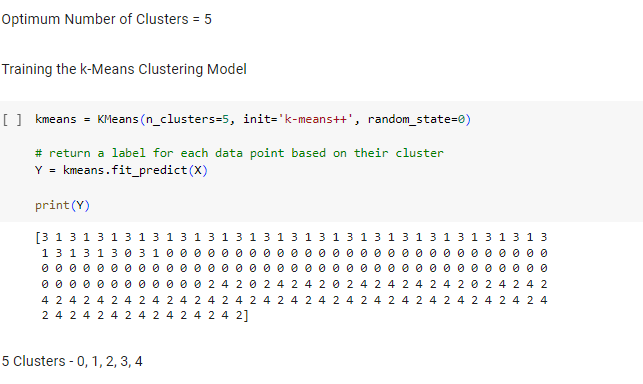
Training the k-Means Clustering Model

kmeans = KMeans(n\_clusters=5, init='k-means++', random\_state=0)

# return a label for each data point based on their clustery

Y=kmeans.fit\_predict(X)

print(Y)



Visualizing all the Clusters

# plotting all the clusters and their Centroids

plt.figure(figsize=(8,8))

plt.scatter(X[Y==0,0], X[Y==0,1], s=50, c='green', label='Cluster 1')

plt.scatter(X[Y==1,0], X[Y==1,1], s=50, c='red', label='Cluster 2')

plt.scatter(X[Y==2,0], X[Y==2,1], s=50, c='yellow', label='Cluster 3')

plt.scatter(X[Y==3,0], X[Y==3,1], s=50, c='violet', label='Cluster 4')

plt.scatter(X[Y==4,0], X[Y==4,1], s=50, c='blue', label='Cluster 5')

# plot the centroids

plt.scatter(kmeans.cluster\_centers\_[:,0], kmeans.cluster\_centers\_[:,1], s=100, c='cyan', label='Centroids')

plt.title('Customer Groups')

plt.xlabel('Annual Income')

plt.ylabel('Spending Score')

plt.show()

**OUTPUT:**

