Machine Learning Assignment

**CUSTOMER SEGMENTATION**

Name – Shruti Badjate

Roll No- BT21CSE027

Guided by – R. B. Keskar sir

**Problem Statement**

Customer segmentation involves categorizing customers based on various characteristics such as demographics, behaviour, and psychology. This enables organizations to understand their customer base better and make informed decisions regarding product development and marketing strategies. Implementing customer segmentation leads to business optimization in budgeting, product design, promotion, marketing, and customer satisfaction.

**Input Attributes for Dataset**

1. **Customer ID**: This remains unchanged as it uniquely identifies each customer.
2. **Gender**: This attribute represents the gender of the customer.
3. **Age**: Age of the customer.
4. **Annual Income (k$)**: This can be derived from the "Money Spent" feature, assuming it represents the annual income of the customer in thousands of dollars.
5. **Spending Score (1-100)**: This can be a combination of the "Products Purchased" and "Complaints" features. A higher number of products purchased may indicate higher spending propensity, while a higher number of complaints may indicate dissatisfaction and thus a lower spending score. This score can be normalized to a scale of 1 to 100.

**Output for Dataset**

Group customers into distinct clusters based on similarities in their annual income and spending behaviour.

**Dataset**

<https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python>

I also provide this data set in mail with name as mall-customers.csv .

**3-Different Algorithms**

1. **K-Means Clustering**: It's a popular unsupervised learning algorithm used for clustering similar data points into groups. K-Means can efficiently handle large datasets and is suitable for identifying distinct customer segments based on similarities in features like annual income and spending behaviour.
2. **Hierarchical Clustering**: This algorithm creates a hierarchy of clusters by either bottom-up (agglomerative) or top-down (divisive) approaches. It's useful for understanding relationships between different customer segments and can provide insights into the hierarchical structure of customer preferences.
3. **Gaussian Mixture Models (GMM)**: GMM assumes that the data points are generated from a mixture of several Gaussian distributions. It's more flexible than K-Means as it accommodates different cluster shapes and can capture complex relationships in the data. GMM is suitable for projects where the underlying data distribution is not well-defined.

For customer segmentation projects, I recommend using the **K-Means Clustering** algorithm. K-Means is straightforward, computationally efficient, and provides clear cluster assignments. It's suitable for segmenting customers based on features like annual income and spending behaviour, allowing businesses to identify distinct customer groups for targeted marketing strategies and personalized offerings.

**Libraries Used**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

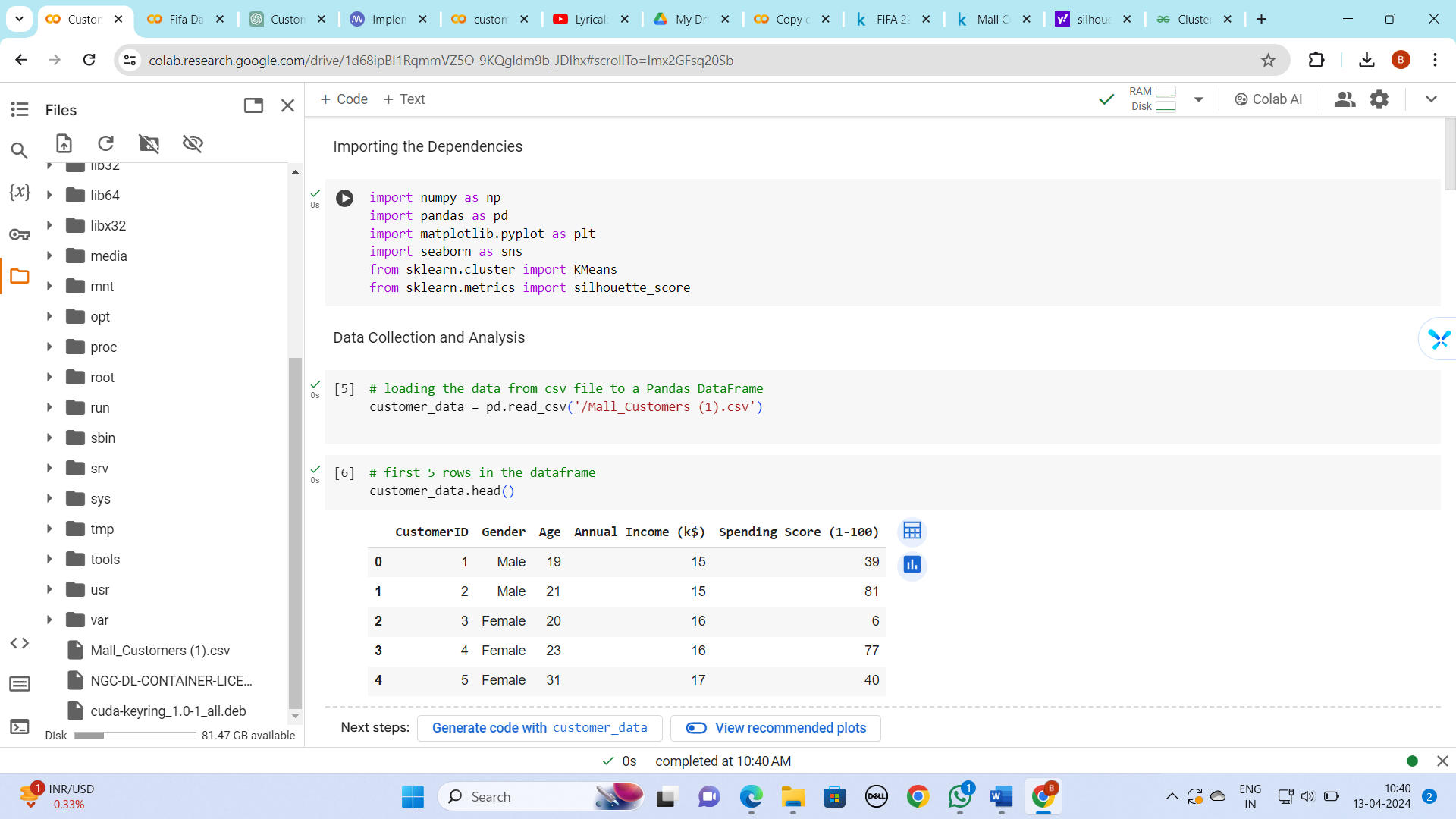
from sklearn.cluster import KMeans

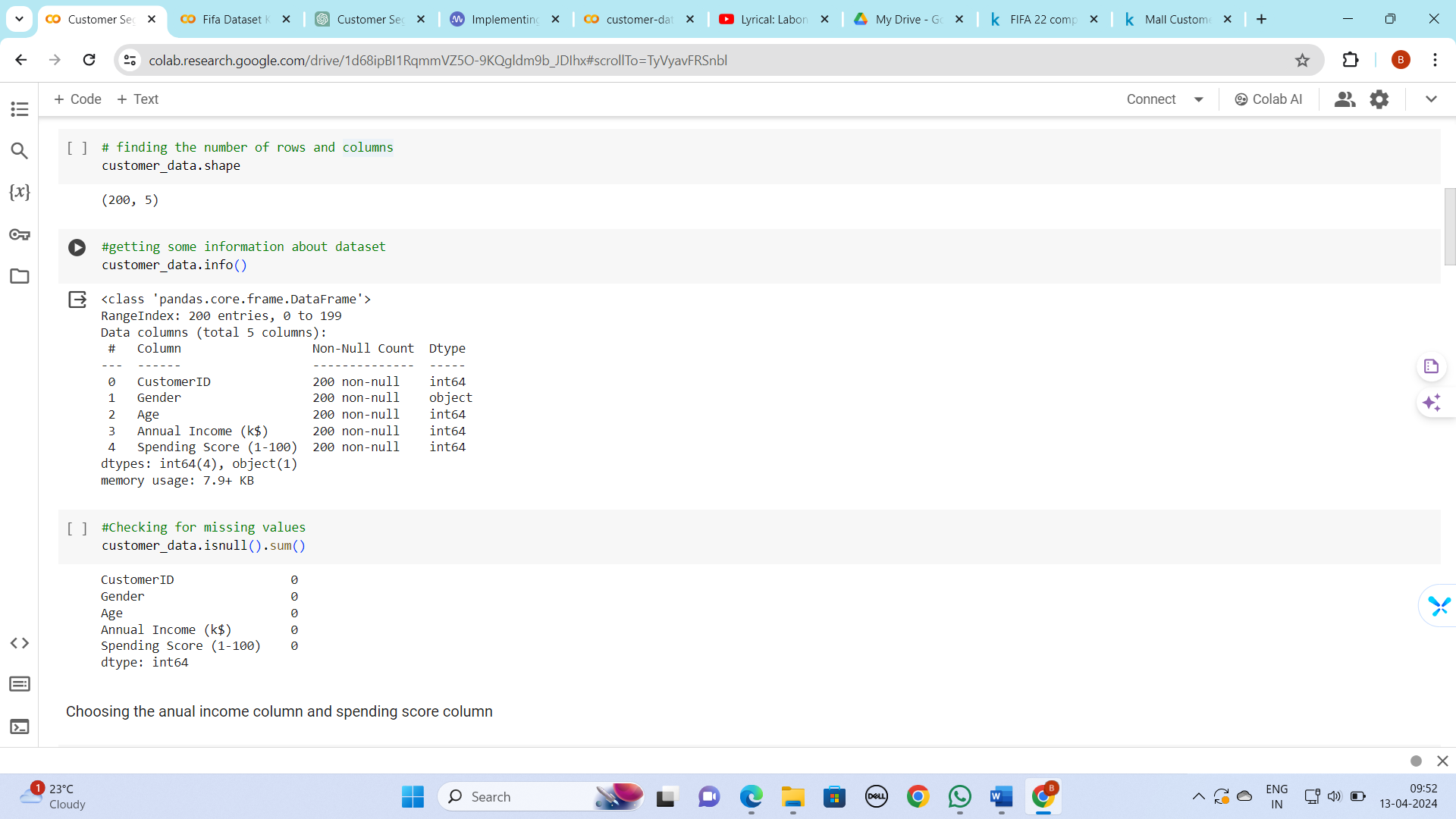
1. **NumPy (np)**: NumPy is a fundamental package for scientific computing in Python. It provides support for arrays, matrices, and mathematical functions, making it essential for numerical operation.
2. **Pandas (pd)**: Pandas is a powerful library for data manipulation and analysis. It offers data structures like DataFrame and Series, along with functions for reading, writing, and analyzing data.
3. **Matplotlib (plt)**: Matplotlib is a widely-used plotting library in Python. It enables the creation of various types of plots, including line plots, scatter plots, histograms, and more, making it indispensable for data visualization.
4. **Seaborn (sns)**: Seaborn is built on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics. It simplifies the process of creating complex visualizations and supports features like color palettes and thematic elements.
5. **scikit-learn's KMeans**: scikit-learn is a machine learning library in Python. The KMeans class from sklearn.cluster implements the K-Means clustering algorithm, which is useful for unsupervised clustering tasks such as customer segmentation.

**Code**

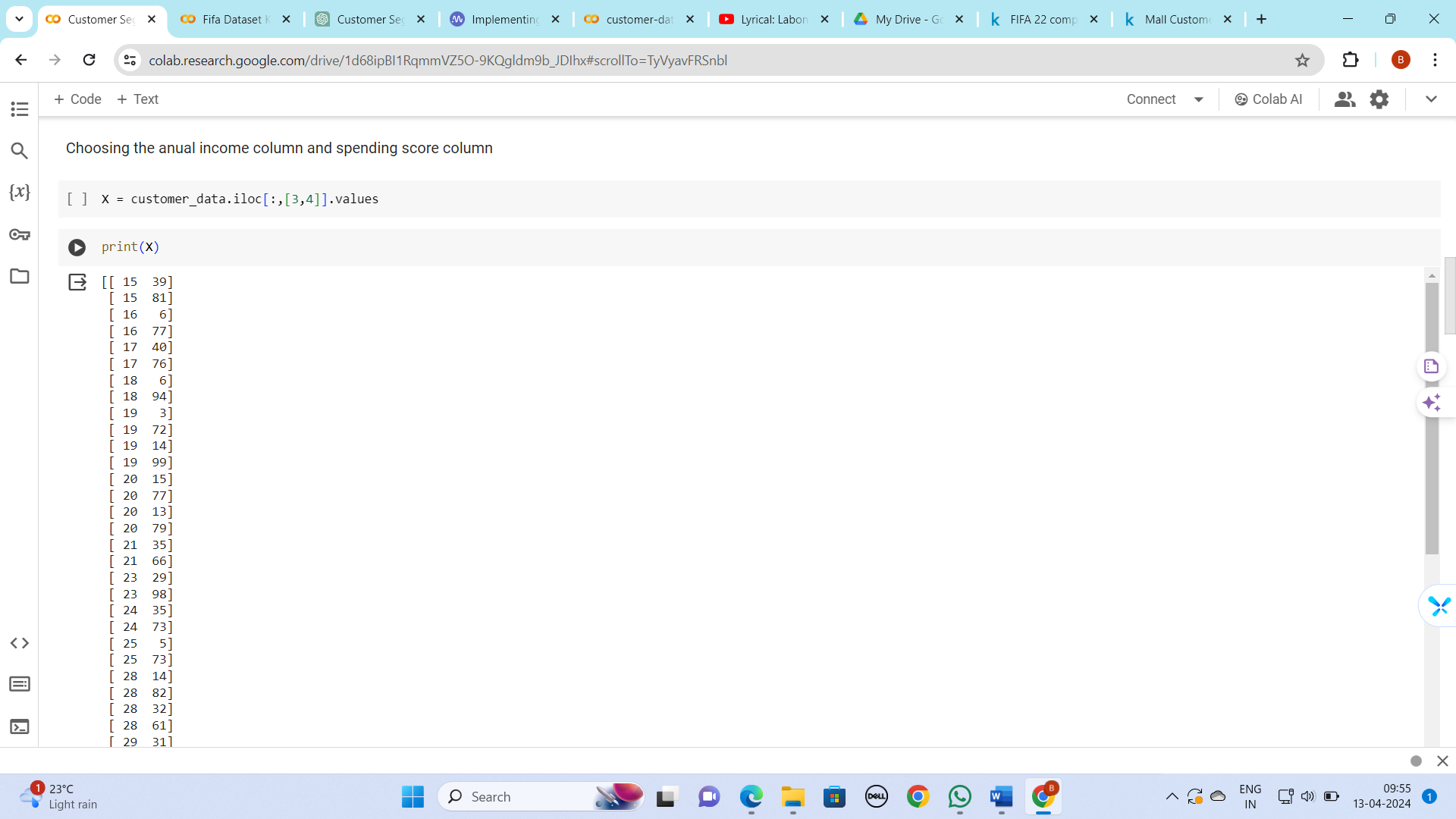
<https://colab.research.google.com/drive/1d68ipBI1RqmmVZ5O-9KQgldm9b_JDIhx?usp=sharing>

1. initially import the libraries.
2. Load the dataset



3)Preprocess the dataset 

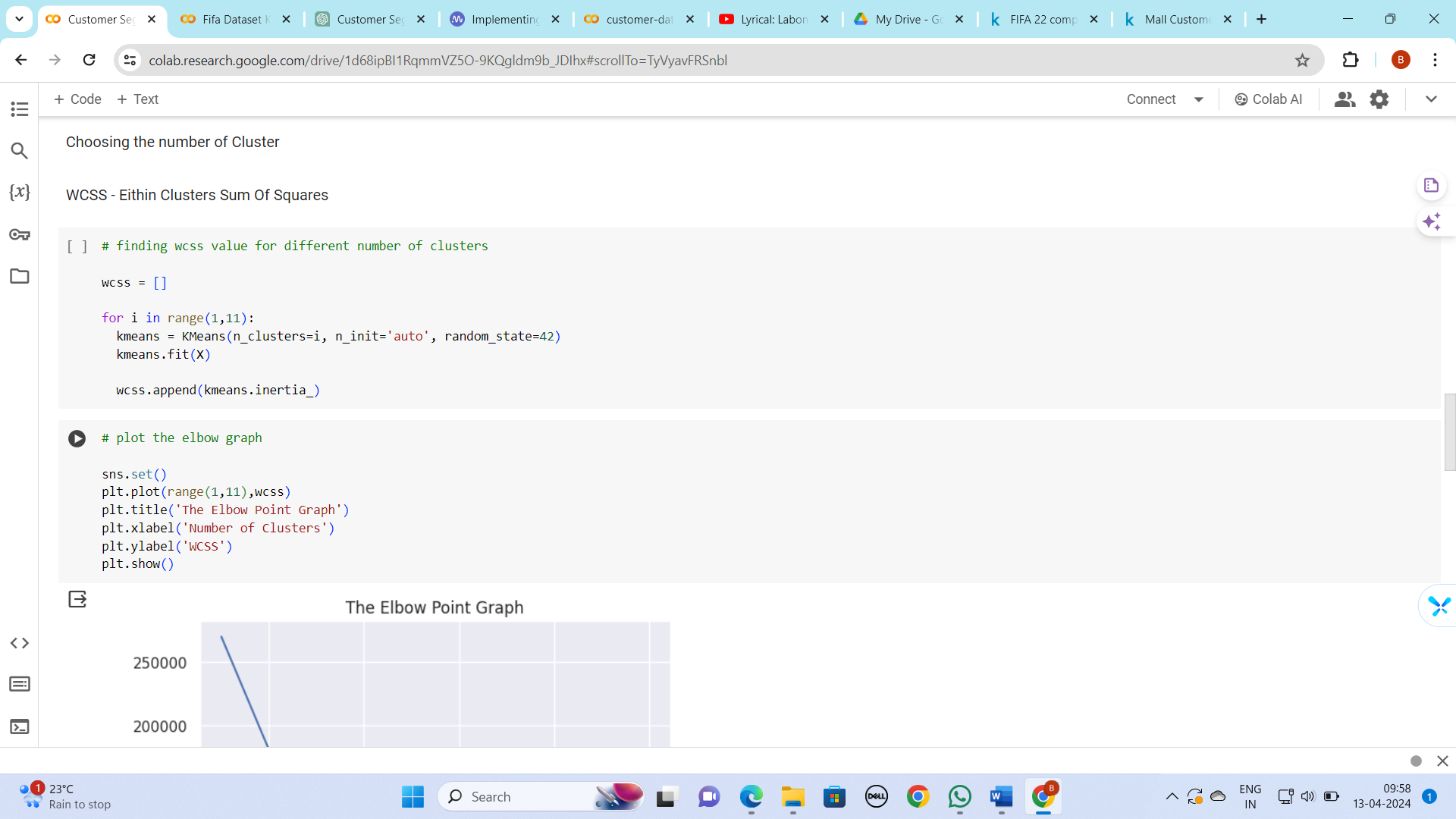
1. Choose spending score and annual income attributes for clustering



1. Finding the optimal number of clusters (value k)

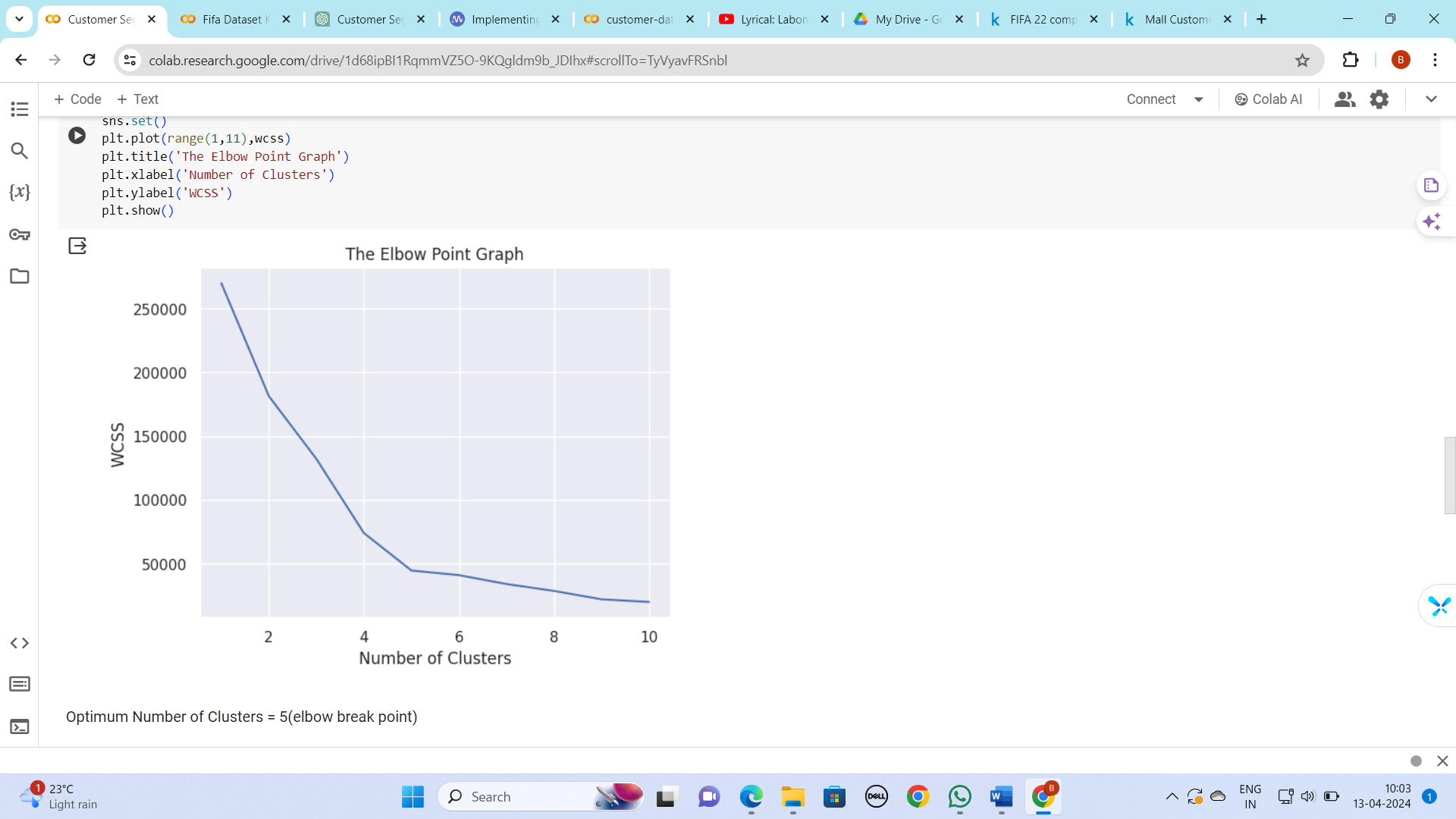
For this we implement elbow method

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

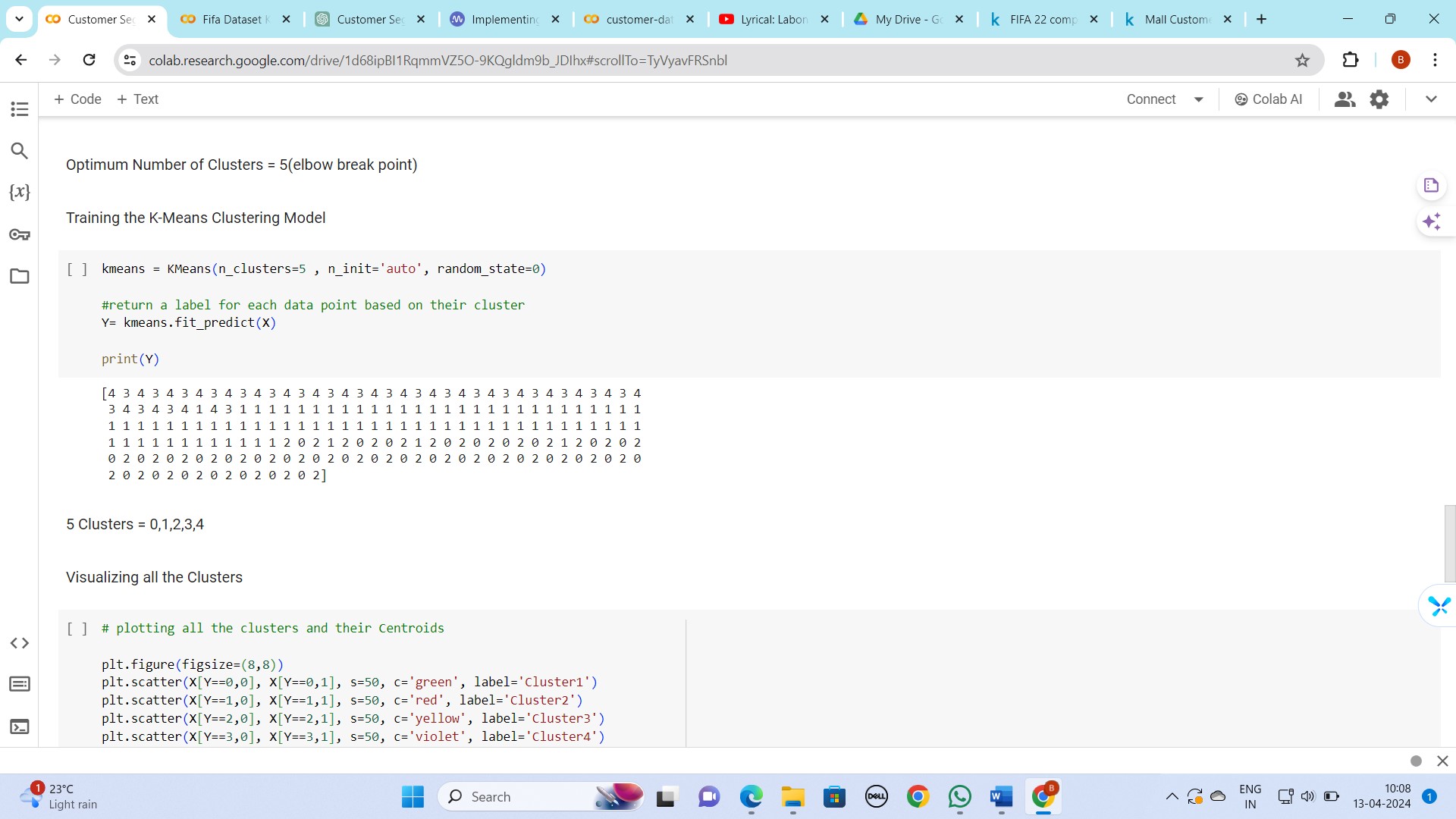


I gradually increase the number of clusters from 1 to 11 and breakpoint here in this graph is at 2, 4, 5 and after 5 it won’t change so elbow point in my case is 5.

Optimal number of clusters = k = 5.

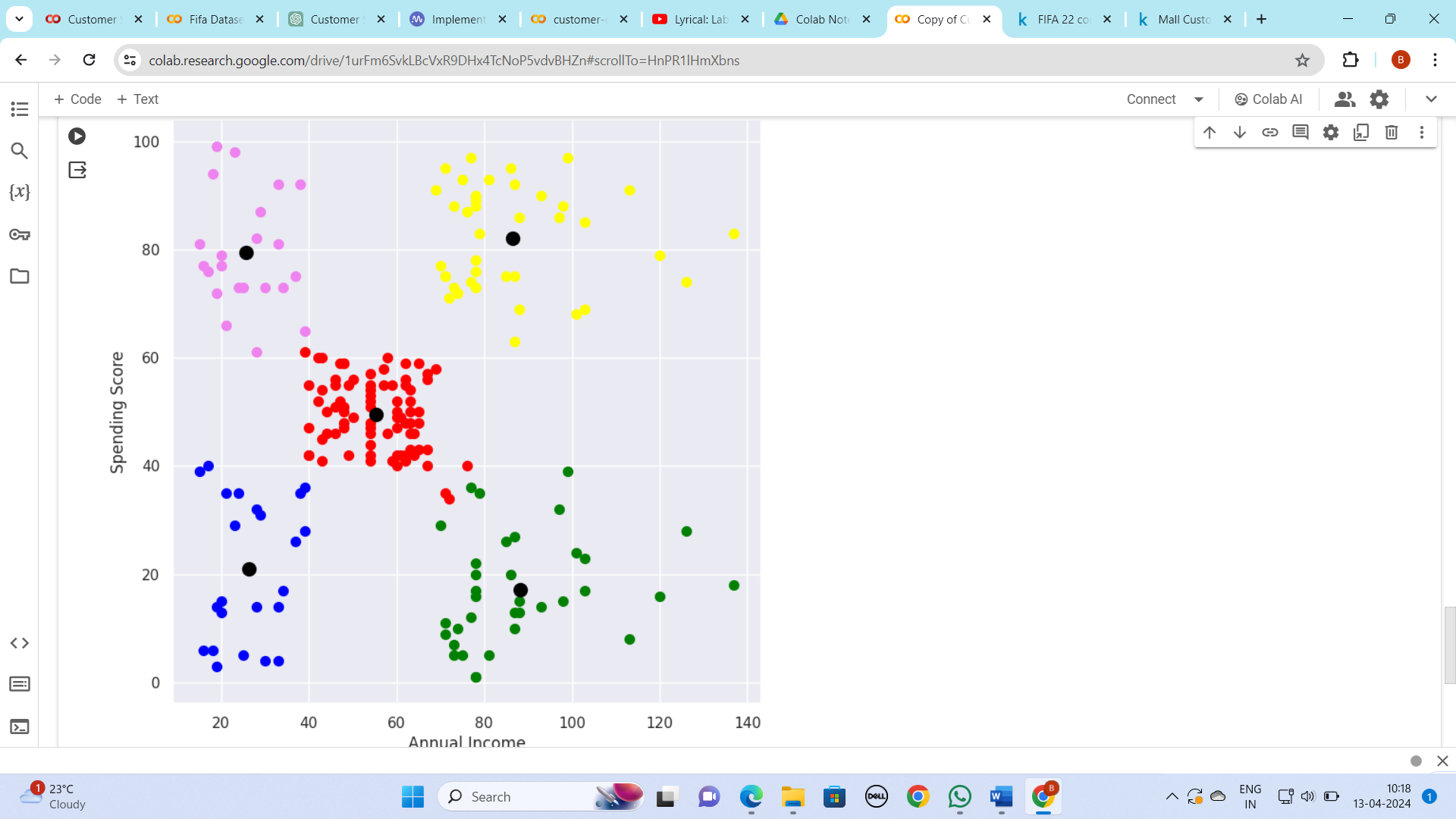


1. Train the Data



8)Visualize the data



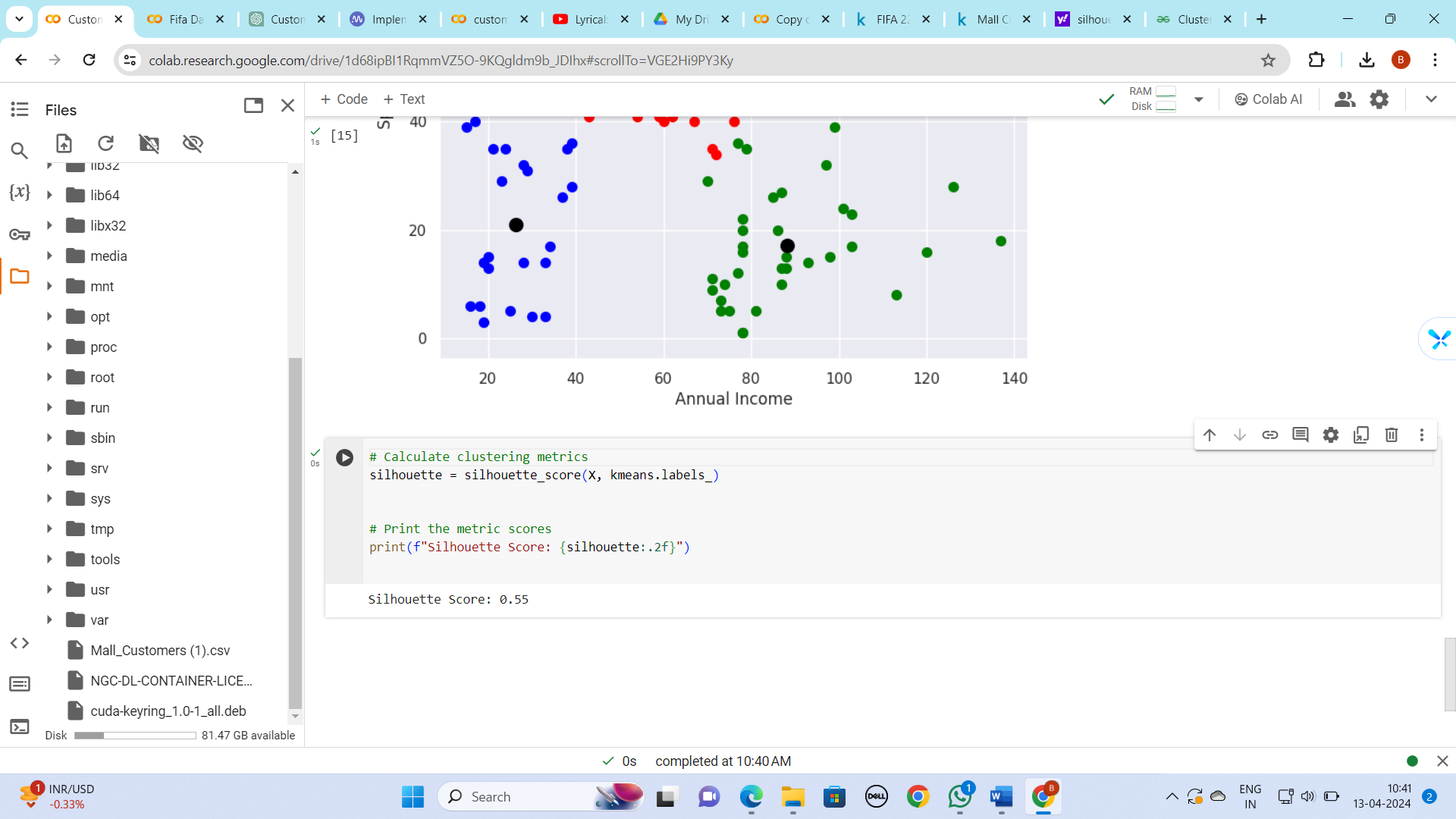


9) Performance Matrix

## Silhouette Score

A metric called the Silhouette Score is employed to assess a dataset’s well-defined clusters. The cohesiveness and separation between clusters are quantified. Better-defined clusters are indicated by higher scores, which range from -1 to 1. An object is said to be well-matched to its own cluster and poorly-matched to nearby clusters if its score is close to 1. A score of about -1, on the other hand, suggests that the object might be in the incorrect cluster. The Silhouette Score is useful for figuring out how appropriate clustering methods are and how many clusters are best for a particular dataset.

**Silhouette Score (0.55):** This score reveals how similar data points are inside their clusters when compared to data points from other clusters. A result of 0.55 indicates that there is some separation between the clusters, but there is still space for improvement. Closer to 1 values suggest better-defined clusters.



**Advantages of K-means Clustering Algorithm**

1. **Simple and Easy to Implement**: K-means is straightforward to understand and implement, making it suitable for large datasets.
2. **Efficient**: It is computationally efficient and works well even with high-dimensional data.
3. **Scalability**: K-means can handle large datasets efficiently, making it suitable for real-world applications.
4. **Interpretability**: The clusters formed by K-means are easy to interpret and can provide valuable insights into customer behaviour.

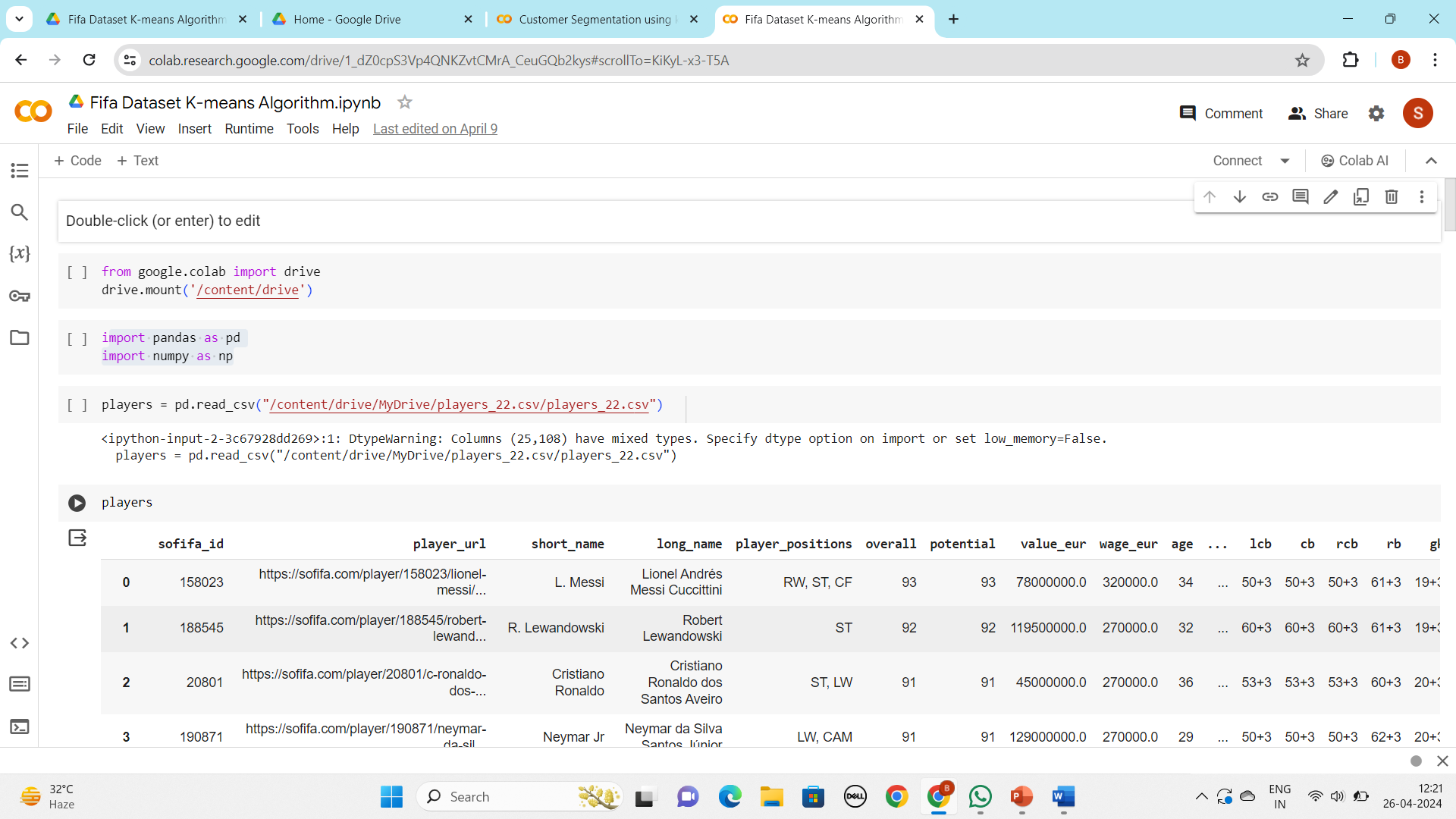
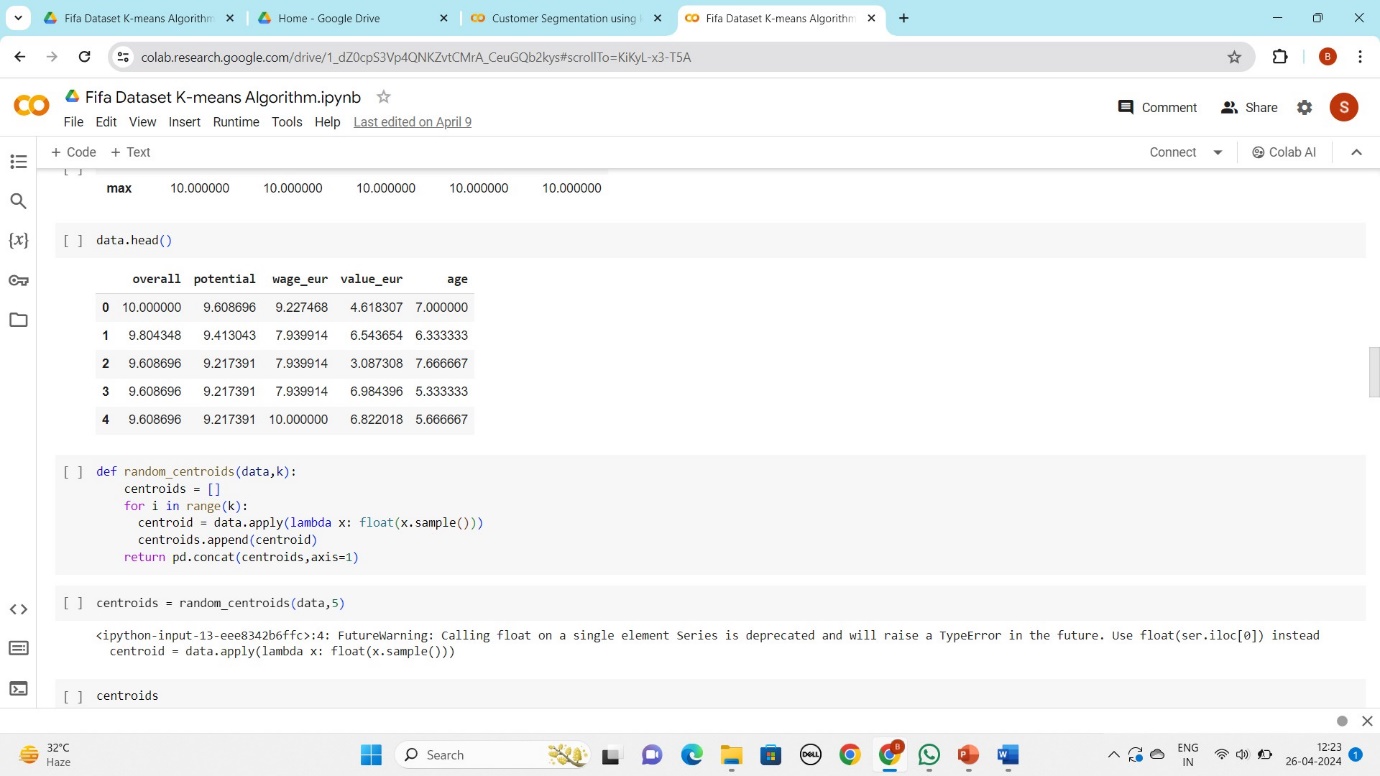
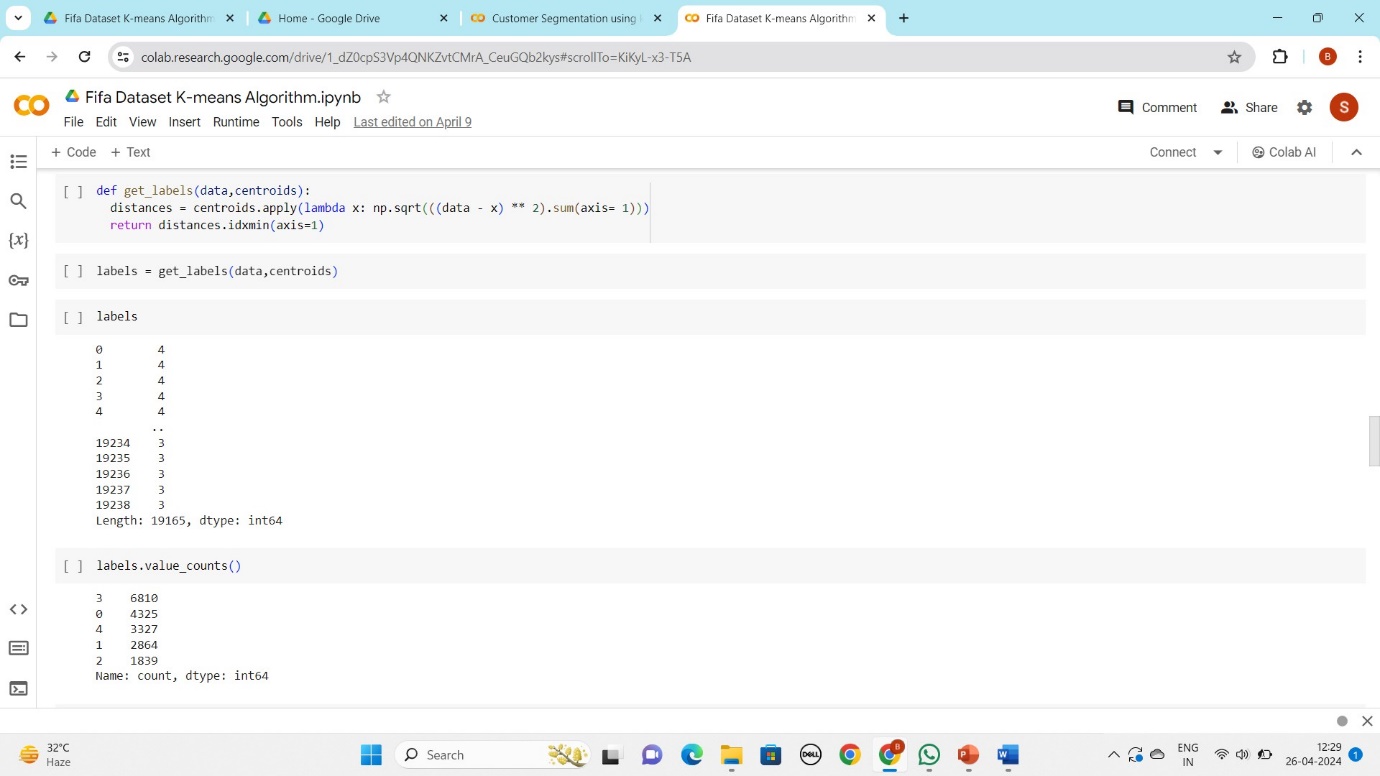
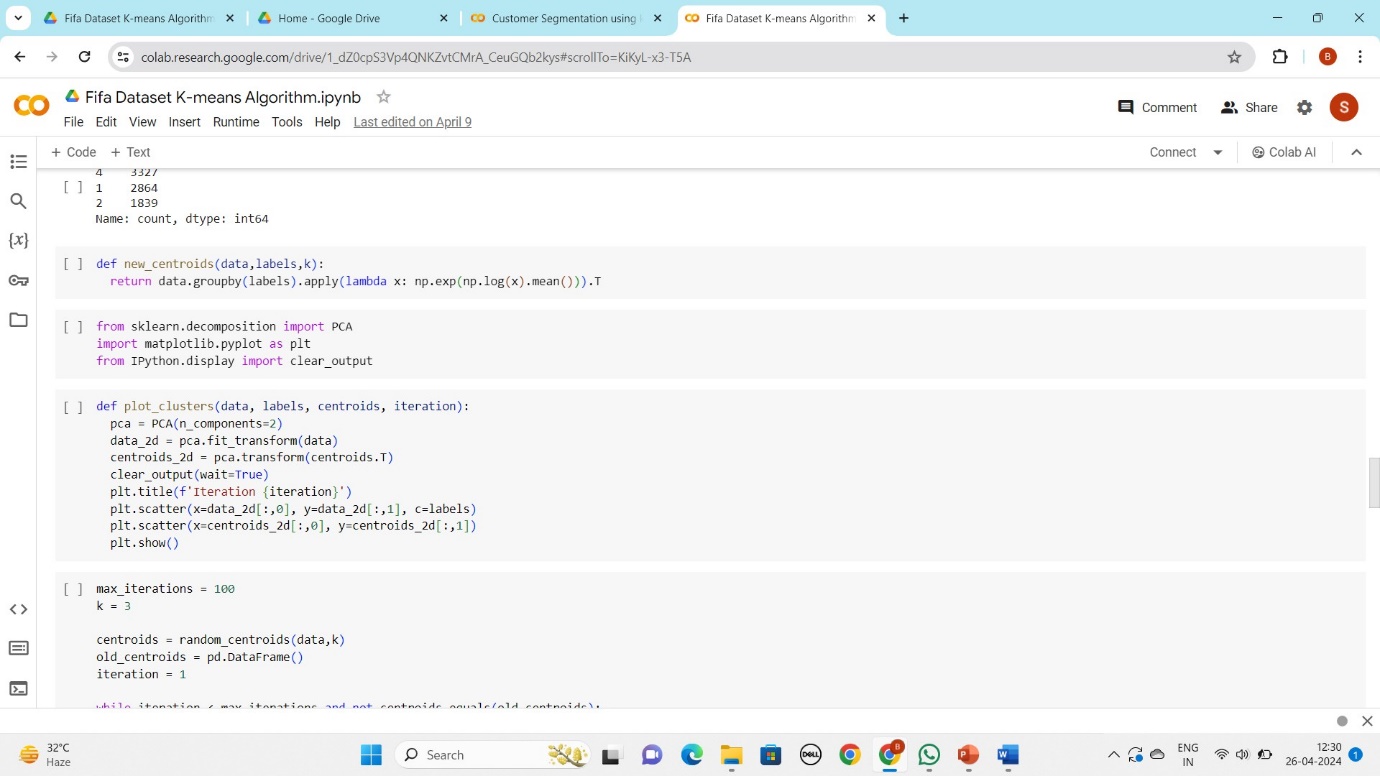
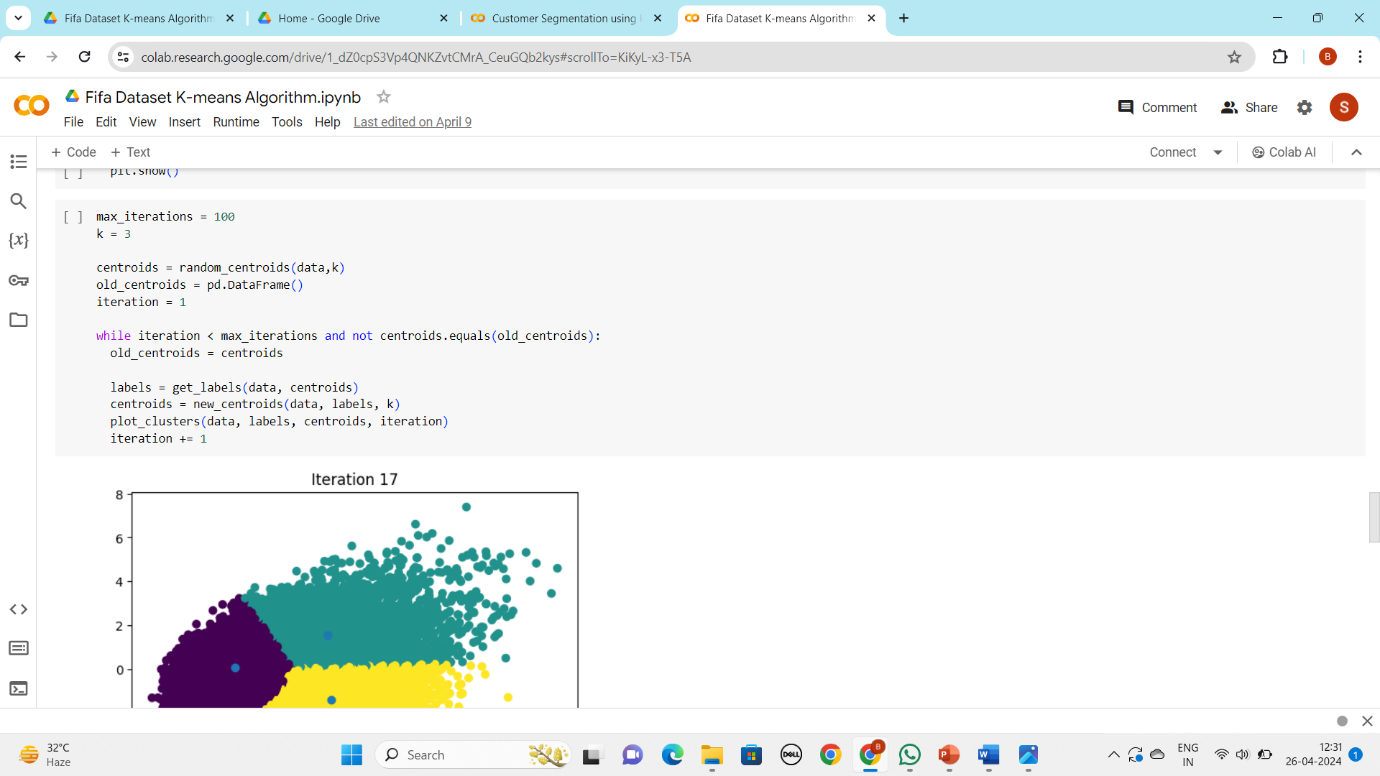
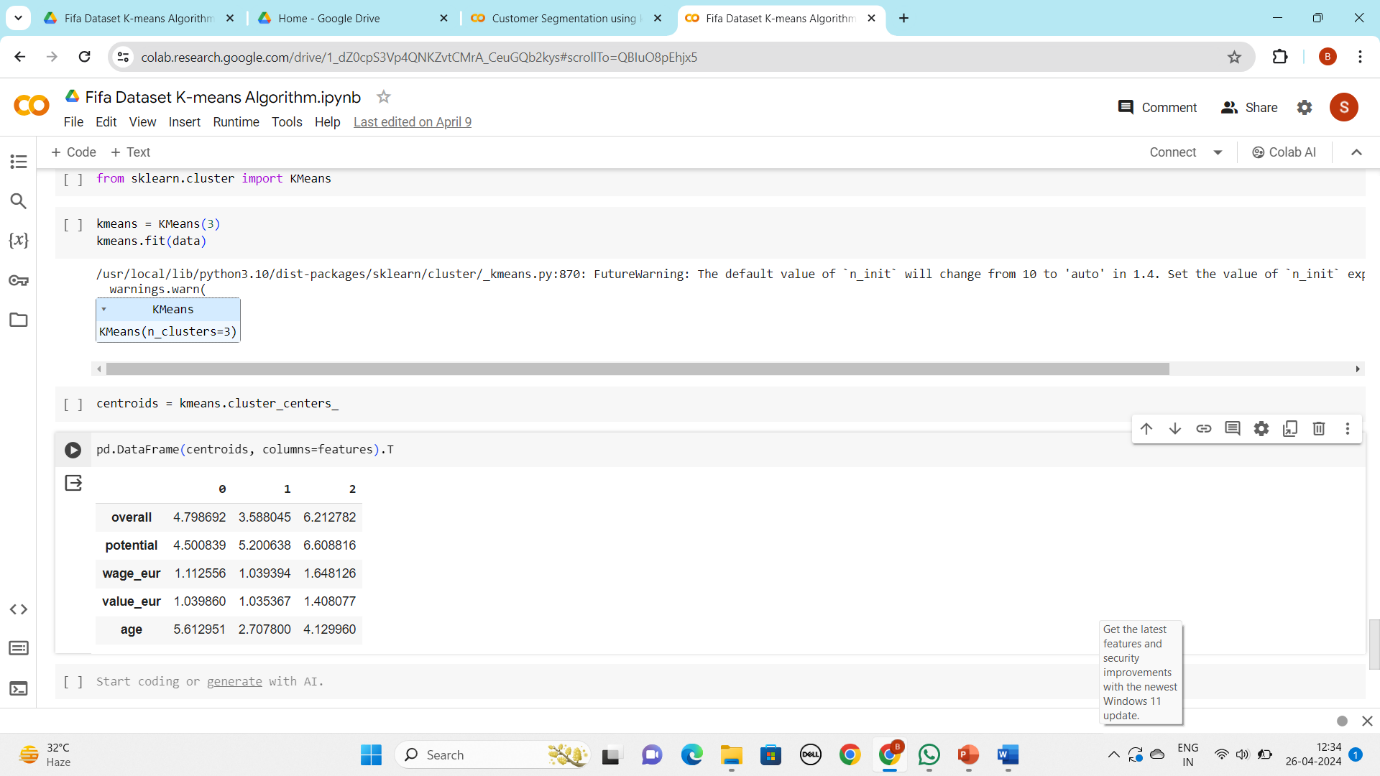
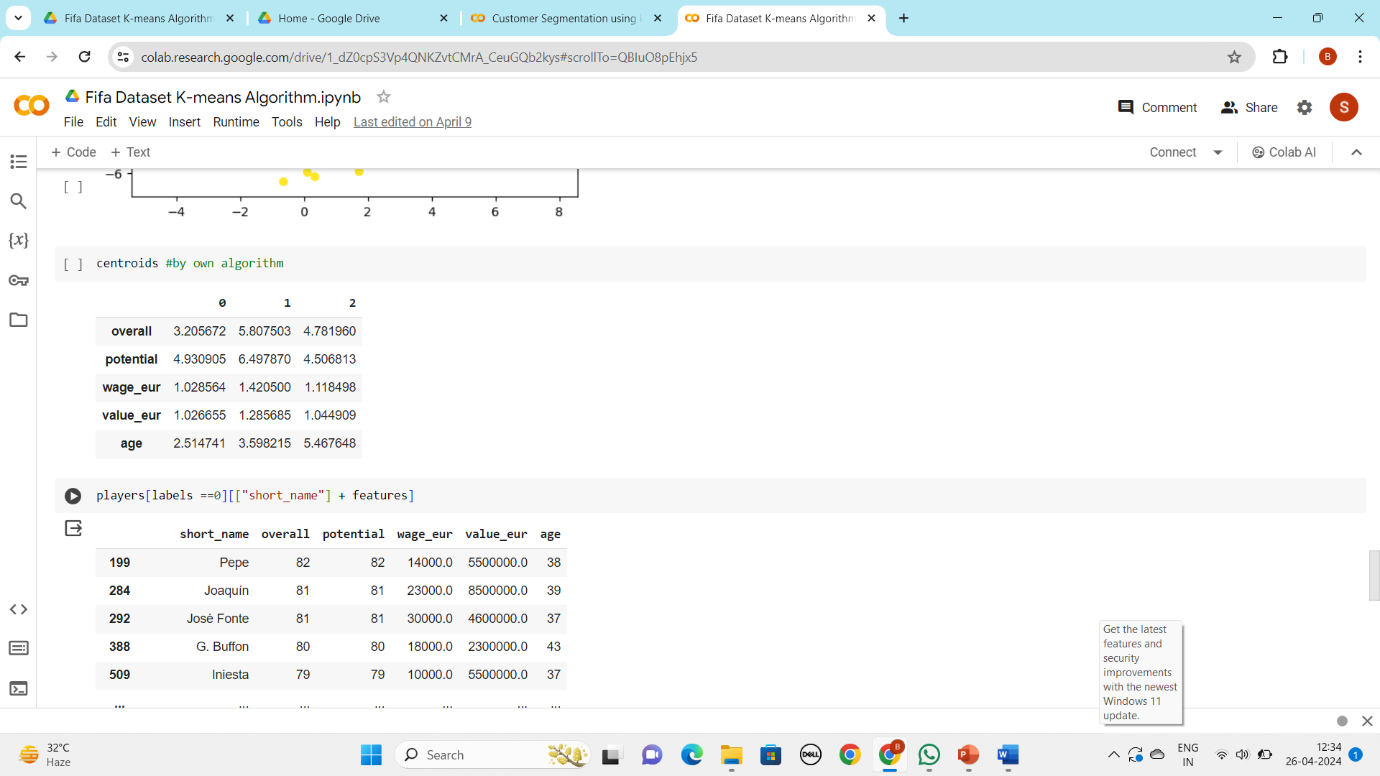
**Disadvantages of K-means Clustering Algorithm**

1. **Sensitive to Initial Centroid Selection**: K-means results can vary based on the initial selection of centroids, leading to suboptimal clustering.
2. **Requires Predefined Number of Clusters (K)**: The number of clusters (K) needs to be specified beforehand, which can be challenging if the optimal number of clusters is unknown.
3. **Assumes Spherical Clusters**: K-means assumes that clusters are spherical and of similar sizes, which may not always hold true in real-world data.

Using the silhouette score, we can assess the effectiveness of K-means clustering for customer segmentation by comparing different clustering solutions and selecting the one with the highest silhouette score.

**Additional Work**

I implement **K – means Clustering Algorithm from Scratch** to segment FIFA 22 Players based on their attributes.

1. **Data Loading and Preprocessing**:
   * The code begins by loading the player data from a CSV file and selecting specific features such as overall rating, potential, wage, value, and age.
   * It then drops any rows with missing values in the selected features.
   * Next, the data is scaled using Min-Max scaling to ensure all features have the same influence on the clustering process.
   * 
2. **Random Initialization of Centroids**:
   * The **random\_centroids** function randomly selects data points as initial centroids for each cluster.
3. 
4. **Assigning Data Points to Nearest Centroids**:
   * The **get\_labels** function calculates the Euclidean distance between each data point and the centroids, assigning each point to the nearest centroid.
5. 
6. **Updating Centroids**:
   * The **new\_centroids** function recalculates the centroids based on the mean of the data points assigned to each cluster.
7. 
8. **Plotting Clusters**:
   * The **plot\_clusters** function uses Principal Component Analysis (PCA) to reduce the dimensionality of the data and visualizes the clusters and centroids in a 2D plot.
9. **K-means Clustering Implementation**:
   * The code then implements K-means clustering using the custom-defined functions.
   * It iteratively updates the centroids until convergence or the maximum number of iterations is reached, plotting the clusters at each iteration.
   * 
10. 
11. **Comparison with sklearn's KMeans**:
    * Finally, the code uses sklearn's KMeans implementation to perform the clustering and compares the centroids obtained from both implementations.

Dataset – <https://www.kaggle.com/datasets/stefanoleone992/fifa-22-complete-player-dataset?select=players_22.csv>

Code- <https://colab.research.google.com/drive/1_dZ0cpS3Vp4QNKZvtCMrA_CeuGQb2kys?usp=sharing>

**Thank You**