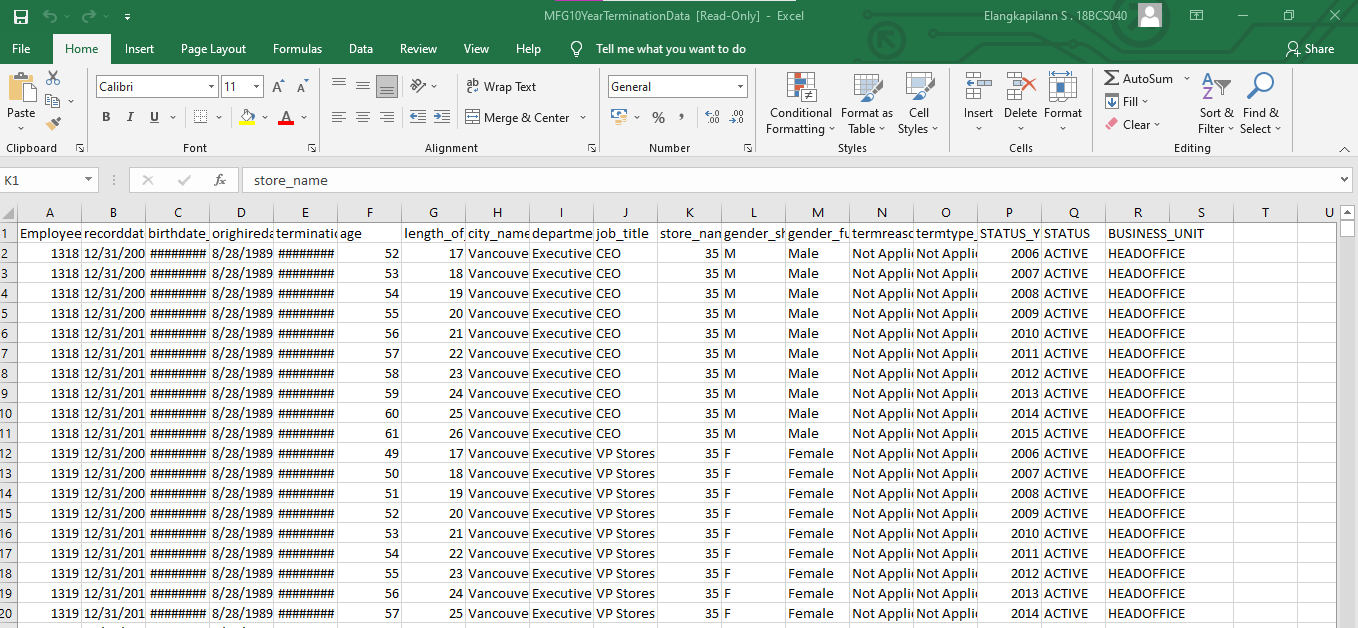
**DWDM** -**SEM PRACTICAL**

**NAME: ELANGKAPILANN S**

**ROLL.NO: 18BCS040**

**DATE : 23-06-2021**

1. Download a sample dataset from any Repository. List the attributes and its type in a word Doc ****

ATTREBUTES TYPE

1.EmployeeID - NUMERIC

2.recorddate\_key - NOMINAL

3. birthdate\_key - NOMINAL

4. orighiredate\_key - NOMINAL

5. terminationdate\_key - NOMINAL

6.age - RATIO SCALED

7. length\_of\_service - RATIO SCALED

8. city\_name - ORDINAL

9. department\_name - NOMINAL

10. job\_title - NOMINAL

11. store\_name - NOMINAL

12. gender\_short - SYMMETRIC BINARY

13. gender\_full - SYMMENTRIC BINARY

14. termreason\_desc - NOMINAL

15. STATUS\_YEAR - RATIO SACLED

16. STATUS - ASYMMETRIC BINARY

2)Create a random dataset of 50 elements with x and y variables using random function between 30 to 100 integers for x and 60 to 150 integers for y. Apply K- means clustering to cluster the data into 3 clusters. Plot the graph and display the result. Use Tkinter GUI to Display the Results

import numpy as np

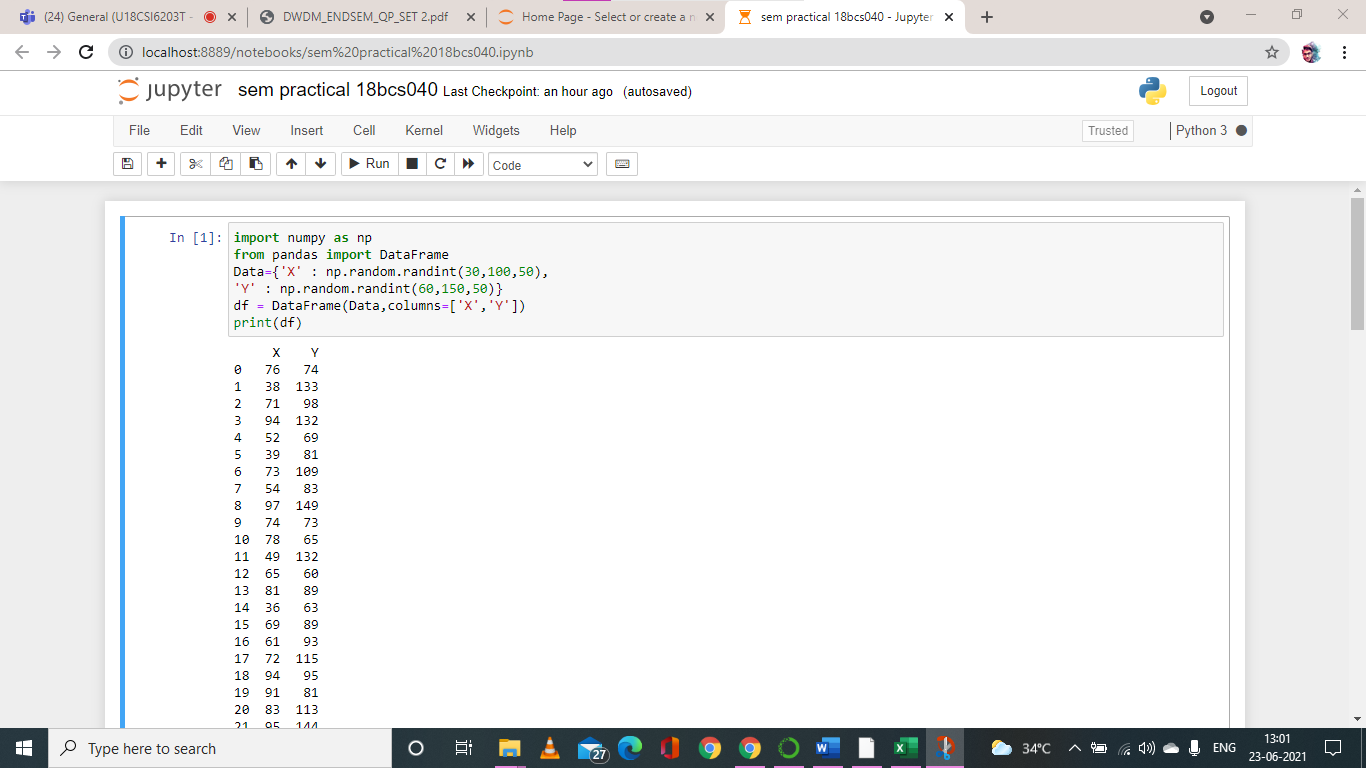
from pandas import DataFrame

Data={'X' : np.random.randint(30,100,50),

'Y' : np.random.randint(60,150,50)}

df = DataFrame(Data,columns=['X','Y'])

print(df)



import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

kmeans = KMeans(n\_clusters=3).fit(df)

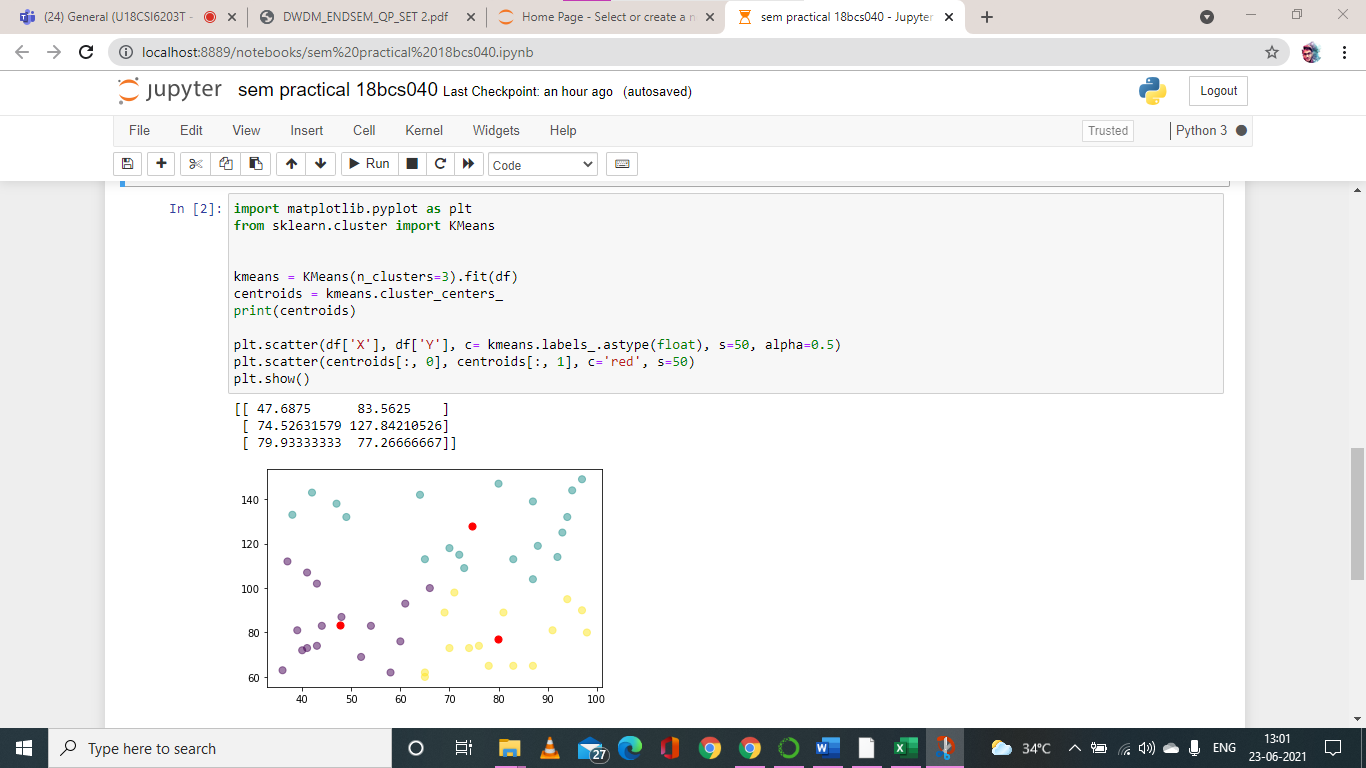
centroids = kmeans.cluster\_centers\_

print(centroids)

plt.scatter(df['X'], df['Y'], c= kmeans.labels\_.astype(float), s=50, alpha=0.5)

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)

plt.show()



import tkinter as tk

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

root= tk.Tk()

canvas1 = tk.Canvas(root, width = 100, height = 100)

canvas1.pack()

label1 = tk.Label(root, text=centroids, justify = 'center')

canvas1.create\_window(80, 60, window=label1)

figure1 = plt.Figure(figsize=(5,4), dpi=100)

ax1 = figure1.add\_subplot(111)

ax1.scatter(df['X'], df['Y'], c= kmeans.labels\_.astype(float), s=50, alpha=0.5)

ax1.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)

scatter1 = FigureCanvasTkAgg(figure1, root)

scatter1.get\_tk\_widget().pack(side=tk.LEFT, fill=tk.BOTH)

root.mainloop()

