SEMESTER PRACTICALS

DATA WAREHOUSING AND DATA MINING

U18CSI6203L

Roll No: 18BCS038

Name: PRAGATHI R

Experiment:

Create a random dataset of 30 elements with x and y variables using random function between 30 to 80 integers for x and 60 to 100 integers for y.

Apply K- means clustering to cluster the data into 2 clusters. Plot the graph and display the result.

Use Tkinter GUI to Display the Results.

Dataset Description:

Dataset: Diabetes dataset.

Link for the Dataset:

https://www.kaggle.com/uciml/pima-indians-diabetes-database

Attribute Description:

No of Pregnancies - Numerical(discrete)

Glucose – Numerical(discrete)

Blood Pressure - Numerical(discrete)

Skin Thickness - Numerical(discrete)

Insulin - Numerical(discrete)

BMI - Numerical(discrete)

DiabetesPedigreeFunction – Numerical(discrete)

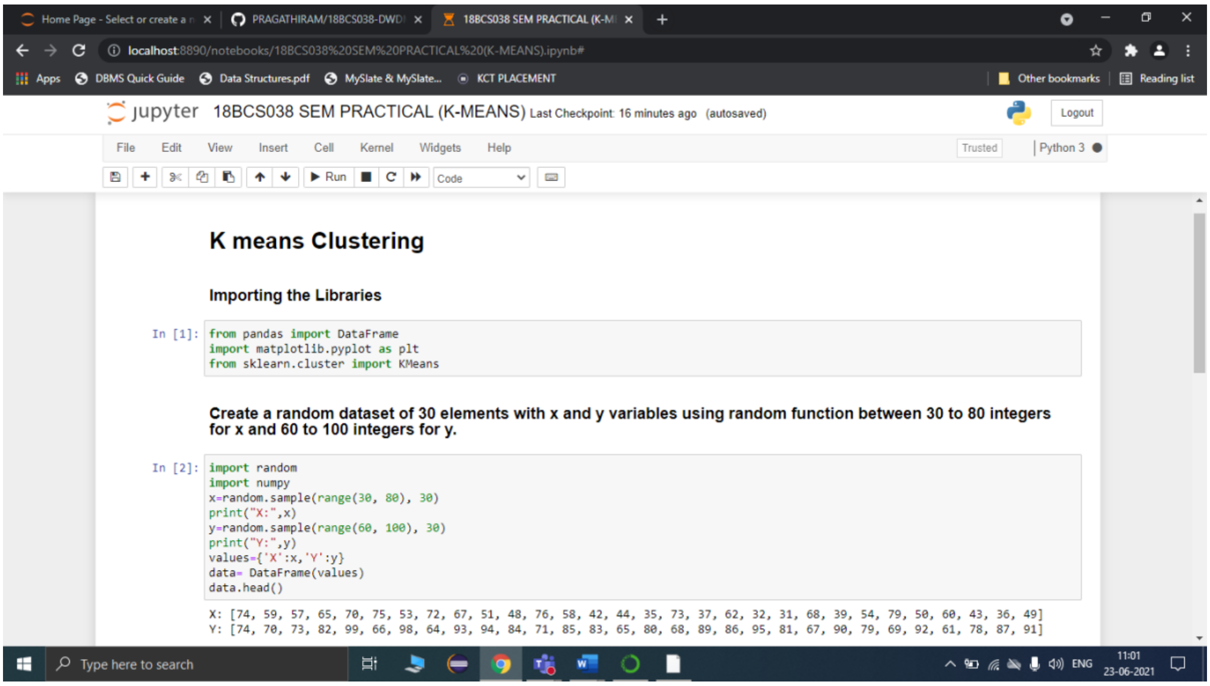
Age – Numerical(discrete)

Outcome – Numerical(categorical)

GitHub Link for code:

https://github.com/PRAGATHIRAM/18BCS038-DWDM-SEM-PRACTICAL

Output:



jupyter 18BCS038 SEM PRACTICAL (K-MEANS) Last Checkpoint: 16 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

K means Clustering

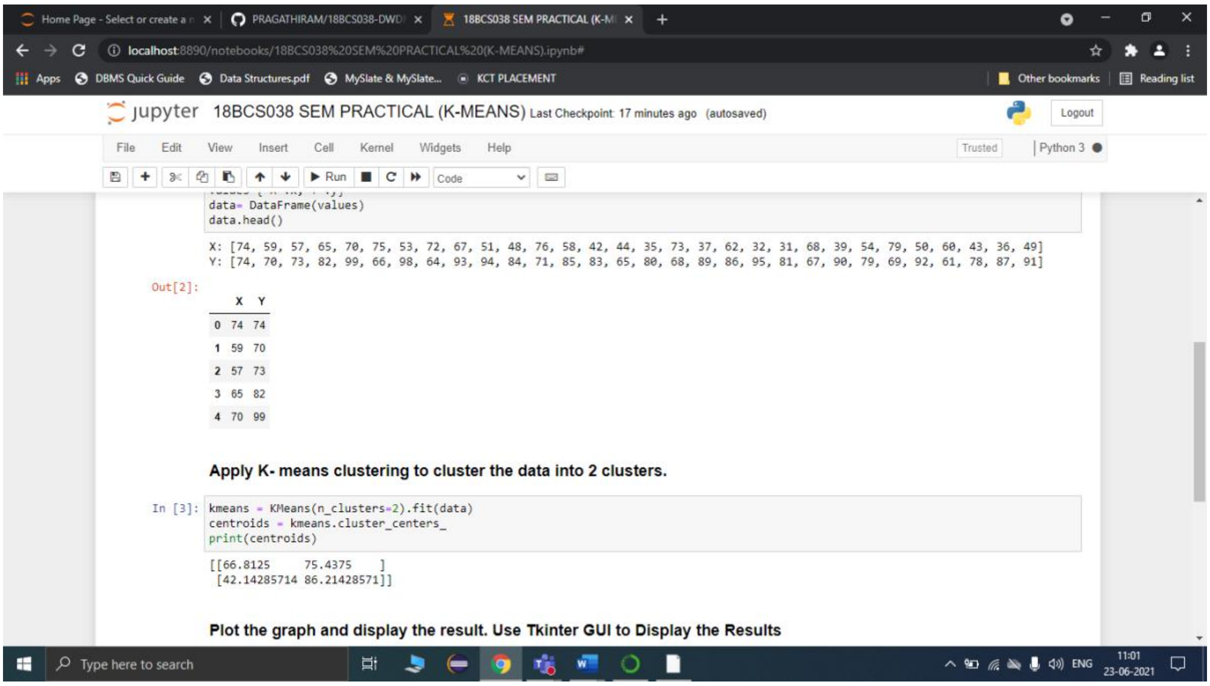
Importing the Libraries

```
In [1]: from pandas import DataFrame
        import matplotlib.pyplot as plt
        from sklearn.cluster import KMeans
```

Create a random dataset of 30 elements with x and y variables using random function between 30 to 80 integers for x and 60 to 100 integers for y.

```
In [2]: import random
        import numpy
        x=random.sample(range(30, 80), 30)
        print("X:",x)
        y=random.sample(range(60, 100), 30)
        print("Y:",y)
        values={'X':x,'Y':y}
        data= DataFrame(values)
        data.head()
```

```
X: [74, 59, 57, 65, 70, 75, 53, 72, 67, 51, 48, 76, 58, 42, 44, 35, 73, 37, 62, 32, 31, 68, 39, 54, 79, 50, 60, 43, 36, 49]
Y: [74, 70, 73, 82, 99, 66, 98, 64, 93, 94, 84, 71, 85, 83, 65, 80, 68, 89, 86, 95, 81, 67, 90, 79, 69, 92, 61, 78, 87, 91]
```



```
data= DataFrame(values)
data.head()
```

```
X: [74, 59, 57, 65, 70, 75, 53, 72, 67, 51, 48, 76, 58, 42, 44, 35, 73, 37, 62, 32, 31, 68, 39, 54, 79, 50, 60, 43, 36, 49]
Y: [74, 70, 73, 82, 99, 66, 98, 64, 93, 94, 84, 71, 85, 83, 65, 80, 68, 89, 86, 95, 81, 67, 90, 79, 69, 92, 61, 78, 87, 91]
```

Out[2]:

	X	Y
0	74	74
1	59	70
2	57	73
3	65	82
4	70	99

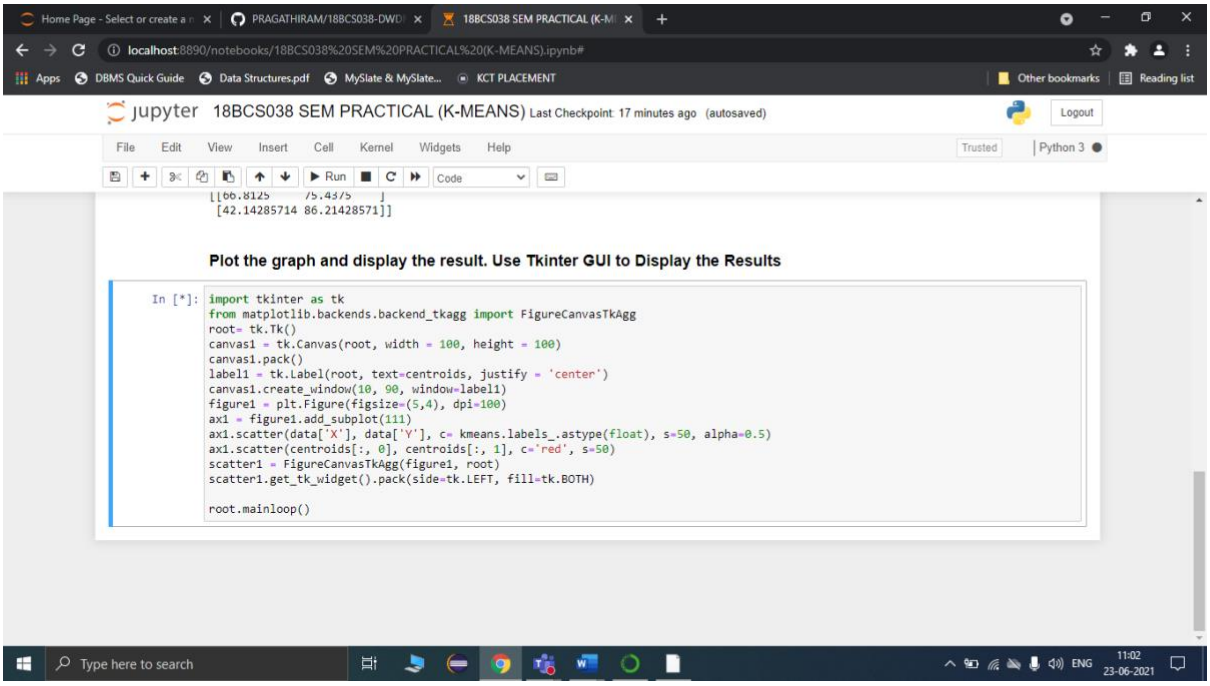
Apply K- means clustering to cluster the data into 2 clusters.

In [3]:

```
kmeans = KMeans(n_clusters=2).fit(data)
centroids = kmeans.cluster_centers_
print(centroids)
```

```
[[66.8125     75.4375    ]
 [42.14285714 86.21428571]]
```

Plot the graph and display the result. Use Tkinter GUI to Display the Results



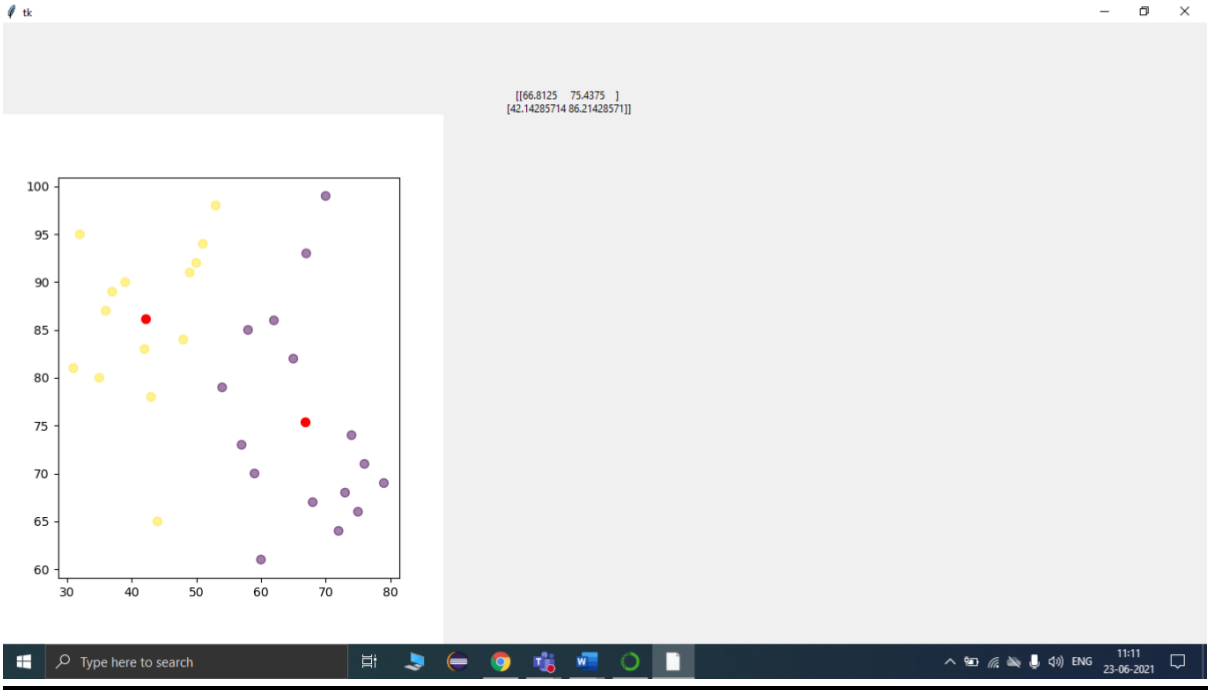
```
[[66.8125     75.4375    ]
 [42.14285714 86.21428571]]
```

Plot the graph and display the result. Use Tkinter GUI to Display the Results

In [*]:

```
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(10, 90, window=label1)
figure1 = plt.Figure(figsize=(5,4), dpi=100)
ax1 = figure1.add_subplot(111)
ax1.scatter(data['X'], data['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()
```



```
[[66.8125    75.4375    ]
 [42.14285714 86.21428571]]
```