<u>DWDM MODEL LAB</u>

PRAVEEN KUMAR V – 18BCS030

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

ATTRIBUTE	TYPE
sl_no	NUMERIC
gender	SYMMETRIC BINARY
ssc_p	NUMERIC
ssc_b	NOMINAL
hsc_p	NUMERIC
Hsc_b	NOMINAL
Degree_p	NUMERIC
Degree_t	NOMINAL
workex	ASYMMETRIC
Etest_p	NUMERIC
specialisation	NOMINAL
Mba_p	NUMERIC
status	ASYMMETRIC BINARY
salary	NUMERIC
Hsc_s	NOMINAL

2.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.

In [21]:
```python
import numpy as np
from pandas import DataFrame
Data={'X' : np.random.randint(30,80,30),
'Y' : np.random.randint(60,100,30)}
df = DataFrame(Data,columns=['X','Y'])
print(df)
```

```
     X   Y
0   35  80
1   64  82
2   47  76
3   65  82
4   64  74
5   45  78
6   39  67
7   44  89
8   53  89
9   78  77
10  60  97
11  73  82
12  72  66
13  52  86
14  30  85
15  59  66
16  69  95
17  70  80
18  45  63
19  35  79
20  46  70
21  52  65
22  50  75
23  78  89
24  73  86
25  49  70
26  76  68
27  56  79
28  64  83
29  40  61
```

```python
In [22]: import matplotlib.pyplot as plt
         from sklearn.cluster import KMeans


         kmeans = KMeans(n_clusters=2).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()
```

```
[[45.70588235 75.17647059]
 [69.69230769 81.61538462]]
```
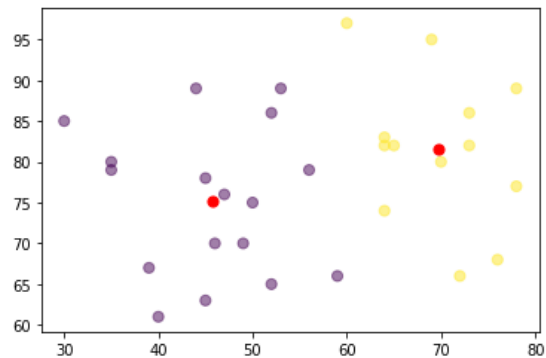


```python
In [13]: import tkinter as tk
         from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg


         root= tk.Tk()

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

         label1 = tk.Label(root, text=centroids, justify = 'center')
         canvas1.create_window(70, 50, 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()
```

Jupyter MODELLAB Last Checkpoint: 5 minutes ago (autosaved)

File   Edit   View   Insert   Cell   Kernel   Widgets   Help
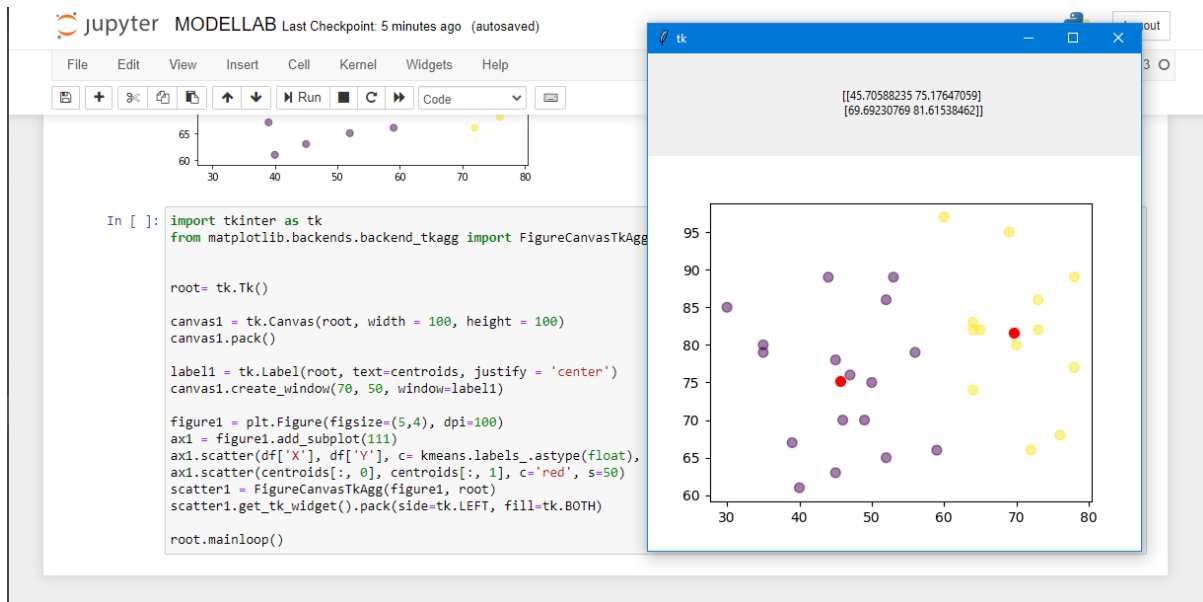
```python
In [ ]: import tkinter as tk
        from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

        root= tk.Tk()

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

        label1 = tk.Label(root, text=centroids, justify = 'center')
        canvas1.create_window(70, 50, 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),
        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()
```

tk
[[45.70588235 75.17647059]
[69.69230769 81.61538462]]

3.Upload in your github account. Provide the link for access.

https://github.com/Praveenvidor/Dataset/blob/main/Placement_Data_Full_Class.csv