KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Artificial Intelligence Lab

KCA 351: Session 2021-22

Experiment – No-12

Problem Statement : **Write a program to implement k-mean clustering problem. Consider the following two-dimensional dataset.**

x	y
25	79
34	51
22	53
27	78
33	59
33	74
31	73
22	57
35	69
34	75
67	51
54	32
57	40
43	47
50	53
57	36
59	35
52	58
65	59
47	50
49	25
48	20
35	14
33	12
44	20
45	5
38	29
43	27
51	8
46	7

Hint: Start by importing the required following libraries:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
```

Program:

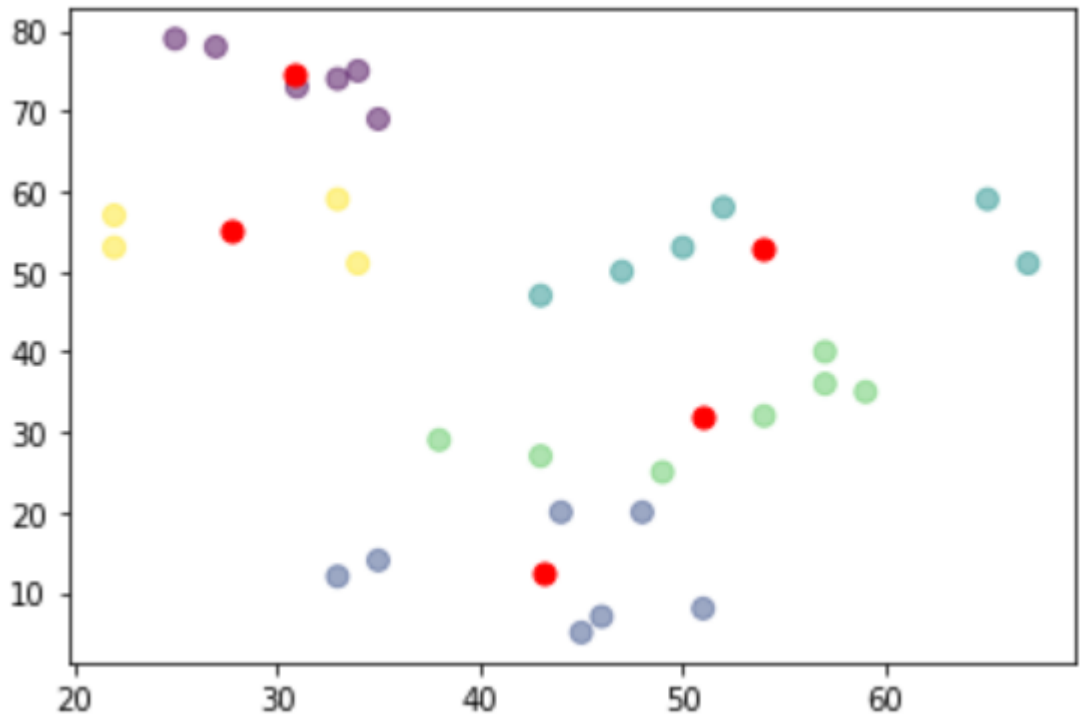
```
from pandas import DataFrame

Data = {'x': [25,34,22,27,33,33,31,22,35,34,67,54,57,43,50,57,59,52,65,47,
49,48,35,33,44,45,38,43,51,46],
        'y': [79,51,53,78,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,
25,20,14,12,20,5,29,27,8,7]
       }

df = DataFrame(Data,columns=['x','y'])
print (df)

import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=5).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()

from pandas import DataFrame
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
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()

import tkinter as tk
from tkinter import filedialog
import pandas as pd
from pandas import DataFrame
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

root= tk.Tk()

canvas1 = tk.Canvas(root, width = 400, height = 300,  relief = 'raised')
canvas1.pack()

label1 = tk.Label(root, text='k-Means Clustering')
label1.config(font=('helvetica', 14))
canvas1.create_window(200, 25, window=label1)

label2 = tk.Label(root, text='Type Number of Clusters:')
label2.config(font=('helvetica', 8))
canvas1.create_window(200, 120, window=label2)

entry1 = tk.Entry (root)
canvas1.create_window(200, 140, window=entry1)

def getExcel ():

    global df
    import_file_path = filedialog.askopenfilename()
    read_file = pd.read_excel (import_file_path)
```

```
    df = DataFrame(read_file,columns=['x','y'])

browseButtonExcel = tk.Button(text=" Import Excel File ", command=getExcel
, bg='green', fg='white', font=('helvetica', 10, 'bold'))
canvas1.create_window(200, 70, window=browseButtonExcel)

def getKMeans ():
    global df
    global numberOfClusters
    numberOfClusters = int(entry1.get())

    kmeans = KMeans(n_clusters=numberOfClusters).fit(df)
    centroids = kmeans.cluster_centers_

    label3 = tk.Label(root, text= centroids)
    canvas1.create_window(200, 250, window=label3)

    figure1 = plt.Figure(figsize=(4,3), dpi=100)
    ax1 = figure1.add_subplot(111)
    ax1.scatter(df['x'], df['y'], c= kmeans.labels_.astype(float), s=50, a
lpha=0.5)
    ax1.scatter(centroids[:, 0], centroids[:, 1], c='red', s=50)
    scatter1 = FigureCanvasTkAgg(figure1, root)
    scatter1.get_tk_widget().pack(side=tk.RIGHT, fill=tk.BOTH)

processButton = tk.Button(text=' Process k-
Means ', command=getKMeans, bg='brown', fg='white', font=('helvetica', 10,
 'bold'))
canvas1.create_window(200, 170, window=processButton)

root.mainloop()
```

Output :

```
     x   y
0   25  79
1   34  51
2   22  53
3   27  78
4   33  59
5   33  74
6   31  73
7   22  57
8   35  69
9   34  75
10  67  51
11  54  32
12  57  40
13  43  47
14  50  53
15  57  36
16  59  35
17  52  58
18  65  59
19  47  50
20  49  25
21  48  20
22  35  14
23  33  12
24  44  20
25  45   5
26  38  29
27  43  27
28  51   8
29  46   7
[[30.83333333 74.66666667]
 [43.14285714 12.28571429]
 [54.         53.        ]
 [51.         32.        ]
 [27.75       55.        ]]
```


80
70
60
50
40
30
20
10
20
30
40
50
60