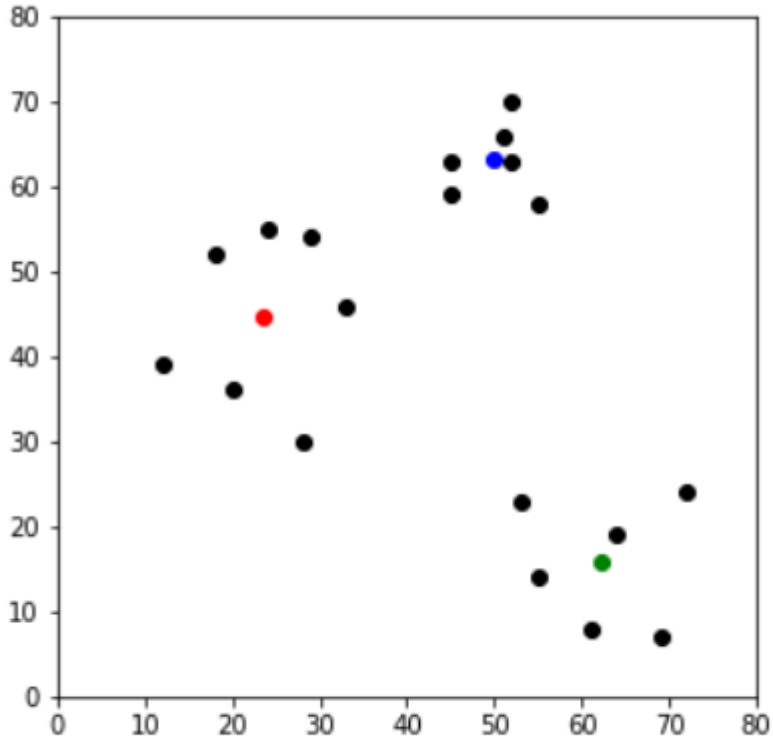
Practical 01

Aim: Program for K-Means Clustering.

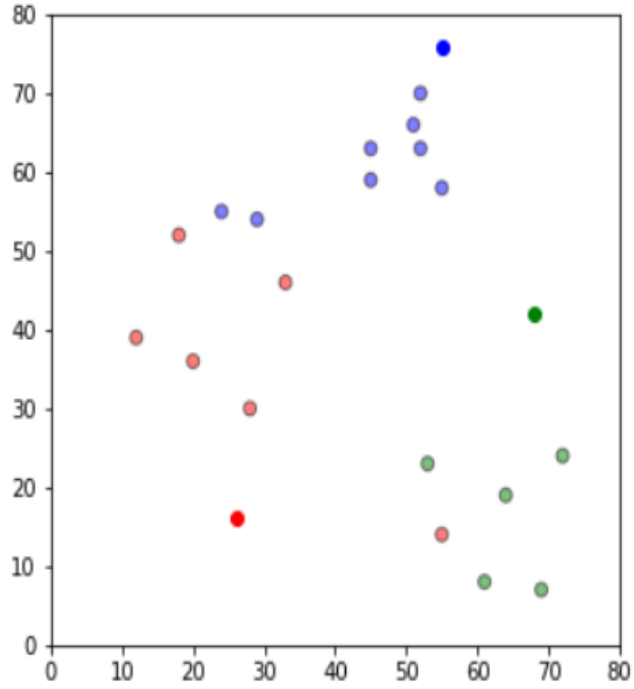
```
## Initialisation

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

df = pd.DataFrame({
    'x': [12, 20, 28, 18, 29, 33, 24, 45, 45, 52, 51, 52, 55, 53, 55, 61, 64, 69, 72],
    'y': [39, 36, 30, 52, 54, 46, 55, 59, 63, 70, 66, 63, 58, 23, 14, 8, 19, 7, 24]
})

np.random.seed(200)
k = 3
# centroids[i] = [x, y]
centroids = {
    i+1: [np.random.randint(0, 80), np.random.randint(0, 80)]
    for i in range(k)
}

fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], color='k')
colmap = {1: 'r', 2: 'g', 3: 'b'}
for i in centroids.keys():
    plt.scatter(*centroids[i], color=colmap[i])
plt.xlim(0, 80)
plt.ylim(0, 80)
plt.show()
print("__By Mazhar Solkar")
```



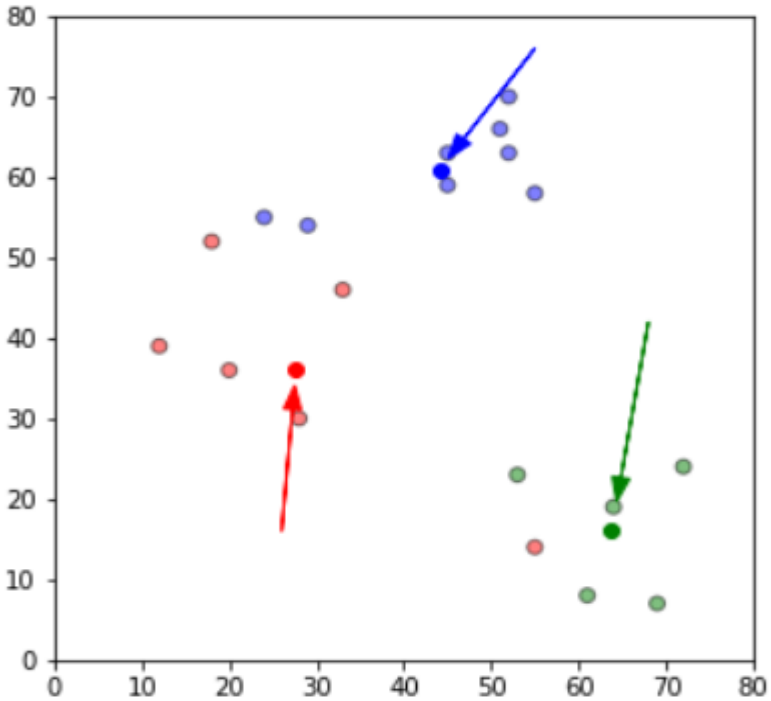

```
__By Mazhar Solkar
```

```python
## Assignment Stage

def assignment(df, centroids):
    for i in centroids.keys():
        # sqrt((x1 - x2)^2 - (y1 - y2)^2)
        df['distance_from_{}'.format(i)] = (
            np.sqrt(
                (df['x'] - centroids[i][0]) ** 2
                + (df['y'] - centroids[i][1]) ** 2
            )
        )
    centroid_distance_cols = ['distance_from_{}'.format(i) for i in centroids.keys()]
    df['closest'] = df.loc[:, centroid_distance_cols].idxmin(axis=1)
    df['closest'] = df['closest'].map(lambda x: int(x.lstrip('distance_from_')))
    df['color'] = df['closest'].map(lambda x: colmap[x])
    return df

df = assignment(df, centroids)
print(df.head())

fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], color=df['color'], alpha=0.5, edgecolor='k')
for i in centroids.keys():
    plt.scatter(*centroids[i], color=colmap[i])
plt.xlim(0, 80)
plt.ylim(0, 80)
plt.show()
```

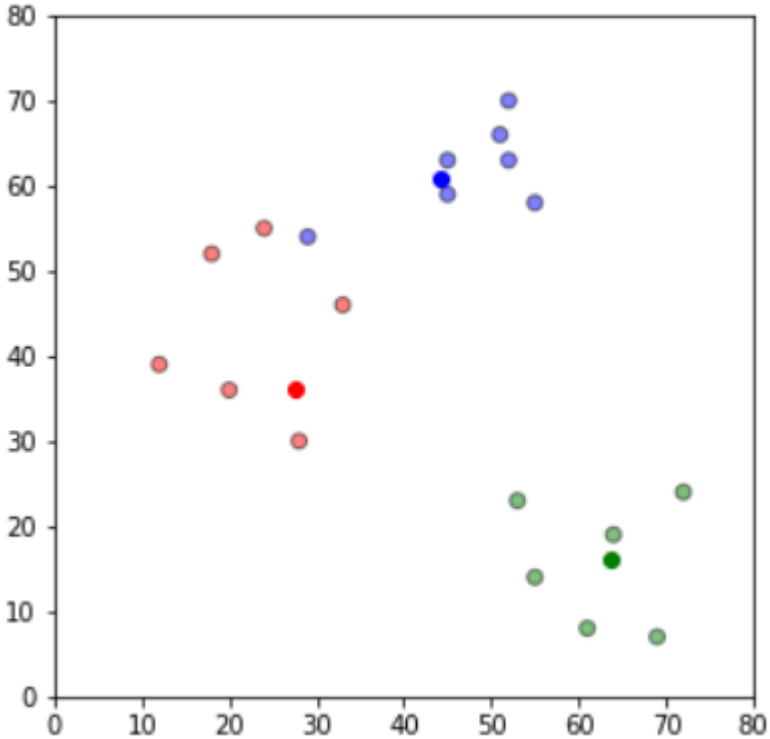
	x	y	distance_from_1	distance_from_2	distance_from_3	closest	color
0	12	39	26.925824	56.080300	56.727418	1	r
1	20	36	20.880613	48.373546	53.150729	1	r
2	28	30	14.142136	41.761226	53.338541	1	r
3	18	52	36.878178	50.990195	44.102154	1	r
4	29	54	38.118237	40.804412	34.058773	3	b



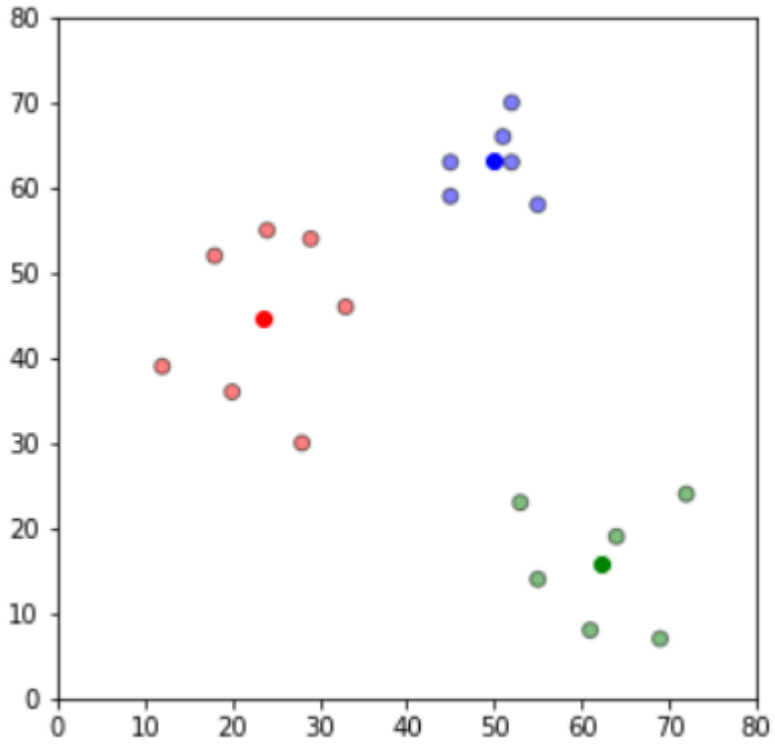
```
## Update Stage

import copy

old_centroids = copy.deepcopy(centroids)

def update(k):
    for i in centroids.keys():
        centroids[i][0] = np.mean(df[df['closest'] == i]['x'])
        centroids[i][1] = np.mean(df[df['closest'] == i]['y'])
    return k

centroids = update(centroids)

fig = plt.figure(figsize=(5, 5))
ax = plt.axes()
plt.scatter(df['x'], df['y'], color=df['color'], alpha=0.5, edgecolor='k')
for i in centroids.keys():
    plt.scatter(*centroids[i], color=colmap[i])
plt.xlim(0, 80)
plt.ylim(0, 80)
for i in old_centroids.keys():
    old_x = old_centroids[i][0]
    old_y = old_centroids[i][1]
    dx = (centroids[i][0] - old_centroids[i][0]) * 0.75
    dy = (centroids[i][1] - old_centroids[i][1]) * 0.75
    ax.arrow(old_x, old_y, dx, dy, head_width=2, head_length=3, fc=colmap[i], ec=colmap[i])
plt.show()
```



```
## Repeat Assigment Stage

df = assignment(df, centroids)

# Plot results
fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], color=df['color'], alpha=0.5, edgecolor='k')
for i in centroids.keys():
    plt.scatter(*centroids[i], color=colmap[i])
plt.xlim(0, 80)
plt.ylim(0, 80)
plt.show()
```



```
# Continue until all assigned categories don't change any more
while True:
    closest_centroids = df['closest'].copy(deep=True)
    centroids = update(centroids)
    df = assignment(df, centroids)
    if closest_centroids.equals(df['closest']):
        break

fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], color=df['color'], alpha=0.5, edgecolor='k')
for i in centroids.keys():
    plt.scatter(*centroids[i], color=colmap[i])
plt.xlim(0, 80)
plt.ylim(0, 80)
plt.show()
print("__By Mazhar Solkar")
```




```
__By Mazhar Solkar
```

Practical 04

Aim: Implement an application that stores big data in MongoDB and manipulate it using python.

STEP-1: Start mongod and mongo

```
C:\Users\Mazhar>mongo
MongoDB shell version v5.0.7
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("fb9497db-176b-4a6c-a93f-5581a578fad1") }
MongoDB server version: 5.0.7
```

STEP-2: Command to check databases in MongoDB is given below

```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use sample
switched to db sample
> show collections
> db.employee.insert({eno:1,ename:"amar",deptno:10})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({eno:2,ename:"arjun",deptno:10})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({eno:3,ename:"shruti",deptno:10})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({eno:4,ename:rohit,deptno:20})
uncaught exception: ReferenceError: rohit is not defined :
@(shell):1:27
> db.employee.insert({eno:4,ename:"rohit",deptno:20})
WriteResult({ "nInserted" : 1 })
> db.employee.insert({eno:5,ename:"sham",deptno:20})
WriteResult({ "nInserted" : 1 })
> db.employee.find()
{ "_id" : ObjectId("6258447d8ce536ffffa4f97d"), "eno" : 1, "ename" : "amar", "deptno" : 10 }
{ "_id" : ObjectId("625844aa8ce536ffffa4f97e"), "eno" : 2, "ename" : "arjun", "deptno" : 10 }
{ "_id" : ObjectId("625844d18ce536ffffa4f97f"), "eno" : 3, "ename" : "shruti", "deptno" : 10 }
{ "_id" : ObjectId("625845218ce536ffffa4f980"), "eno" : 4, "ename" : "rohit", "deptno" : 20 }
{ "_id" : ObjectId("6258454a8ce536ffffa4f981"), "eno" : 5, "ename" : "sham", "deptno" : 20 }
```

STEP-3: Type the following code in python.

mongo.py > ...

```python
from pymongo import MongoClient
client = MongoClient('localhost:27017')
db = client.sample
db = client.get_database('sample')
records = db.employee
print(records.count_documents({}))
print(list(records.find()))
print("\n__By Mazhar Solkar")
```

Output :-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Mazhar@DESKTOP-0PG7LTN MINGW64 /c/0_MSc_IT_Notes/Big Data Analytics/practicals (main)
$ python -u "c:\0_MSc_IT_Notes\Big Data Analytics\practicals\mongo.py"
5
[{'_id': ObjectId('6258447d8ce536ffffa4f97d'), 'eno': 1.0, 'ename': 'amar', 'deptno': 10.0}, {'_id': ObjectId
('625844aa8ce536ffffa4f97e'), 'eno': 2.0, 'ename': 'arjun', 'deptno': 10.0}, {'_id': ObjectId('625844d18ce536
ffffa4f97f'), 'eno': 3.0, 'ename': 'shruti', 'deptno': 10.0}, {'_id': ObjectId('625845218ce536ffffa4f980'), '
eno': 4.0, 'ename': 'rohit', 'deptno': 20.0}, {'_id': ObjectId('6258454a8ce536ffffa4f981'), 'eno': 5.0, 'enam
e': 'sham', 'deptno': 20.0}]

__By Mazhar Solkar
```