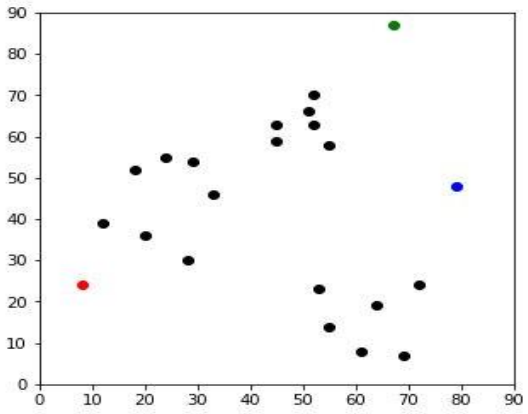
Name : Devashish Prasad
Roll no : 43320
Batch : Q11
Title : : K-means algorithm for Clustering
==========

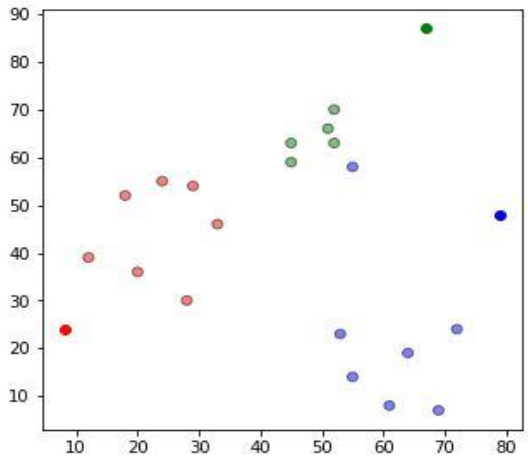
#Importing all libraries

```
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(100)
k=3
centroids = {}
# Randomly 3 initial centroids are selected
centroids={
   i+1:[np.random.randint(0,90),np.random.randint(0,90)]

  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(0,90)
   plt.ylim(0,90)
plt.show()
```



#Assignment of data points to the clusters closest to it.

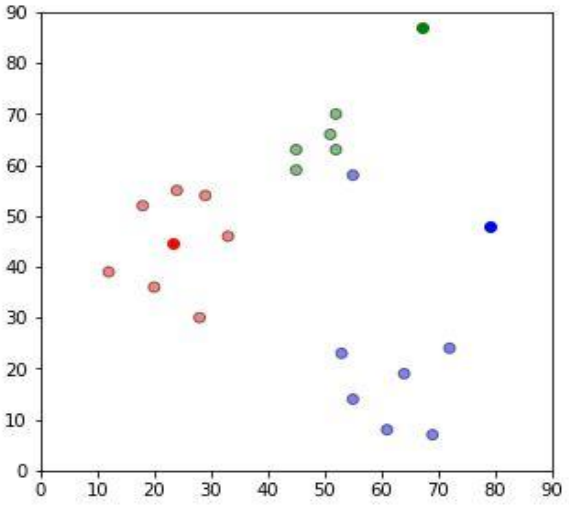
```
def assignment(df, centroids):
    for i in centroids.keys():
        df['distance_from_{}'.format(i)] = (
            np.sqrt(
                (df['x'] - centroids[i][0]) ** 2
                + (df['y'] - centroids[i][1]) ** 2
            )
        )
    centroid_distance_cols = ['distance_from_{}'.format(i) 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)
df
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])
```



#Updating the centroids(Mean of each cluster is calculated and this mean will act as a new centroid of the cluster)

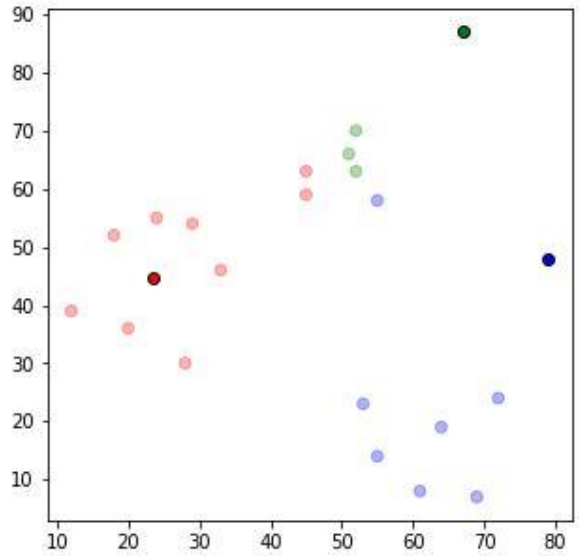
```
import copy
old_centroids=copy.deepcopy(centroids)
def update(k):
    for i in centroids.keys():
        centroids[i][0]=np.mean(df[df['closest']==i]['x'])
        centroids[i][1]=np.mean(df[df['closest']==i]['y'])
        return k
centroids=update(centroids)

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(0,90)
    plt.ylim(0,90)
```



 # Repeat Assignment step

```
df = assignment(df, centroids)
#Plot result
fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], color=df['color'], alpha=0.3)
for i in centroids.keys():
    plt.scatter(centroids[i][0],centroids[i][1], color=colmap[i], edgecolor='k')
plt.show()
```



#Continue until assigned categories dont 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
```

#Final Result :

```
fig = plt.figure(figsize=(5, 5))
plt.scatter(df['x'], df['y'], color=df['color'])
for i in centroids.keys():
    plt.scatter(centroids[i][0],centroids[i][1], color=colmap[i], edgecolor='k')
plt.show()
```