Assignment_Unsupervized_Learning

In [1]:

```python
#How many cluster possible?
#We can make the clusters.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```python
df=pd.DataFrame({'x':[12,20,28,18,29,33,24,45,52,51,53,55,54,65,61,67,69,72,76],
                 'y':[39,36,30,52,54,46,55,59,63,70,66,63,58,23,14,8,19,7,78]})
```

https://pastebin.com/ZB0yap9r (https://pastebin.com/ZB0yap9r)

In [3]:

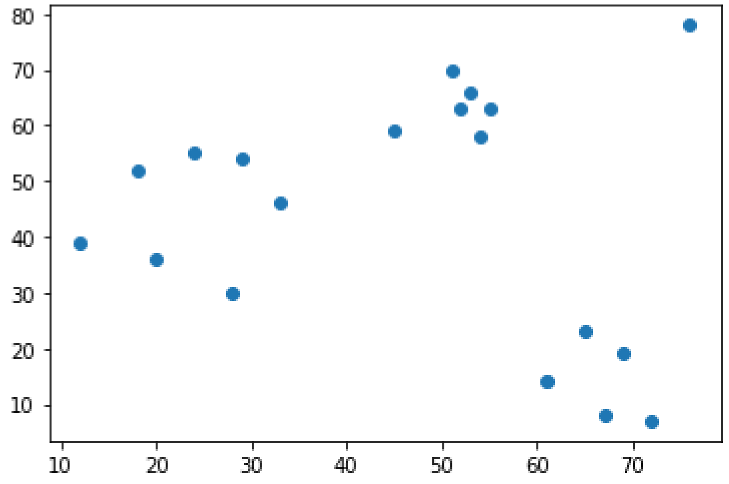
```python
df.head()
```

Out[3]:

	x	y
0	12	39
1	20	36
2	28	30
3	18	52
4	29	54

In [4]:

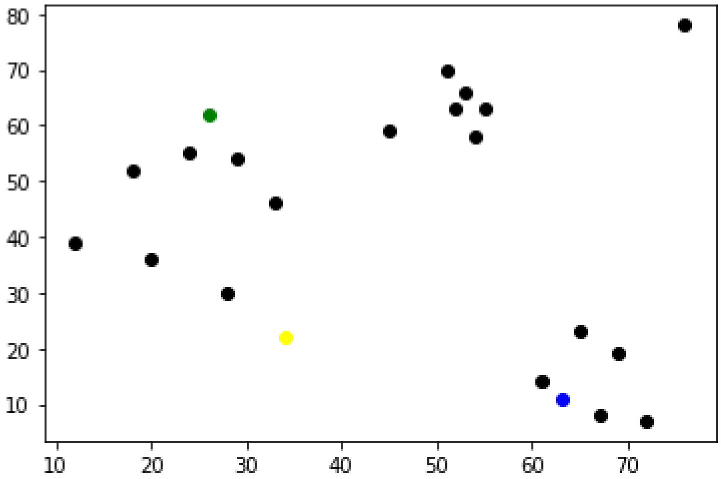
```python
plt.scatter(df.x,df.y)
plt.show()
```



In [5]:

```python
cent={1:[np.random.randint(10,70),np.random.randint(10,70)],
      2:[np.random.randint(10,70),np.random.randint(10,70)],
      3:[np.random.randint(10,70),np.random.randint(10,70)]}
```

In [6]:
```
colmap={1:'yellow',2:'g',3:'b'}
plt.scatter(df.x,df.y,color='black')
for i in cent.keys():
    plt.scatter(cent[i][0],cent[i][1],
                color=colmap[i])
plt.show()
```



In [7]:
```
cent[1][0]
cent[1][1]
```

Out[7]:
```
22
```

In [8]:
```
#Assignment Stage
def assignment(df,cent):
    for i in cent.keys():
        df[str(i)]=np.sqrt((df['x']-cent[i][0])**2 +
                           (df['y']-cent[i][1])**2)
    df['closest']=df.loc[:,'1':'3'].idxmin(axis=1)
    df['color']=df['closest'].map({'1':'yellow',
                                   '2':'g','3':'b'})
    return df
```

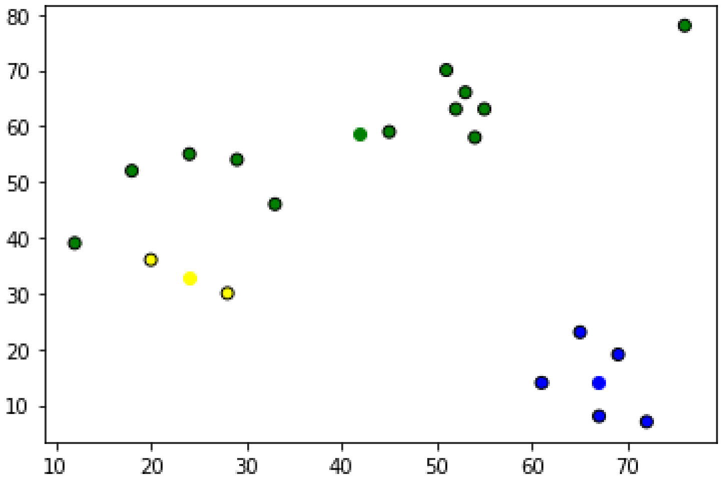
In [9]:
```
#Update Stage
def update(k):
    for i in cent.keys():
        #First Centeroid First Coordinate
        k[i][0] = df[df['closest']==str(i)].x.mean()
        #First Centeroid 2nd Cordinate
        k[i][1] = df[df['closest']==str(i)].y.mean()
    return k
```

In [10]:
```
df = assignment(df,cent)
```

In [11]:
```
cent=update(cent)
```

In [12]:
```
import copy
old_cent=copy.deepcopy(cent)
while cent != old_cent:
    cent=update(cent)
    df = assignment(df,cent)
    old_cent=copy.deepcopy(cent)
```

In [13]:
```
plt.scatter(df.x,df.y,color=df['color'],edgecolor='k')
colmap = {1 : 'yellow', 2: 'g', 3: 'b'}
for i in cent.keys():
    plt.scatter(*cent[i], color = colmap[i])
plt.show()
```



In [14]:
```
cent
```

Out[14]:
```
{1: [24.0, 33.0], 2: [41.833333333333336, 58.583333333333336], 3: [66.8, 14.2]}
```