K - Means Clustering Model

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import make_blobs
```

In [2]:
```
dataset = make_blobs(n_samples=200,
                     centers=4,
                     n_features=2,
                     cluster_std=1.6,
                     random_state=50)
```

In [3]:
```
points = dataset[0]
```

In [4]:
```
# KMeans
from sklearn.cluster import KMeans
```

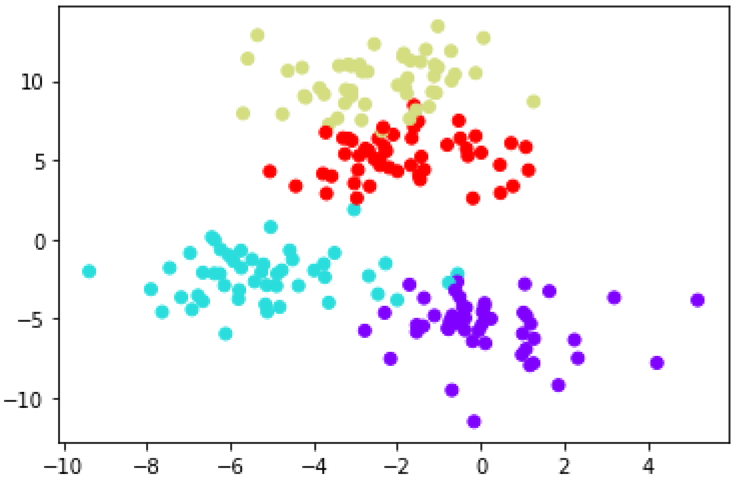
In [5]:
```
# Create a K means object
kmeans = KMeans(n_clusters=4)
```

In [6]:
```
# Fit this Kmeans object to this dataset
kmeans.fit(points)
```

Out[6]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
       n_clusters=4, n_init=10, n_jobs=None, precompute_distances='auto',
       random_state=None, tol=0.0001, verbose=0)
```

In [7]:
```
plt.scatter(dataset[0][:,0],dataset[0][:,1],c=dataset[1],cmap="rainbow")
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0x222085f7f48>
```



In [8]:
```
clusters = kmeans.cluster_centers_
```

In [9]:
```
print(clusters)
```

```
[[ 0.05161133 -5.35489826]
 [-1.92101646  5.21673484]
 [-5.56465793 -2.34988939]
 [-2.40167949 10.17352695]]
```

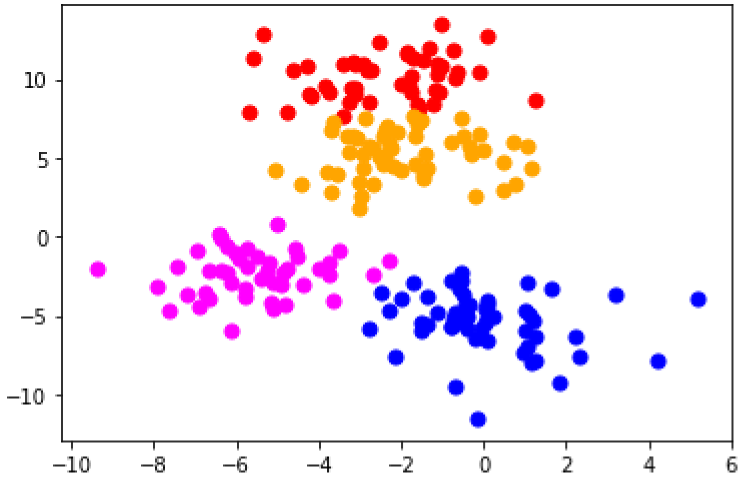
In [10]:
```
y_km = kmeans.fit_predict(points)
```

In [11]:
```
y_km
```

Out[11]:
```
array([0, 1, 2, 0, 3, 1, 0, 1, 1, 3, 3, 0, 3, 2, 3, 2, 0, 0, 3, 3, 3, 2,
       0, 0, 0, 2, 0, 1, 2, 0, 0, 3, 3, 1, 1, 3, 3, 3, 2, 2, 0, 3, 2, 2,
       1, 1, 0, 2, 2, 0, 1, 3, 2, 0, 2, 2, 1, 3, 3, 2, 1, 1, 0, 1, 1, 0,
       1, 2, 0, 0, 1, 3, 0, 1, 2, 2, 3, 2, 2, 1, 3, 2, 3, 3, 3, 3, 0, 1,
       0, 3, 1, 0, 1, 2, 2, 3, 1, 2, 1, 2, 3, 1, 2, 0, 2, 1, 2, 1, 3, 0,
       1, 3, 0, 0, 3, 2, 0, 0, 2, 1, 0, 2, 0, 0, 1, 0, 0, 2, 2, 1, 3, 1,
       1, 2, 0, 2, 1, 0, 0, 3, 0, 2, 2, 1, 3, 0, 1, 2, 1, 3, 0, 3, 2, 1,
       1, 2, 0, 1, 1, 1, 2, 2, 1, 1, 3, 3, 3, 2, 1, 1, 1, 1, 3, 2, 1, 2,
       3, 1, 0, 2, 1, 2, 2, 0, 3, 1, 3, 0, 2, 1, 2, 2, 3, 2, 1, 0, 3, 2,
       3, 0])
```

In [12]:
```
plt.scatter(points[y_km == 0,0],points[y_km == 0,1],s=50,color="red")
plt.scatter(points[y_km == 1,0],points[y_km == 1,1],s=50,color="blue")
plt.scatter(points[y_km == 2,0],points[y_km == 2,1],s=50,color="orange")
plt.scatter(points[y_km == 3,0],points[y_km == 3,1],s=50,color="magenta")
```

Out[12]:
```
<matplotlib.collections.PathCollection at 0x2220bb568c8>
```

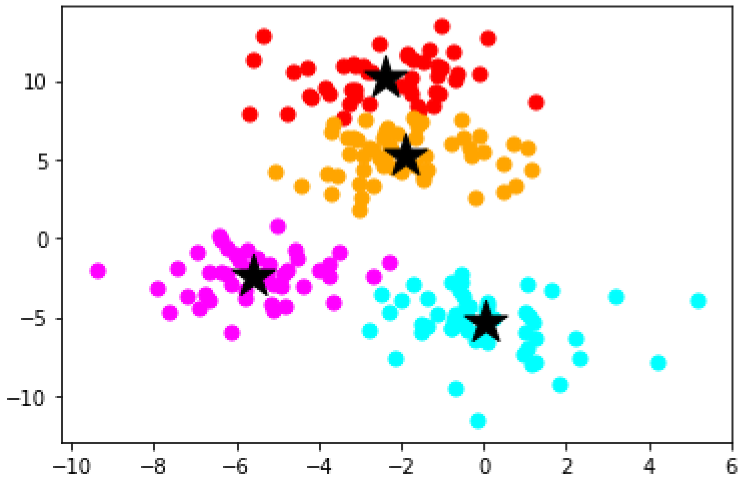


In [13]:

```
plt.scatter(points[y_km == 0,0],points[y_km == 0,1],s=50,color="red")
plt.scatter(points[y_km == 1,0],points[y_km == 1,1],s=50,color="cyan")
plt.scatter(points[y_km == 2,0],points[y_km == 2,1],s=50,color="orange")
plt.scatter(points[y_km == 3,0],points[y_km == 3,1],s=50,color="magenta")
plt.scatter(clusters[0][0],clusters[0][1],marker="*",s=500,color="black")
plt.scatter(clusters[1][0],clusters[1][1],marker="*",s=500,color="black")
plt.scatter(clusters[2][0],clusters[2][1],marker="*",s=500,color="black")
plt.scatter(clusters[3][0],clusters[3][1],marker="*",s=500,color="black")
```

Out[13]:

```
<matplotlib.collections.PathCollection at 0x2220bbedd48>
```

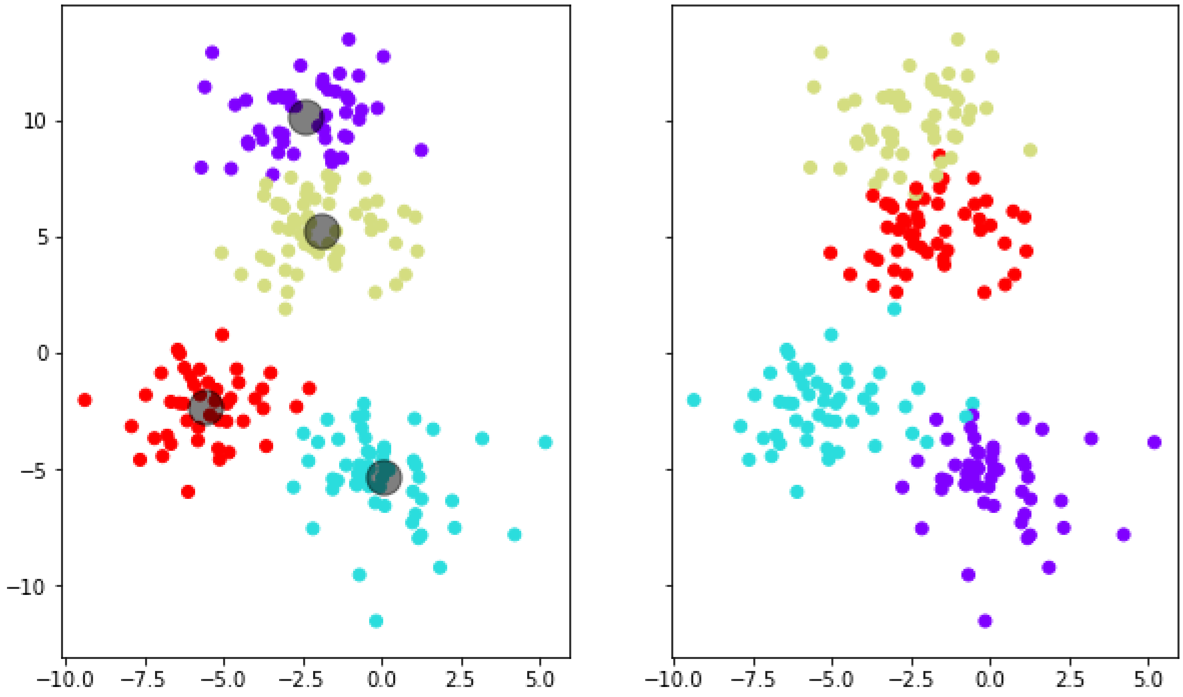


```
In [14]: # Making the Clusters
         f, (ax1, ax2) = plt.subplots(nrows=1,
          ncols=2,
          sharey=True,
          figsize=(10,6))

         ax1.scatter(dataset[0][:,0],
          dataset[0][:,1],
          c=y_km,
          cmap='rainbow')

         ax2.scatter(dataset[0][:,0],
          dataset[0][:,1],
          c=dataset[1],
          cmap='rainbow')

         ax1.scatter(x=clusters[:, 0],
          y=clusters[:, 1],
          c='black',
          s=300,
          alpha=0.5);
```


10
5
0
−5
−10
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0