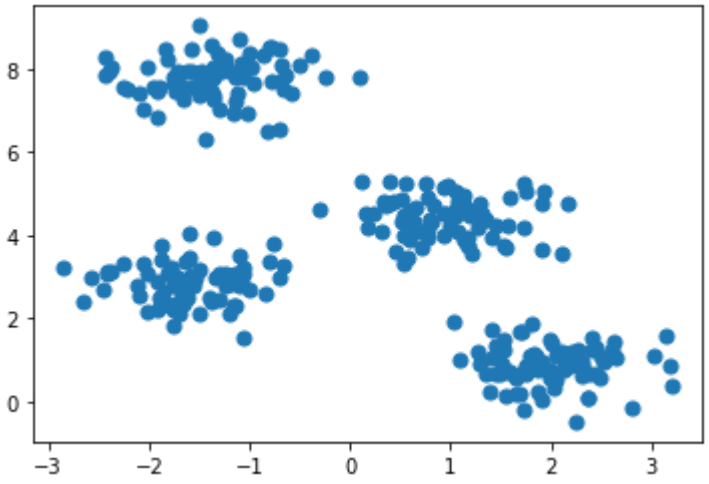
Clustering

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
from sklearn.datasets import make_blobs #to create data
X,y_true=make_blobs(n_samples=300,centers=4,
                    cluster_std=0.5,random_state=0)
plt.scatter(X[:,0],X[:,1],s=50)
```

Out[2]: <matplotlib.collections.PathCollection at 0x27913e0c940>



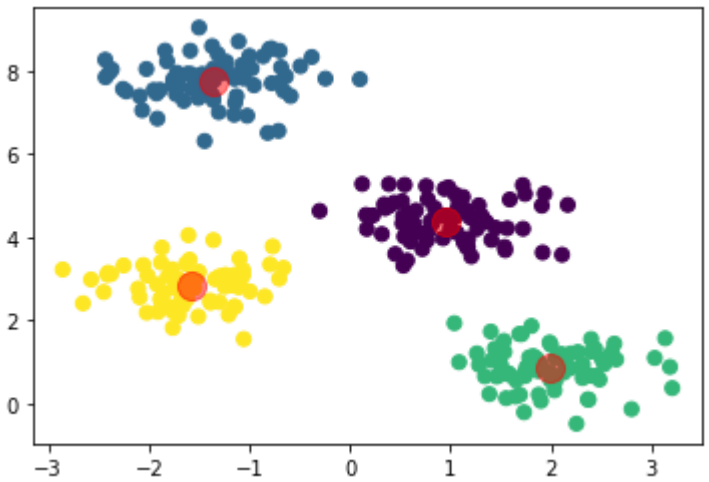
In [3]:
```
kmeans=KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans=kmeans.predict(X)
```

In [4]:
```
y_kmeans
```

Out[4]:
```
array([2, 1, 0, 1, 2, 2, 3, 0, 1, 1, 3, 1, 0, 1, 2, 0, 0, 2, 3, 3, 2, 2,
       0, 3, 3, 0, 2, 0, 3, 0, 1, 1, 0, 1, 1, 1, 1, 1, 3, 2, 0, 3, 0, 0,
       3, 3, 1, 3, 1, 2, 3, 2, 1, 2, 2, 3, 1, 3, 1, 2, 1, 0, 1, 3, 3, 3,
       1, 2, 1, 3, 0, 3, 1, 3, 3, 1, 3, 0, 2, 1, 2, 0, 2, 2, 1, 0, 2, 0,
       1, 1, 0, 2, 1, 3, 3, 0, 2, 2, 0, 3, 1, 2, 1, 2, 0, 2, 2, 0, 1, 0,
       3, 3, 2, 1, 2, 0, 1, 2, 2, 0, 3, 2, 3, 2, 2, 2, 2, 3, 2, 3, 1, 3,
       3, 2, 1, 3, 3, 1, 0, 1, 1, 3, 0, 3, 0, 3, 1, 0, 1, 1, 1, 0, 1, 0,
       2, 3, 1, 3, 2, 0, 1, 0, 0, 2, 0, 3, 3, 0, 2, 0, 0, 1, 2, 0, 3, 1,
       2, 2, 0, 3, 2, 0, 3, 3, 0, 0, 0, 0, 2, 1, 0, 3, 0, 0, 3, 3, 3, 0,
       3, 1, 0, 3, 2, 3, 0, 1, 3, 1, 0, 1, 0, 3, 0, 0, 1, 3, 3, 2, 2, 0,
       1, 2, 2, 3, 2, 3, 0, 1, 1, 0, 0, 1, 0, 2, 3, 0, 2, 3, 1, 3, 2, 0,
       2, 1, 1, 1, 1, 3, 3, 1, 0, 3, 2, 0, 3, 3, 3, 2, 2, 1, 0, 0, 3, 2,
       1, 3, 0, 1, 0, 2, 2, 3, 3, 0, 2, 2, 2, 0, 1, 1, 2, 2, 0, 2, 2, 2,
       1, 3, 1, 0, 2, 2, 1, 1, 1, 2, 2, 0, 1, 3])
```

In [5]:
```
plt.scatter(X[:,0],X[:,1],c=y_kmeans,s=50,cmap='viridis')
centers=kmeans.cluster_centers_
plt.scatter(centers[:,0],centers[:,1],c='red',s=200,alpha=0.5)
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x279149ef340>
```



In [6]:
```
from sklearn.datasets import load_digits
digits=load_digits()
digits.data.shape #64 pixel and 1797 images
```

Out[6]:
```
(1797, 64)
```

In [7]:
```
digits.data
```

Out[7]:
```
array([[ 0.,  0.,  5., ...,  0.,  0.,  0.],
       [ 0.,  0.,  0., ..., 10.,  0.,  0.],
       [ 0.,  0.,  0., ..., 16.,  9.,  0.],
       ...,
       [ 0.,  0.,  1., ...,  6.,  0.,  0.],
       [ 0.,  0.,  2., ..., 12.,  0.,  0.],
       [ 0.,  0., 10., ..., 12.,  1.,  0.]])
```

In [8]:
```
kmeans=KMeans(n_clusters=10,random_state=42)
cluster=kmeans.fit_predict(digits.data)
kmeans.cluster_centers_.shape
```

Out[8]:
```
(10, 64)
```

In [9]:
```
cluster
```

Out[9]:
```
array([0, 5, 5, ..., 5, 1, 1])
```

In [10]:
```
kmeans.cluster_centers_
```

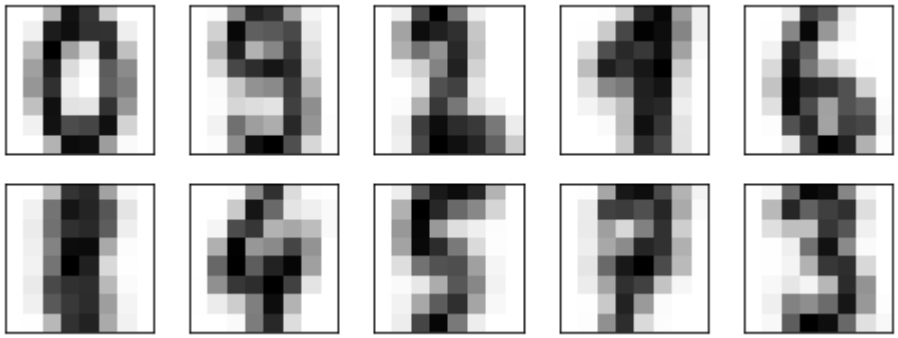
Out[10]:
```
array([[ 0.00000000e+00,  2.23463687e-02,  4.22905028e+00,
         1.31396648e+01,  1.12681564e+01,  2.93854749e+00,
         3.35195531e-02, -5.55111512e-17,  2.60208521e-18,
         8.82681564e-01,  1.26201117e+01,  1.33687151e+01,
         1.14078212e+01,  1.13687151e+01,  9.60893855e-01,
         0.00000000e+00,  1.30104261e-18,  3.72625698e+00,
         1.42122905e+01,  5.25139665e+00,  2.10614525e+00,
         1.21173184e+01,  3.53072626e+00, -1.38777878e-17,
        -4.33680869e-19,  5.29608939e+00,  1.26424581e+01,
         2.03351955e+00,  2.29050279e-01,  9.07821229e+00,
         6.47486034e+00, -8.67361738e-19,  0.00000000e+00,
         5.88268156e+00,  1.14916201e+01,  8.65921788e-01,
         3.35195531e-02,  8.81005587e+00,  7.15083799e+00,
         0.00000000e+00, -3.46944695e-18,  3.51396648e+00,
         1.32849162e+01,  1.65921788e+00,  1.49162011e+00,
         1.13519553e+01,  5.84357542e+00,  3.46944695e-18,
         8.67361738e-19,  8.04469274e-01,  1.31117318e+01,
         9.96089385e+00,  1.03519553e+01,  1.32960894e+01,
         2.47486034e+00,  2.23463687e-02, -2.16840434e-19,
         5.58659218e-03,  4.19553073e+00,  1.35865922e+01,
         1.33407821e+01,  5.48044693e+00,  3.18435754e-01,
         1.67597765e-02],
       [ 0.00000000e+00,  2.00819672e-01,  6.55737705e+00,
         1.25614754e+01,  1.17950820e+01,  5.55327869e+00,
         5.98360656e-01,  8.19672131e-03,  4.09836066e-03,
         2.63524590e+00,  1.39959016e+01,  9.10655738e+00,
         9.40163934e+00,  1.03524590e+01,  1.21311475e+00,
         4.09836066e-03,  1.30104261e-18,  4.33196721e+00,
         1.27991803e+01,  4.36885246e+00,  6.86475410e+00,
         1.11721311e+01,  1.88524590e+00,  0.00000000e+00,
         2.16840434e-19,  2.30737705e+00,  1.04672131e+01,
         1.19262295e+01,  1.32336066e+01,  1.20860656e+01,
         2.49180328e+00,  4.33680869e-19,  0.00000000e+00,
         3.07377049e-01,  3.22131148e+00,  6.22950820e+00,
         6.68032787e+00,  1.12090164e+01,  4.30737705e+00,
         0.00000000e+00,  1.73472348e-18,  2.21311475e-01,
         2.37295082e+00,  1.93852459e+00,  1.64344262e+00,
         1.08606557e+01,  6.44672131e+00,  1.63934426e-02,
         3.46944695e-18,  7.54098361e-01,  8.11475410e+00,
         5.51639344e+00,  4.65983607e+00,  1.22172131e+01,
         6.02049180e+00,  1.14754098e-01,  1.08420217e-19,
         1.72131148e-01,  6.47950820e+00,  1.35245902e+01,
         1.45163934e+01,  9.98360656e+00,  2.32377049e+00,
         1.14754098e-01],
       [ 0.00000000e+00,  9.42857143e-01,  1.01885714e+01,
         1.44400000e+01,  7.77142857e+00,  9.82857143e-01,
        -6.66133815e-16,  0.00000000e+00,  2.28571429e-02,
         5.24000000e+00,  1.37200000e+01,  1.26228571e+01,
         1.16914286e+01,  3.23428571e+00,  1.71428571e-02,
        -1.38777878e-17,  1.14285714e-02,  4.56000000e+00,
         8.11428571e+00,  6.13714286e+00,  1.21600000e+01,
         3.56000000e+00,  1.71428571e-02, -6.93889390e-18,
        -2.16840434e-19,  9.65714286e-01,  2.81714286e+00,
         7.00571429e+00,  1.25371429e+01,  2.56000000e+00,
         4.00000000e-02, -4.33680869e-19,  0.00000000e+00,
         4.57142857e-02,  1.57142857e+00,  9.89714286e+00,
         1.06971429e+01,  1.45142857e+00,  0.00000000e+00,
```

```
  0.00000000e+00, -1.73472348e-18,  2.51428571e-01,
  4.45714286e+00,  1.12457143e+01,  7.74285714e+00,
  2.37142857e+00,  8.45714286e-01,  1.14285714e-02,
  1.73472348e-18,  1.19428571e+00,  1.09942857e+01,
  1.37314286e+01,  1.19257143e+01,  1.11600000e+01,
  7.66857143e+00,  1.10285714e+00, -1.08420217e-19,
  9.31428571e-01,  1.03885714e+01,  1.44685714e+01,
  1.35028571e+01,  1.23542857e+01,  8.96571429e+00,
  2.95428571e+00],
[ 0.00000000e+00,  1.66533454e-16,  1.53061224e-01,
  2.57142857e+00,  1.11530612e+01,  1.31836735e+01,
  5.31632653e+00,  7.44897959e-01,  1.73472348e-18,
  9.18367347e-02,  2.80612245e+00,  9.80612245e+00,
  1.33775510e+01,  1.28367347e+01,  6.46938776e+00,
  7.55102041e-01,  8.67361738e-19,  1.64285714e+00,
  9.33673469e+00,  1.13367347e+01,  1.05714286e+01,
  1.26428571e+01,  4.97959184e+00,  2.55102041e-01,
 -2.16840434e-19,  3.53061224e+00,  1.19693878e+01,
  1.08367347e+01,  1.23469388e+01,  1.35714286e+01,
  2.82653061e+00, -4.33680869e-19,  0.00000000e+00,
  1.66326531e+00,  6.34693878e+00,  7.07142857e+00,
  1.18367347e+01,  1.24591837e+01,  1.51020408e+00,
  0.00000000e+00, -1.73472348e-18,  6.22448980e-01,
  1.71428571e+00,  3.55102041e+00,  1.17040816e+01,
  1.14285714e+01,  9.38775510e-01,  0.00000000e+00,
  8.67361738e-19,  3.06122449e-02,  2.95918367e-01,
  3.64285714e+00,  1.26428571e+01,  1.06938776e+01,
  1.50000000e+00, -5.55111512e-17, -1.08420217e-19,
  0.00000000e+00,  0.00000000e+00,  2.95918367e+00,
  1.12551020e+01,  9.70408163e+00,  1.55102041e+00,
  0.00000000e+00],
[ 0.00000000e+00,  1.11022302e-16,  1.15934066e+00,
  1.12252747e+01,  9.53296703e+00,  1.41758242e+00,
  5.49450549e-03, -5.55111512e-17,  2.60208521e-18,
  6.04395604e-02,  7.18131868e+00,  1.45604396e+01,
  6.19230769e+00,  8.29670330e-01,  2.74725275e-02,
 -2.77555756e-17,  1.30104261e-18,  7.69230769e-01,
  1.24560440e+01,  9.47252747e+00,  9.34065934e-01,
  1.09890110e-01,  0.00000000e+00, -2.08166817e-17,
 -6.50521303e-19,  2.29670330e+00,  1.36208791e+01,
  8.09340659e+00,  3.87362637e+00,  1.92857143e+00,
  1.04395604e-01, -1.30104261e-18,  0.00000000e+00,
  3.52747253e+00,  1.46758242e+01,  1.29175824e+01,
  1.22527473e+01,  1.02857143e+01,  2.71978022e+00,
  0.00000000e+00, -5.20417043e-18,  1.86813187e+00,
  1.45164835e+01,  1.06538462e+01,  5.57692308e+00,
  1.01923077e+01,  9.13186813e+00,  2.30769231e-01,
  8.67361738e-19,  1.75824176e-01,  1.02857143e+01,
  1.26263736e+01,  5.41758242e+00,  1.13241758e+01,
  1.08956044e+01,  6.26373626e-01, -3.25260652e-19,
 -5.55111512e-17,  1.44505495e+00,  1.07362637e+01,
  1.50989011e+01,  1.31318681e+01,  4.62087912e+00,
  1.70329670e-01],
[ 0.00000000e+00,  1.12107623e-01,  3.91479821e+00,
  1.17668161e+01,  1.24484305e+01,  5.38565022e+00,
  4.34977578e-01,  2.77555756e-17,  8.96860987e-03,
  8.29596413e-01,  8.08520179e+00,  1.35470852e+01,
```

```
  1.26816143e+01,  9.82062780e+00,  1.56502242e+00,
 -4.16333634e-17,  1.30104261e-18,  1.18834081e+00,
  8.19730942e+00,  1.19686099e+01,  1.23408072e+01,
  9.37668161e+00,  1.02242152e+00,  0.00000000e+00,
  0.00000000e+00,  9.77578475e-01,  7.20627803e+00,
  1.40941704e+01,  1.41748879e+01,  5.00448430e+00,
  2.06278027e-01,  0.00000000e+00,  0.00000000e+00,
  7.89237668e-01,  8.08071749e+00,  1.48161435e+01,
  1.29013453e+01,  2.25112108e+00,  6.27802691e-02,
  0.00000000e+00,  0.00000000e+00,  1.23318386e+00,
  1.05291480e+01,  1.19865471e+01,  1.21748879e+01,
  4.05381166e+00,  2.69058296e-01,  6.93889390e-18,
  1.34529148e-02,  8.78923767e-01,  9.58744395e+00,
  1.14932735e+01,  1.22152466e+01,  5.66816143e+00,
  6.81614350e-01,  4.48430493e-03,  4.48430493e-03,
  1.12107623e-01,  4.11659193e+00,  1.19013453e+01,
  1.27533632e+01,  4.97309417e+00,  8.56502242e-01,
  8.96860987e-03],
[ 0.00000000e+00,  1.66533454e-16,  2.60355030e-01,
  6.85798817e+00,  1.20177515e+01,  2.08875740e+00,
  1.47928994e-01,  5.32544379e-02,  1.73472348e-18,
  1.18343195e-02,  3.10059172e+00,  1.35621302e+01,
  8.72781065e+00,  1.56804734e+00,  9.52662722e-01,
  3.13609467e-01,  8.67361738e-19,  5.97633136e-01,
  1.03905325e+01,  1.16804734e+01,  4.46153846e+00,
  5.13609467e+00,  3.84615385e+00,  3.49112426e-01,
  5.91715976e-03,  4.59171598e+00,  1.45976331e+01,
  6.07100592e+00,  6.79289941e+00,  1.07810651e+01,
  6.25443787e+00,  1.77514793e-02,  0.00000000e+00,
  8.80473373e+00,  1.48284024e+01,  9.40236686e+00,
  1.27692308e+01,  1.44674556e+01,  5.52662722e+00,
  0.00000000e+00,  9.46745562e-02,  6.47928994e+00,
  1.15976331e+01,  1.22189349e+01,  1.47751479e+01,
  1.09822485e+01,  1.62130178e+00,  3.46944695e-18,
  5.91715976e-02,  1.11242604e+00,  2.91124260e+00,
  7.44970414e+00,  1.40295858e+01,  4.43195266e+00,
  1.77514793e-02, -1.38777878e-16, -1.08420217e-19,
  2.36686391e-02,  3.13609467e-01,  7.64497041e+00,
  1.24023669e+01,  1.98816568e+00, -8.88178420e-16,
  0.00000000e+00],
[ 0.00000000e+00,  1.10810811e+00,  1.00135135e+01,
  1.33986486e+01,  1.41891892e+01,  1.26081081e+01,
  4.40540541e+00,  4.05405405e-02,  6.75675676e-03,
  4.56081081e+00,  1.49391892e+01,  1.26418919e+01,
  8.72297297e+00,  7.00000000e+00,  2.47972973e+00,
  3.37837838e-02,  1.35135135e-02,  6.06756757e+00,
  1.45270270e+01,  5.99324324e+00,  1.97972973e+00,
  9.39189189e-01,  1.75675676e-01, -2.08166817e-17,
  6.75675676e-03,  5.31756757e+00,  1.43310811e+01,
  1.23648649e+01,  7.81081081e+00,  2.20945946e+00,
  1.48648649e-01, -4.33680869e-19,  0.00000000e+00,
  1.95945946e+00,  8.18918919e+00,  1.00608108e+01,
  1.03310811e+01,  5.54729730e+00,  6.41891892e-01,
  0.00000000e+00, -1.73472348e-18,  3.04054054e-01,
  1.40540541e+00,  4.88513514e+00,  9.85135135e+00,
  7.06081081e+00,  7.83783784e-01,  3.46944695e-18,
 -8.67361738e-19,  8.10810811e-01,  5.09459459e+00,
```

```
  9.54054054e+00,  1.21418919e+01,  5.27027027e+00,
  4.45945946e-01, -1.11022302e-16, -1.08420217e-19,
  1.06081081e+00,  1.08918919e+01,  1.45540541e+01,
  7.80405405e+00,  1.06756757e+00,  2.02702703e-02,
  0.00000000e+00],
[ 0.00000000e+00,  1.66666667e-01,  5.15151515e+00,
  1.33080808e+01,  1.39444444e+01,  1.05202020e+01,
  4.44444444e+00,  7.22222222e-01,  2.60208521e-18,
  1.18181818e+00,  1.09646465e+01,  1.13181818e+01,
  1.02121212e+01,  1.25555556e+01,  4.93434343e+00,
  2.97979798e-01,  1.30104261e-18,  1.24747475e+00,
  5.60606061e+00,  2.29292929e+00,  7.29292929e+00,
  1.18737374e+01,  2.84848485e+00,  3.03030303e-02,
 -4.33680869e-19,  1.02020202e+00,  5.07070707e+00,
  6.75757576e+00,  1.26060606e+01,  1.19090909e+01,
  4.59595960e+00,  5.05050505e-03,  0.00000000e+00,
  1.50505051e+00,  8.71717172e+00,  1.33737374e+01,
  1.48030303e+01,  1.04595960e+01,  4.01515152e+00,
  0.00000000e+00, -3.46944695e-18,  1.11111111e+00,
  5.22727273e+00,  1.20151515e+01,  1.09343434e+01,
  3.43434343e+00,  5.55555556e-01,  3.46944695e-18,
  2.60208521e-18,  1.06060606e-01,  3.15656566e+00,
  1.27525253e+01,  5.85353535e+00,  2.82828283e-01,
 -2.22044605e-15, -1.11022302e-16, -2.16840434e-19,
  1.31313131e-01,  6.46464646e+00,  1.21969697e+01,
  2.18686869e+00,  1.86868687e-01,  1.01010101e-02,
  5.55111512e-17],
[ 0.00000000e+00,  5.85635359e-01,  8.66298343e+00,
  1.45248619e+01,  1.40331492e+01,  7.09392265e+00,
  6.62983425e-01, -2.77555756e-17,  1.10497238e-02,
  4.13812155e+00,  1.26574586e+01,  9.20994475e+00,
  1.12486188e+01,  1.20386740e+01,  1.95580110e+00,
  1.10497238e-02,  5.52486188e-03,  1.88397790e+00,
  3.77348066e+00,  3.68508287e+00,  1.17458564e+01,
  9.95580110e+00,  9.00552486e-01, -6.93889390e-18,
 -4.33680869e-19,  6.07734807e-02,  9.72375691e-01,
  8.17679558e+00,  1.38066298e+01,  6.87292818e+00,
  3.31491713e-01, -8.67361738e-19,  0.00000000e+00,
  6.07734807e-02,  6.68508287e-01,  4.53591160e+00,
  1.17071823e+01,  1.22375691e+01,  2.29834254e+00,
  0.00000000e+00, -3.46944695e-18,  4.58563536e-01,
  1.48066298e+00,  6.85082873e-01,  4.23756906e+00,
  1.23977901e+01,  6.24309392e+00,  5.52486188e-03,
  1.73472348e-18,  9.33701657e-01,  7.34254144e+00,
  6.60773481e+00,  8.66298343e+00,  1.36298343e+01,
  6.01657459e+00,  1.71270718e-01, -2.16840434e-19,
  4.64088398e-01,  9.45303867e+00,  1.49226519e+01,
  1.40828729e+01,  8.81215470e+00,  1.84530387e+00,
  4.08839779e-01]])
```

In [11]:
```
fig,ax=plt.subplots(2,5,figsize=(8,3))
centers=kmeans.cluster_centers_.reshape(10,8,8)
for axi,center in zip(ax.flat,centers):
    axi.set(xticks=[],yticks=[])
    axi.imshow(center,interpolation='nearest',cmap=plt.cm.binary)
```



In [12]:
```
import seaborn as sns;sns.set()#for plot styling
from sklearn.metrics import confusion_matrix
mat=confusion_matrix(digits.target,cluster)
mat
```

Out[12]:
```
array([[177,   0,   0,   0,   0,   0,   1,   0,   0,   0],
       [  0,   0,  24,  54,   2, 100,   0,   1,   0,   1],
       [  1,   2, 148,   3,   0,   8,   0,   0,   2,  13],
       [  0,  12,   0,   0,   0,   7,   0,   2,   6, 156],
       [  0,   0,   0,   2,   0,   2, 166,   0,  11,   0],
       [  0,  41,   0,   0,   1,   0,   2, 136,   0,   2],
       [  1,   0,   0,   0, 177,   3,   0,   0,   0,   0],
       [  0,   0,   0,  10,   0,   2,   0,   0, 167,   0],
       [  0,  50,   3,   9,   2, 100,   0,   4,   4,   2],
       [  0, 139,   0,  20,   0,   1,   0,   5,   8,   7]], dtype=int64)
```

Using Clustering on iris dataset

In [13]:
```
df=pd.read_csv('D:\\Downloads\\iris.csv')
df
```

Out[13]:

	Sepal length	Sepal width	Petal length	Petal width	Class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 5 columns

In [14]:
```
X=df[['Sepal length','Sepal width','Petal length','Petal width']].values
```

In [15]:
```
kmeans=KMeans(n_clusters=3)
kmeans.fit(X)
y_kmeans=kmeans.predict(X)
```

In [16]:
```
print(y_kmeans)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 2 2 2 1 2 2 2 2
 2 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2
 2 1]
```

In [17]:
```
kmeans.cluster_centers_
```

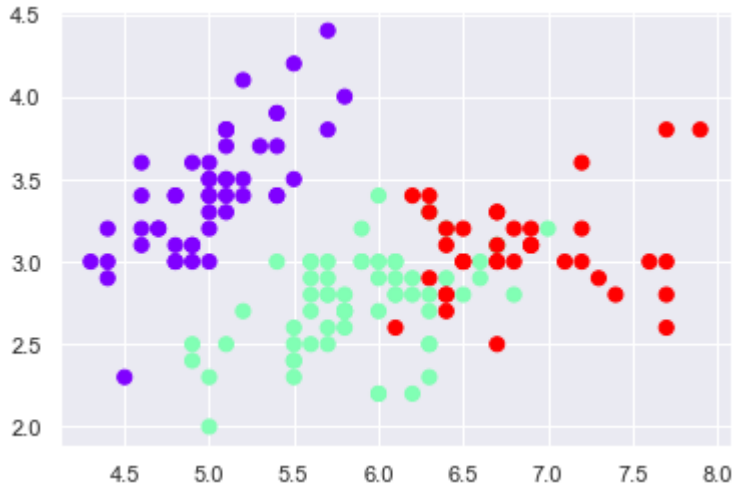
Out[17]:
```
array([[5.006     , 3.428     , 1.462     , 0.246     ],
       [5.9016129 , 2.7483871 , 4.39354839, 1.43387097],
       [6.85      , 3.07368421, 5.74210526, 2.07105263]])
```

In [18]:
```
n_cluster=3
silhouette_avg=silhouette_score(X,y_kmeans)
print("For n_cluster= ",n_cluster,"average silhouette_score is :",silhouette_a
vg)
```

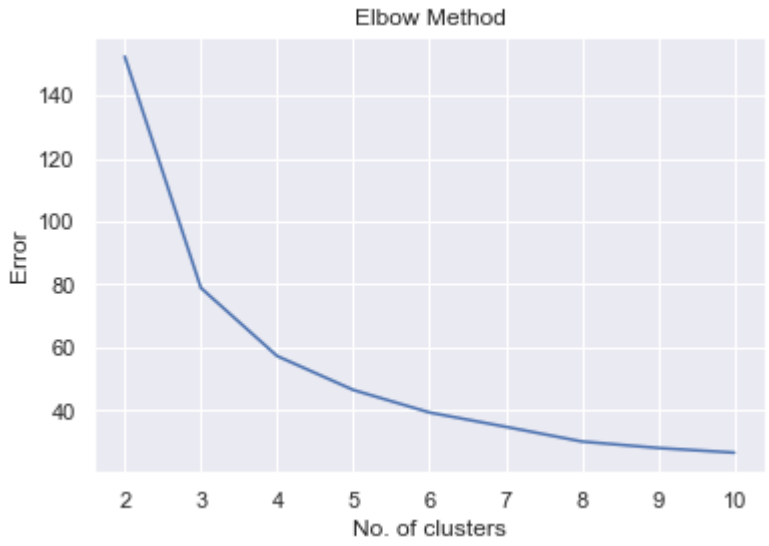
```
For n_cluster=  3 average silhouette_score is : 0.5528190123564102
```

In [20]:
```
plt.scatter(X[:,0],X[:,1],c=y_kmeans,s=50,cmap='rainbow')
```

Out[20]: <matplotlib.collections.PathCollection at 0x27918af4070>



In [21]:
```
Error=[]                              #for selecting no. of clusters
for i in range(2,11):
    kmeans=KMeans(n_clusters=i).fit(X)
    Error.append(kmeans.inertia_)
import matplotlib.pyplot as plt
plt.plot(range(2,11),Error)
plt.title('Elbow Method')
plt.xlabel('No. of clusters')
plt.ylabel('Error')
plt.show()
```



we will select 4