KMeans_Clustering_LabeledData

September 17, 2019

[1]:
```
# Using scikit-learn to perform K-Means clustering
from sklearn.cluster import KMeans
from sklearn.datasets import load_digits
digits = load_digits()
digits.data.shape
```

[1]:
```
(1797, 64)
```

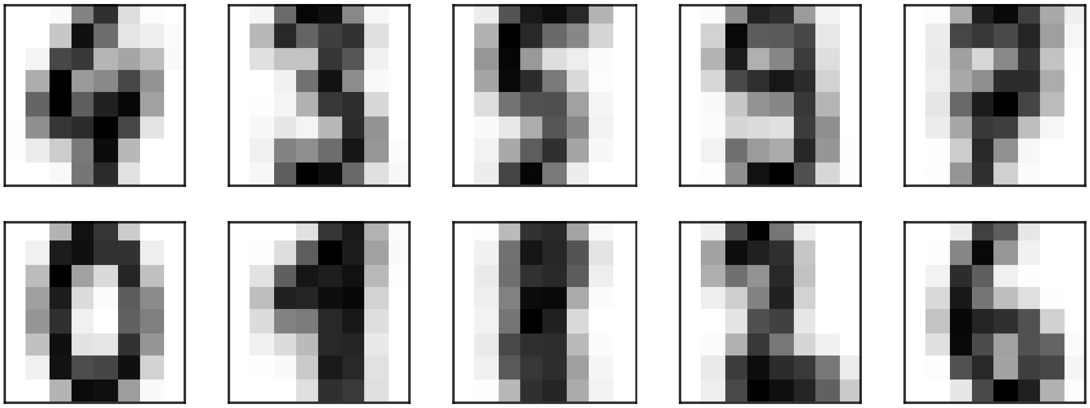
[10]:
```
kmeans = KMeans(n_clusters=10, random_state=0)
prediction = kmeans.fit_predict(digits.data)
prediction
```

[10]:
```
array([5, 7, 7, ..., 7, 3, 3], dtype=int32)
```

[5]:
```
kmeans.cluster_centers_.shape
```

[5]:
```
(10, 64)
```

[7]:
```
# Scatter plot of the data points
import matplotlib.pyplot as plt

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)
```

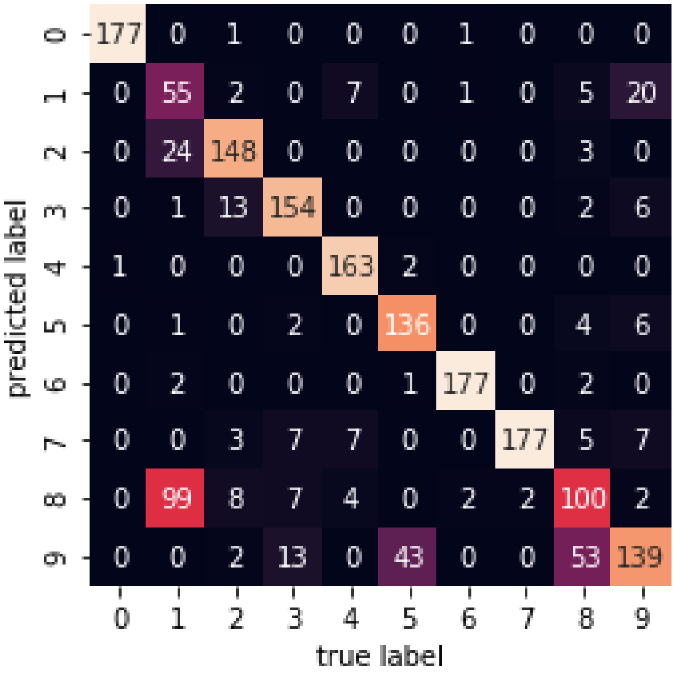


[21]:
```
import numpy as np
from scipy.stats import mode

labels = np.zeros_like(prediction)
for i in range(10):
    mask = (prediction == i)
    labels[mask] = mode(digits.target[mask])[0]
```

[22]:
```
from sklearn.metrics import accuracy_score
accuracy_score(digits.target, labels)
```

[22]:
```
0.7935447968836951
```

[23]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns

mat = confusion_matrix(digits.target, labels)
ax = sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=False,
            xticklabels=digits.target_names,
            yticklabels=digits.target_names)
ax.set_ylim(10.0,0)
plt.xlabel('true label')
plt.ylabel('predicted label');
```



	0	1	2	3	4	5	6	7	8	9
0	177	0	1	0	0	0	1	0	0	0
1	0	55	2	0	7	0	1	0	5	20
2	0	24	148	0	0	0	0	0	3	0
3	0	1	13	154	0	0	0	0	2	6
4	1	0	0	0	163	2	0	0	0	0
5	0	1	0	2	0	136	0	0	4	6
6	0	2	0	0	0	1	177	0	2	0
7	0	0	3	7	7	0	0	177	5	7
8	0	99	8	7	4	0	2	2	100	2
9	0	0	2	13	0	43	0	0	53	139

[ ]:
```

```