

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=23)
clf = RandomForestClassifier(n_estimators=200
                             , random_state=42)
clf.fit(X_train, y_train)

RandomForestClassifier(n_estimators=200, random_state=42)

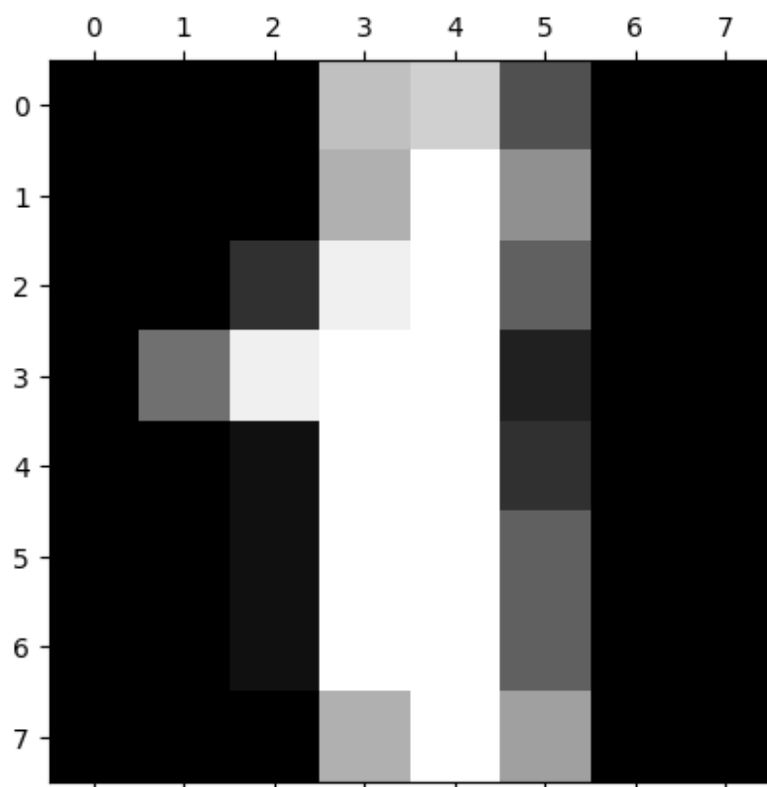
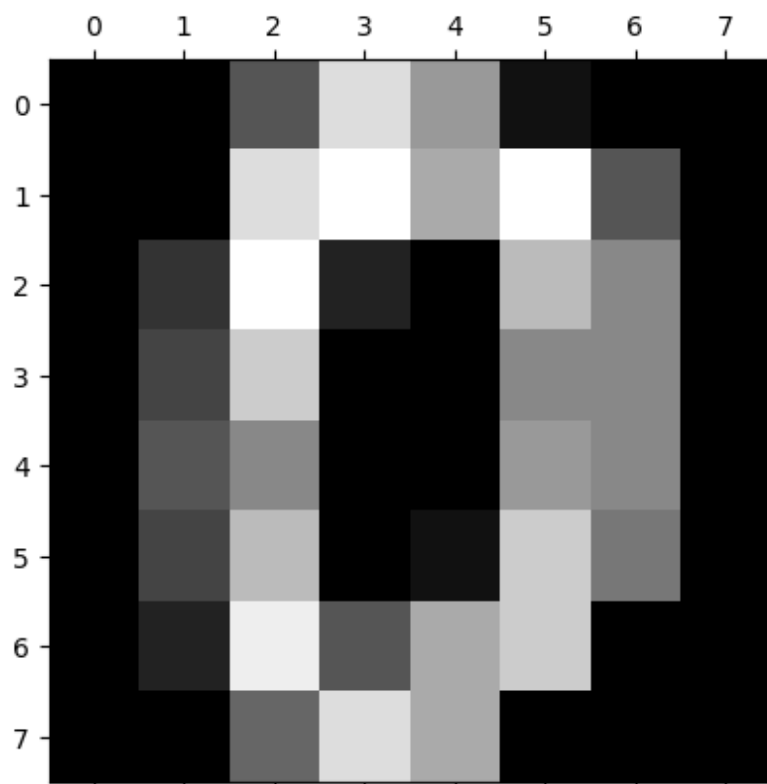
clf.score(X_test, y_test)

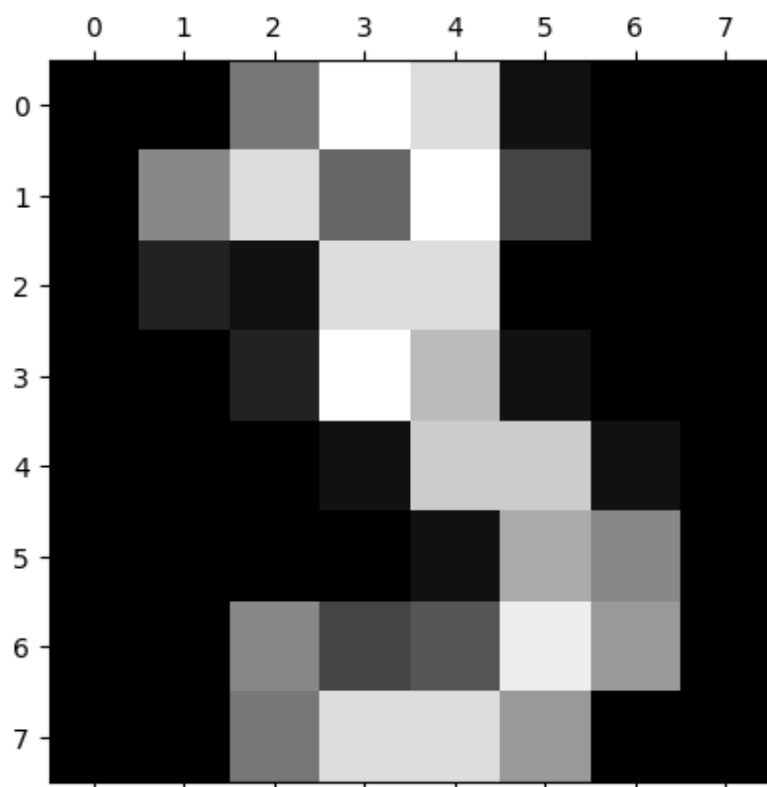
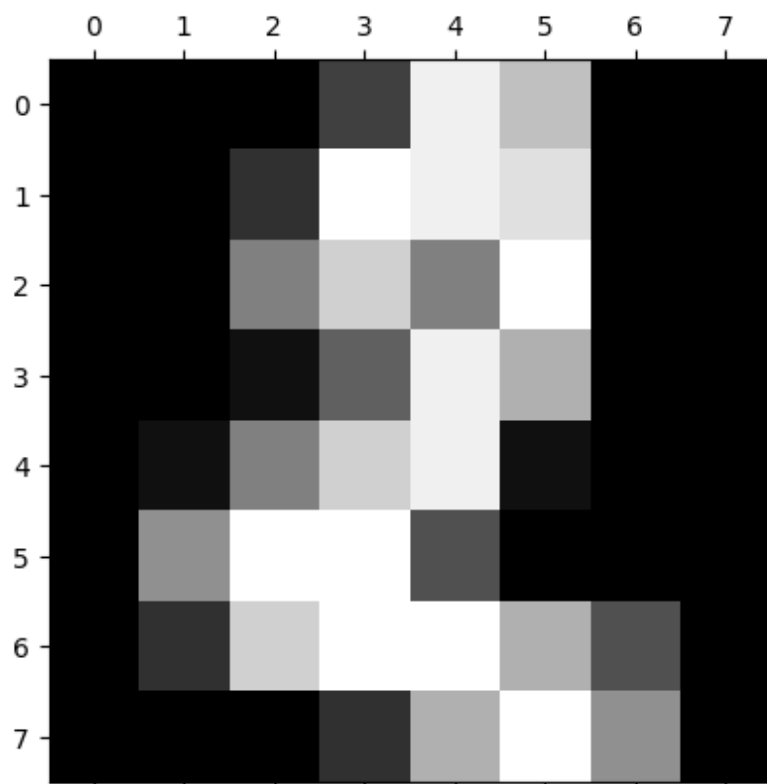
0.9666666666666667

from sklearn.datasets import load_digits
digits = load_digits()
import pandas as pd

import matplotlib.pyplot as plt
plt.gray()
for i in range(4):
    plt.matshow(digits.images[i])

<Figure size 640x480 with 0 Axes>
```





```
X, y = load_digits(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
clf = RandomForestClassifier(n_estimators=1, random_state=42)
clf.fit(X_train, y_train)
clf.score(X_test, y_test)
```

0.8166666666666667

```
df = pd.DataFrame(digits.data, columns=digits.feature_names)
df
```

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5
\						
0	0.0	0.0	5.0	13.0	9.0	1.0
1	0.0	0.0	0.0	12.0	13.0	5.0
2	0.0	0.0	0.0	4.0	15.0	12.0
3	0.0	0.0	7.0	15.0	13.0	1.0
4	0.0	0.0	0.0	1.0	11.0	0.0
...
1792	0.0	0.0	4.0	10.0	13.0	6.0
1793	0.0	0.0	6.0	16.0	13.0	11.0
1794	0.0	0.0	1.0	11.0	15.0	1.0
1795	0.0	0.0	2.0	10.0	7.0	0.0
1796	0.0	0.0	10.0	14.0	8.0	1.0

	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_6
pixel_6_7 \						
0	0.0	0.0	0.0	0.0	...	0.0
0.0						
1	0.0	0.0	0.0	0.0	...	0.0
0.0						
2	0.0	0.0	0.0	0.0	...	5.0
0.0						
3	0.0	0.0	0.0	8.0	...	9.0
0.0						
4	0.0	0.0	0.0	0.0	...	0.0
0.0						
...
...						

1792	0.0	0.0	0.0	1.0	...	4.0
0.0						
1793	1.0	0.0	0.0	0.0	...	1.0
0.0						
1794	0.0	0.0	0.0	0.0	...	0.0
0.0						
1795	0.0	0.0	0.0	0.0	...	2.0
0.0						
1796	0.0	0.0	0.0	2.0	...	8.0
0.0						
	pixel_7_0	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5
\						
0	0.0	0.0	6.0	13.0	10.0	0.0
1	0.0	0.0	0.0	11.0	16.0	10.0
2	0.0	0.0	0.0	3.0	11.0	16.0
3	0.0	0.0	7.0	13.0	13.0	9.0
4	0.0	0.0	0.0	2.0	16.0	4.0
...
1792	0.0	0.0	2.0	14.0	15.0	9.0
1793	0.0	0.0	6.0	16.0	14.0	6.0
1794	0.0	0.0	2.0	9.0	13.0	6.0
1795	0.0	0.0	5.0	12.0	16.0	12.0
1796	0.0	1.0	8.0	12.0	14.0	12.0
	pixel_7_6	pixel_7_7				
0	0.0	0.0				
1	0.0	0.0				
2	9.0	0.0				
3	0.0	0.0				
4	0.0	0.0				
...				
1792	0.0	0.0				
1793	0.0	0.0				
1794	0.0	0.0				
1795	0.0	0.0				
1796	1.0	0.0				
[1797 rows x 64 columns]						

```
df['target']= digits.target
```

```
df.head()
```

	pixel_0_0	pixel_0_1	pixel_0_2	pixel_0_3	pixel_0_4	pixel_0_5	\
0	0.0	0.0	5.0	13.0	9.0	1.0	
1	0.0	0.0	0.0	12.0	13.0	5.0	
2	0.0	0.0	0.0	4.0	15.0	12.0	
3	0.0	0.0	7.0	15.0	13.0	1.0	
4	0.0	0.0	0.0	1.0	11.0	0.0	

	pixel_0_6	pixel_0_7	pixel_1_0	pixel_1_1	...	pixel_6_7	\
0	0.0	0.0	0.0	0.0	...	0.0	
1	0.0	0.0	0.0	0.0	...	0.0	
2	0.0	0.0	0.0	0.0	...	0.0	
3	0.0	0.0	0.0	8.0	...	0.0	
4	0.0	0.0	0.0	0.0	...	0.0	

	pixel_7_1	pixel_7_2	pixel_7_3	pixel_7_4	pixel_7_5	pixel_7_6	\
0	0.0	6.0	13.0	10.0	0.0	0.0	
1	0.0	0.0	11.0	16.0	10.0	0.0	
2	0.0	0.0	3.0	11.0	16.0	9.0	
3	0.0	7.0	13.0	13.0	9.0	0.0	
4	0.0	0.0	2.0	16.0	4.0	0.0	

	pixel_7_7	target
0	0.0	0
1	0.0	1
2	0.0	2
3	0.0	3
4	0.0	4

```
[5 rows x 65 columns]
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
```

```
X, y = load_digits(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=54)
```

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
RandomForestClassifier(random_state=42)
```

```

y_pred = model.score(X_test, y_test)
print(y_pred*100)

98.05555555555556

import seaborn as sns
y_pred1 = model.predict(X_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred1)
cm

array([[30,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 35,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 36,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 35,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0, 39,  0,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  1, 39,  1,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0, 36,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 38,  0,  0],
       [ 0,  1,  0,  0,  0,  0,  0,  0, 33,  1],
       [ 0,  0,  0,  1,  1,  0,  0,  0,  0, 32]])

import matplotlib.pyplot as plt
plt.figure(figsize=(12, 8))
sns.heatmap(cm, annot=True, cmap="YlGnBu")
plt.title('Confusion matrix')
plt.ylabel('True label')
plt.xlabel('Predicted label')

Text(0.5, 58.72222222222222, 'Predicted label')

```

