```
import numpy as np
from sklearn.datasets import load_digits
```

```
dataset = load_digits()
print(dataset.data)
print(dataset.target)
```

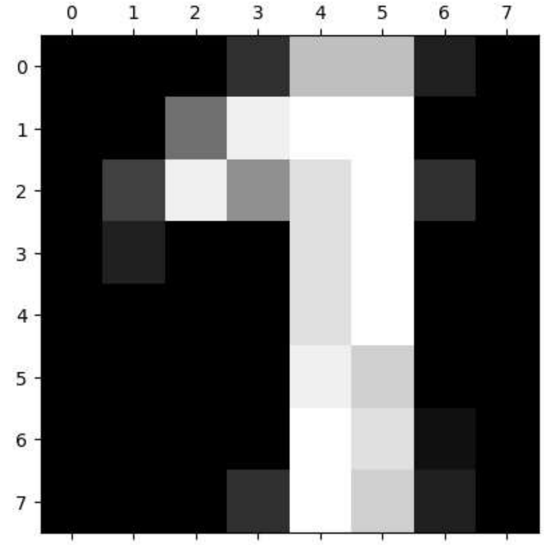
```
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]
[0 1 2 ... 8 9 8]
```

```
print(dataset.data.shape)
print(dataset.images.shape)
dataimageLength = len(dataset.images)
print(dataimageLength)
```

```
(1797, 64)
(1797, 8, 8)
1797
```

```
n = 1500
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(dataset.images[n])
plt.show()
```

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



```
dataset.images[n]
```

```
array([[ 0.,  0.,  0.,  3., 12., 12.,  2.,  0.],
       [ 0.,  0.,  7., 15., 16., 16.,  0.,  0.],
       [ 0.,  4., 15.,  9., 14., 16.,  3.,  0.],
       [ 0.,  2.,  0.,  0., 14., 16.,  0.,  0.],
       [ 0.,  0.,  0.,  0., 14., 16.,  0.,  0.],
       [ 0.,  0.,  0.,  0., 15., 13.,  0.,  0.],
       [ 0.,  0.,  0.,  0., 16., 14.,  1.,  0.],
       [ 0.,  0.,  0.,  3., 16., 13.,  2.,  0.]])
```

```
X = dataset.images.reshape((dataimageLength,-1))
X
```

```
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.]])
```

```
Y = dataset.target
Y
```

```
     array([0, 1, 2, ..., 8, 9, 8])
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size = 0.25,random_state= 0)
```

```
print(X_train.shape)
print(X_test.shape)
```

```
     (1347, 64)
     (450, 64)
```

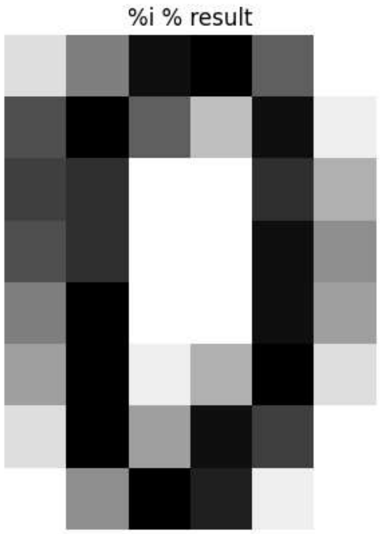
```
from sklearn import svm
model = svm.SVC(gamma= 0.001)
model.fit(X_train,Y_train)
```

```
     ▾      SVC
     SVC(gamma=0.001)
```

```
 n = 1205
result = model.predict(dataset.images[n].reshape((1,-1)))
```

```
plt.imshow(dataset.images[n],cmap= plt.cm.gray_r, interpolation = 'nearest')
print(result)
print("/n")
plt.axis("off")
plt.title("%i % result")
plt.show()
```

```
     [0]
     /n
```



```
Y_pred = model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print("Accuracy of the model:{0}%".format(accuracy_score(Y_test,Y_pred)*100))
```

```
     Accuracy of the model:99.55555555555556%
```

✓ 0s completed at 4:21 PM