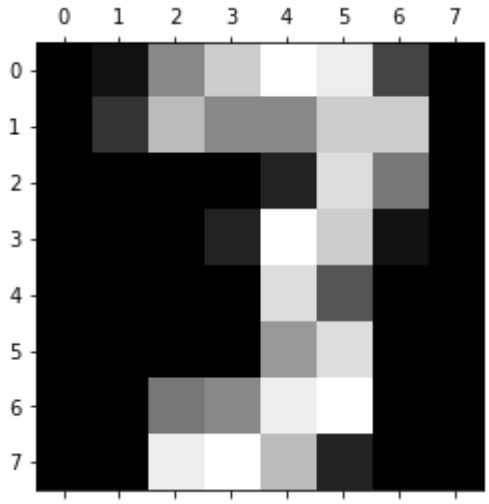
```
from sklearn import datasets
import matplotlib.pyplot as plt
import numpy as np


from sklearn.datasets import load_digits

dgts = load_digits()
print(dgts.data.shape)
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(dgts.images[23])
plt.show()
```

```
(1797, 64)
<Figure size 432x288 with 0 Axes>
```




```
digits = datasets.load_digits()


fig = plt.figure()
plt.imshow(digits.images[23],cmap = plt.cm.gray_r)
txt = "This is %d"%digits.target[23]
fig.text(0.1,0.1,txt)
plt.show()
```

```
digits.images[23]
```

```
array([[ 0.,  1.,  8., 12., 15., 14.,  4.,  0.],
       [ 0.,  3., 11.,  8.,  8., 12., 12.,  0.],
       [ 0.,  0.,  0.,  0.,  2., 13.,  7.,  0.],
       [ 0.,  0.,  0.,  2., 15., 12.,  1.,  0.],
       [ 0.,  0.,  0.,  0., 13.,  5.,  0.,  0.],
       [ 0.,  0.,  0.,  0.,  9., 13.,  0.,  0.],
       [ 0.,  0.,  7.,  8., 14., 15.,  0.,  0.],
       [ 0.,  0., 14., 15., 11.,  2.,  0.,  0.]])
```

```
x = 100 #length of training data set
X_train = digits.data[0:x]
Y_train = digits.target[0:x]
```

```
pred = 813
X_test = digits.data[pred]
print("X_test's real value is %d"%digits.target[pred])
```

```
X_test's real value is 9
```

```
def dist(x,y):
 return np.sqrt(np.sum((x-y)**2))
```

```
l = len(X_train)
distance = np.zeros(l)
for i in range(l):
 distance[i] = dist(X_train[i],X_test)
min_index = np.argmin(distance)
print("Preditcted value is ",)
print(Y_train[min_index])
```

```
Preditcted value is
9
```

```
l = len(X_train)
no_err = 0
distance = np.zeros(l)
for j in range(1697,1797):
 X_test = digits.data[j]
 for i in range(l):
  distance[i] = dist(X_train[i],X_test)
 min_index = np.argmin(distance)
 if Y_train[min_index] != digits.target[j]:
  no_err+=1
print("Total errors for train length = %d is %d"%(x,no_err))
```

```
Total errors for train length = 100 is 14
```