```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

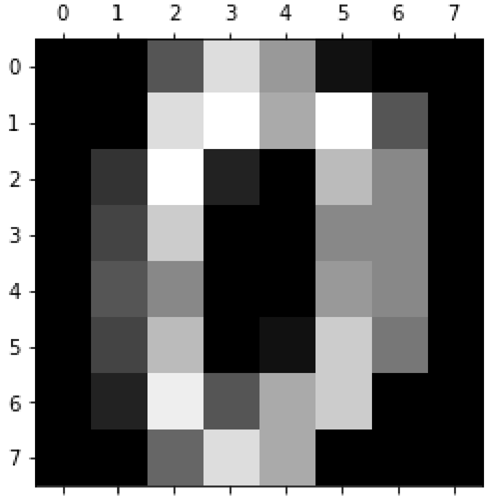
```
from sklearn.datasets import load_digits
df = load_digits()
```

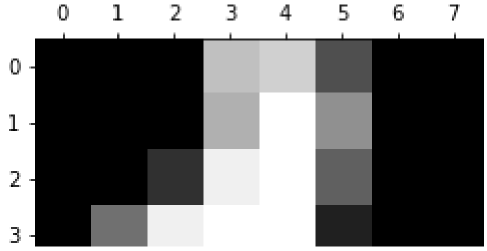
```
dir(df)
```

```
    ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

```
%matplotlib inline
plt.gray()
for i in range(0,4):
  plt.matshow(df.images[i])
```

```
<Figure size 432x288 with 0 Axes>
```





```
df.images.shape
```

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

```
df.images[0]
```

```
array([[ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.],
       [ 0.,  0., 13., 15., 10., 15.,  5.,  0.],
       [ 0.,  3., 15.,  2.,  0., 11.,  8.,  0.],
       [ 0.,  4., 12.,  0.,  0.,  8.,  8.,  0.],
       [ 0.,  5.,  8.,  0.,  0.,  9.,  8.,  0.],
       [ 0.,  4., 11.,  0.,  1., 12.,  7.,  0.],
       [ 0.,  2., 14.,  5., 10., 12.,  0.,  0.],
       [ 0.,  0.,  6., 13., 10.,  0.,  0.,  0.]])
```

```
df1 =pd.DataFrame(df.data)
```

```
df1['target'] = df.target
```

```
df1.head()
```

```
        0    1    2    3     4     5    6    7    8    9   ...  55   56   57   58   59    60
```

```
x = df1.drop('target',axis =1)
```

```
   1  0.0  0.0  0.0  12.0  13.0  5.0  0.0  0.0  0.0  0.0  ...  0.0  0.0  0.0  0.0  11.0  16.0
```

```
y=df1['target']
```

```
   3  0.0  0.0  7.0  15.0  13.0  1.0  0.0  0.0  0.0  8.0  ...  0.0  0.0  0.0  7.0  13.0  13.0
```

```
from sklearn.preprocessing import MinMaxScaler
```

```
sc = MinMaxScaler()
sc.fit_transform(x)
```

```
    array([[0.    , 0.    , 0.3125, ..., 0.    , 0.    , 0.    ],
           [0.    , 0.    , 0.    , ..., 0.625 , 0.    , 0.    ],
           [0.    , 0.    , 0.    , ..., 1.    , 0.5625, 0.    ],
           ...,
           [0.    , 0.    , 0.0625, ..., 0.375 , 0.    , 0.    ],
           [0.    , 0.    , 0.125 , ..., 0.75  , 0.    , 0.    ],
           [0.    , 0.    , 0.625 , ..., 0.75  , 0.0625, 0.    ]])
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size = 0.2,random_state=29)
```

```
from sklearn.ensemble import RandomForestClassifier

rf=RandomForestClassifier()
rf.fit(xtrain,ytrain)
rf.score(xtest,ytest)
```

```
    0.9777777777777777
```

```
from sklearn.metrics import accuracy_score
ypred = rf.predict(xtest)
ytpred = rf.predict(xtrain)
print('testing score: ',accuracy_score(ytest,ypred))
print('training score: ',accuracy_score(ytrain,ytpred))
```

```
    testing score:  0.9777777777777777
    training score:  1.0
```

```
from sklearn.metrics import classification_report,confusion_matrix
print('confusion matrix: \n',confusion_matrix(ytest,ypred))
```

```
    confusion matrix: 
     [[42  0  0  0  2  0  0  0  0  0]
     [ 0 41  0  0  0  0  0  0  0  0]
     [ 1  0 38  0  0  0  0  0  0  0]
     [ 0  0  0 42  0  1  0  0  0  0]
     [ 0  0  0  0 35  0  0  1  0  0]
     [ 0  0  0  0  0 38  0  0  0  0]
```

```
 [ 0  0  0  0  0  0 26  0  0  0]
 [ 0  0  0  0  0  0  0 30  0  0]
 [ 0  1  0  0  0  0  0  0 24  0]
 [ 0  0  0  0  0  0  0  0  2 36]]
```

```
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

           0       0.98      0.95      0.97        44
           1       0.98      1.00      0.99        41
           2       1.00      0.97      0.99        39
           3       1.00      0.98      0.99        43
           4       0.95      0.97      0.96        36
           5       0.97      1.00      0.99        38
           6       1.00      1.00      1.00        26
           7       0.97      1.00      0.98        30
           8       0.92      0.96      0.94        25
           9       1.00      0.95      0.97        38

    accuracy                           0.98       360
   macro avg       0.98      0.98      0.98       360
weighted avg       0.98      0.98      0.98       360
```

✓ 0s completed at 11:47 AM