

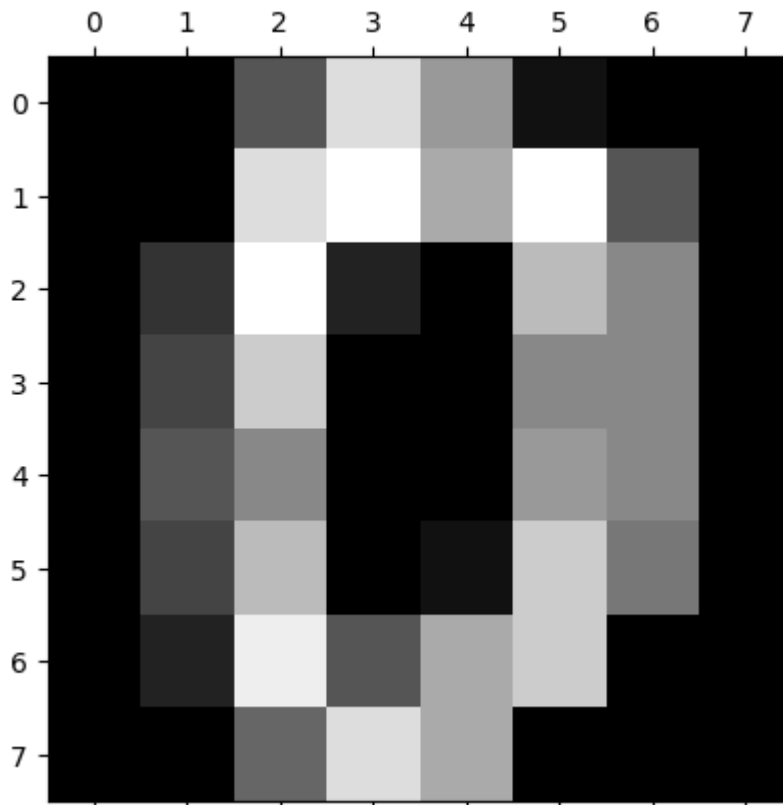
# Digits dataset from sklearn

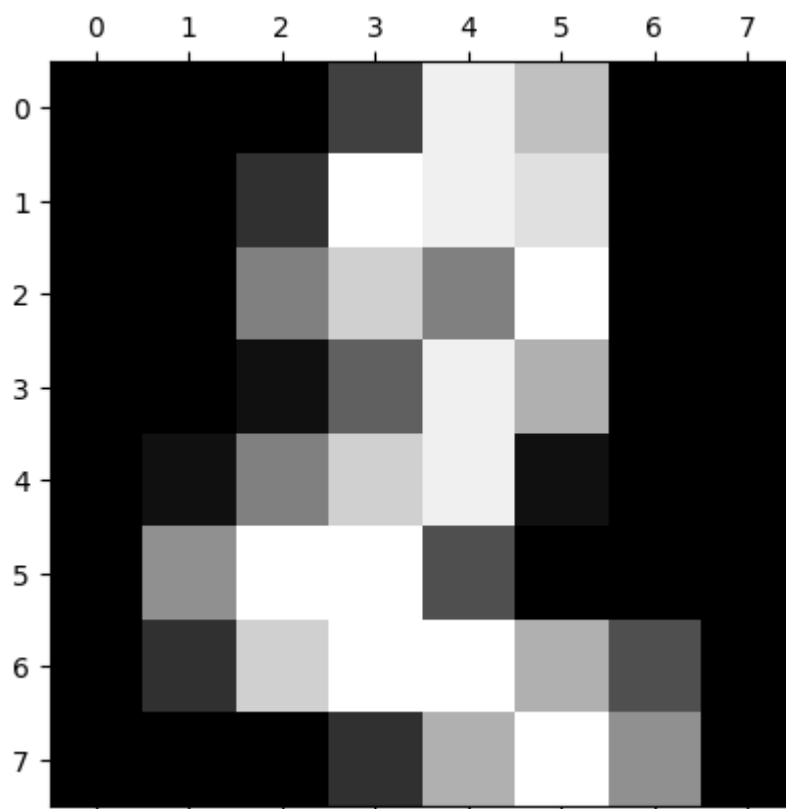
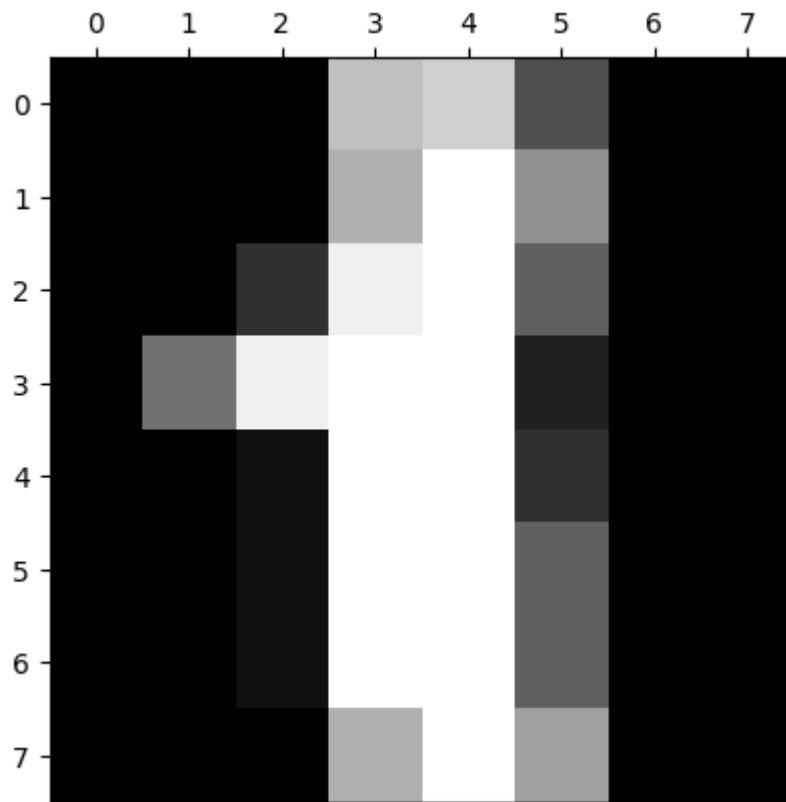
```
In [8]: import pandas as pd
        from sklearn.datasets import load_digits
        digits = load_digits()
        dir(digits)
```

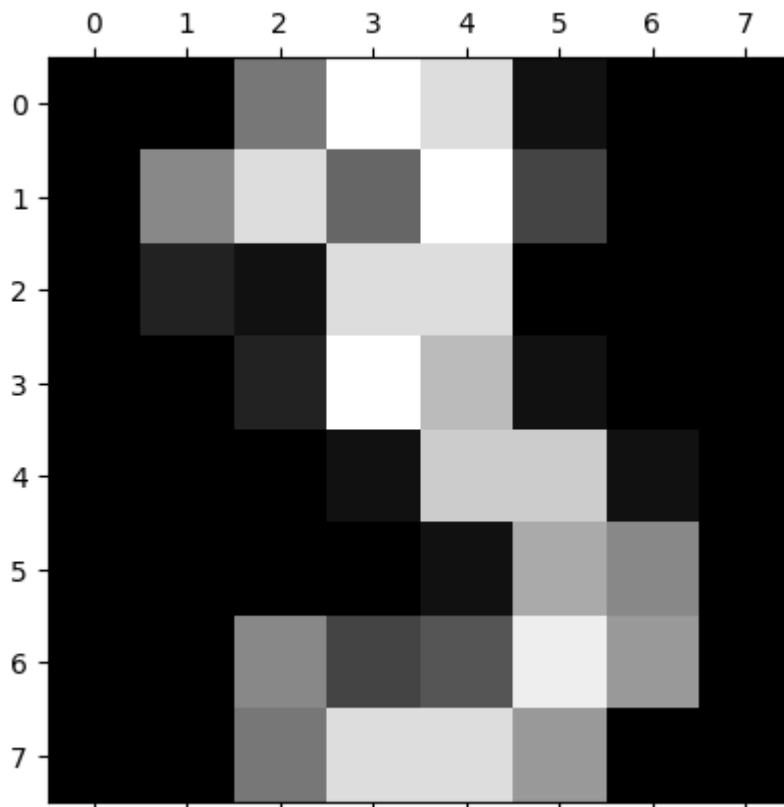
```
Out[8]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

```
In [15]: %matplotlib inline
        import matplotlib.pyplot as pp
        pp.gray()
        for i in range(4):
            pp.matshow(digits.images[i])
```

<Figure size 640x480 with 0 Axes>







```
In [19]: digits.data[0:4]
```

```
Out[19]: 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.],
       [ 0.,  0.,  0., 12., 13.,  5.,  0.,  0.,  0.,  0.,  0., 11., 16.,
         9.,  0.,  0.,  0.,  0.,  3., 15., 16.,  6.,  0.,  0.,  0.,  7.,
        15., 16., 16.,  2.,  0.,  0.,  0.,  0.,  1., 16., 16.,  3.,  0.,
         0.,  0.,  0.,  1., 16., 16.,  6.,  0.,  0.,  0.,  0.,  1., 16.,
        16.,  6.,  0.,  0.,  0.,  0.,  0.,  0., 11., 16., 10.,  0.,  0.],
       [ 0.,  0.,  0.,  4., 15., 12.,  0.,  0.,  0.,  0.,  3., 16., 15.,
        14.,  0.,  0.,  0.,  0.,  8., 13.,  8., 16.,  0.,  0.,  0.,  0.,
         1.,  6., 15., 11.,  0.,  0.,  0.,  1.,  8., 13., 15.,  1.,  0.,
         0.,  0.,  9., 16., 16.,  5.,  0.,  0.,  0.,  0.,  3., 13., 16.,
        16., 11.,  5.,  0.,  0.,  0.,  0.,  3., 11., 16.,  9.,  0.],
       [ 0.,  0.,  7., 15., 13.,  1.,  0.,  0.,  0.,  8., 13.,  6., 15.,
         4.,  0.,  0.,  0.,  2.,  1., 13., 13.,  0.,  0.,  0.,  0.,  0.,
         2., 15., 11.,  1.,  0.,  0.,  0.,  0.,  0.,  1., 12., 12.,  1.,
         0.,  0.,  0.,  0.,  0.,  1., 10.,  8.,  0.,  0.,  0.,  8.,  4.,
         5., 14.,  9.,  0.,  0.,  0.,  7., 13., 13.,  9.,  0.,  0.]])
```

```
In [20]: df = pd.DataFrame(digits.data)
df
```

Out[20]:

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59
<b>0</b>	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	6.0	13.0
<b>1</b>	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	0.0	11.0
<b>2</b>	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	...	5.0	0.0	0.0	0.0	0.0	3.0
<b>3</b>	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	...	9.0	0.0	0.0	0.0	7.0	13.0
<b>4</b>	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	2.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
<b>1792</b>	0.0	0.0	4.0	10.0	13.0	6.0	0.0	0.0	0.0	1.0	...	4.0	0.0	0.0	0.0	2.0	14.0
<b>1793</b>	0.0	0.0	6.0	16.0	13.0	11.0	1.0	0.0	0.0	0.0	...	1.0	0.0	0.0	0.0	6.0	16.0
<b>1794</b>	0.0	0.0	1.0	11.0	15.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	9.0
<b>1795</b>	0.0	0.0	2.0	10.0	7.0	0.0	0.0	0.0	0.0	0.0	...	2.0	0.0	0.0	0.0	5.0	12.0
<b>1796</b>	0.0	0.0	10.0	14.0	8.0	1.0	0.0	0.0	0.0	2.0	...	8.0	0.0	0.0	1.0	8.0	12.0

1797 rows × 64 columns

In [21]: `digits.target`Out[21]: `array([0, 1, 2, ..., 8, 9, 8])`In [23]: `df['target'] = digits.target`  
`df`

Out[23]:

	0	1	2	3	4	5	6	7	8	9	...	55	56	57	58	59	60
<b>0</b>	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	13.0	10.0
<b>1</b>	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
<b>2</b>	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	11.0
<b>3</b>	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
<b>4</b>	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	16.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
<b>1792</b>	0.0	0.0	4.0	10.0	13.0	6.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	2.0	14.0	15.0
<b>1793</b>	0.0	0.0	6.0	16.0	13.0	11.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	16.0	14.0
<b>1794</b>	0.0	0.0	1.0	11.0	15.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	2.0	9.0	13.0
<b>1795</b>	0.0	0.0	2.0	10.0	7.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	5.0	12.0	16.0
<b>1796</b>	0.0	0.0	10.0	14.0	8.0	1.0	0.0	0.0	0.0	2.0	...	0.0	0.0	1.0	8.0	12.0	14.0

1797 rows × 65 columns



# Train and the model and prediction

```
In [25]: X=df  
         y=digits.target
```

```
In [26]: from sklearn.model_selection import train_test_split  
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```
In [27]: len(X_train)
```

```
Out[27]: 1437
```

```
In [28]: len(X_test)
```

```
Out[28]: 360
```

```
In [43]: from sklearn.ensemble import RandomForestClassifier  
         modal=RandomForestClassifier()
```

```
In [44]: modal.fit(X_train,y_train)
```

```
c:\users\sriha\appdata\local\programs\python\python37\lib\site-packages\sklearn\utils\validation.py:1692: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.  
FutureWarning,
```

```
Out[44]: RandomForestClassifier()
```

```
In [45]: modal.score(X_test,y_test)
```

```
c:\users\sriha\appdata\local\programs\python\python37\lib\site-packages\sklearn\utils\validation.py:1692: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.  
FutureWarning,
```

```
Out[45]: 0.9972222222222222
```

```
In [47]: y_pred = modal.predict(X_test)  
         y_pred
```

```
c:\users\sriha\appdata\local\programs\python\python37\lib\site-packages\sklearn\utils\validation.py:1692: FutureWarning: Feature names only support names that are all strings. Got feature names with dtypes: ['int', 'str']. An error will be raised in 1.2.  
FutureWarning,
```

```
Out[47]: array([8, 6, 5, 4, 9, 4, 4, 0, 0, 0, 2, 7, 1, 5, 6, 4, 3, 7, 9, 6, 4, 4,
                2, 9, 2, 0, 0, 8, 6, 1, 7, 2, 0, 2, 6, 4, 3, 9, 7, 9, 8, 7, 4, 1,
                5, 5, 1, 8, 8, 3, 1, 1, 2, 0, 1, 8, 9, 5, 8, 5, 7, 9, 0, 3, 0, 9,
                2, 8, 4, 9, 5, 4, 3, 5, 3, 9, 4, 1, 5, 7, 6, 9, 0, 9, 6, 0, 8, 8,
                6, 1, 2, 8, 0, 2, 2, 9, 6, 2, 2, 3, 1, 7, 5, 4, 0, 0, 4, 7, 2, 4,
                0, 5, 3, 7, 7, 7, 5, 4, 3, 1, 1, 8, 3, 9, 3, 5, 9, 9, 2, 3, 4, 6,
                3, 0, 6, 8, 3, 1, 3, 9, 4, 8, 2, 1, 6, 3, 3, 5, 8, 0, 2, 8, 7, 4,
                2, 9, 1, 6, 1, 1, 3, 3, 2, 6, 9, 5, 1, 0, 2, 6, 3, 7, 9, 3, 5, 4,
                0, 1, 9, 9, 2, 7, 3, 9, 2, 0, 6, 4, 6, 0, 2, 2, 2, 0, 8, 1, 2, 9,
                7, 7, 2, 3, 2, 9, 3, 4, 8, 4, 8, 8, 7, 5, 6, 9, 1, 7, 6, 4, 0, 0,
                1, 1, 7, 9, 7, 7, 0, 0, 1, 8, 2, 1, 5, 7, 4, 9, 9, 7, 2, 6, 4, 6,
                7, 4, 8, 5, 1, 0, 1, 1, 3, 5, 1, 1, 3, 5, 5, 7, 9, 5, 8, 9, 4, 1,
                0, 9, 3, 9, 0, 0, 2, 4, 0, 5, 0, 7, 4, 3, 7, 7, 1, 2, 3, 3, 8, 4,
                5, 4, 8, 1, 5, 8, 2, 7, 9, 7, 0, 3, 6, 9, 8, 7, 0, 2, 4, 3, 4, 5,
                0, 8, 1, 1, 4, 4, 3, 9, 7, 3, 5, 7, 5, 0, 1, 2, 3, 8, 5, 0, 3, 6,
                4, 4, 4, 0, 3, 6, 2, 8, 2, 3, 3, 6, 9, 0, 6, 3, 2, 9, 6, 6, 6, 4,
                7, 0, 8, 1, 4, 2, 2, 0])
```

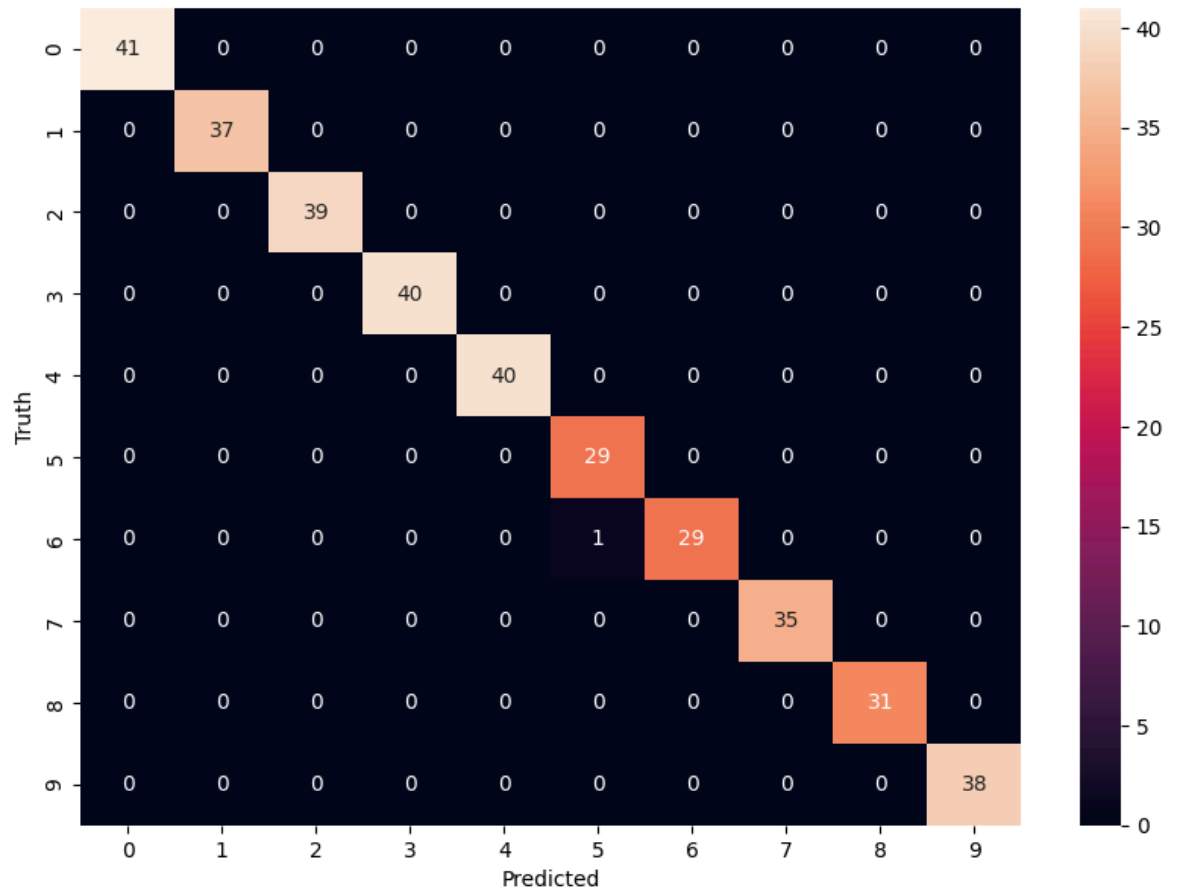
## Confusion Matrix

```
In [48]: from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test,y_pred)
cm
```

```
Out[48]: array([[41,  0,  0,  0,  0,  0,  0,  0,  0,  0],
                [ 0, 37,  0,  0,  0,  0,  0,  0,  0,  0],
                [ 0,  0, 39,  0,  0,  0,  0,  0,  0,  0],
                [ 0,  0,  0, 40,  0,  0,  0,  0,  0,  0],
                [ 0,  0,  0,  0, 40,  0,  0,  0,  0,  0],
                [ 0,  0,  0,  0,  0, 29,  0,  0,  0,  0],
                [ 0,  0,  0,  0,  0,  1, 29,  0,  0,  0],
                [ 0,  0,  0,  0,  0,  0,  0, 35,  0,  0],
                [ 0,  0,  0,  0,  0,  0,  0,  0, 31,  0],
                [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 38]], dtype=int64)
```

```
In [51]: %matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sn
plt.figure(figsize=(10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

```
Out[51]: Text(95.7222222222221, 0.5, 'Truth')
```



In [ ]: