In [1]:
```
import pandas as pd
from sklearn.datasets import load_digits
```

In [5]:
```
digits = load_digits()
digits
```

Out[5]:
```
{'data': 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.]]),
 'target': array([0, 1, 2, ..., 8, 9, 8]),
 'frame': None,
 'feature_names': ['pixel_0_0',
  'pixel_0_1',
  'pixel_0_2',
  'pixel_0_3',
  'pixel_0_4',
  'pixel_0_5',
  'pixel_0_6',
  'pixel_0_7',
  'pixel_1_0',
  'pixel_1_1',
```

In [6]:
```
dir(digits)
```

Out[6]:
```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

In [ ]:
```

```

In [9]:
```
df = pd.DataFrame(digits.data)
df
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	6
0	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	10.0	0.
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	0.0	11.0	16.0	10.
2	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	11.0	16.
3	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	13.0	9.
4	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	16.0	4.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
1792	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	15.0	9.
1793	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	14.0	6.
1794	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	13.0	6.
1795	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	16.0	12.
1796	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	14.0	12.

1797 rows × 64 columns

In [10]:
```
df['target'] = digits.target
df
```

Out[10]:

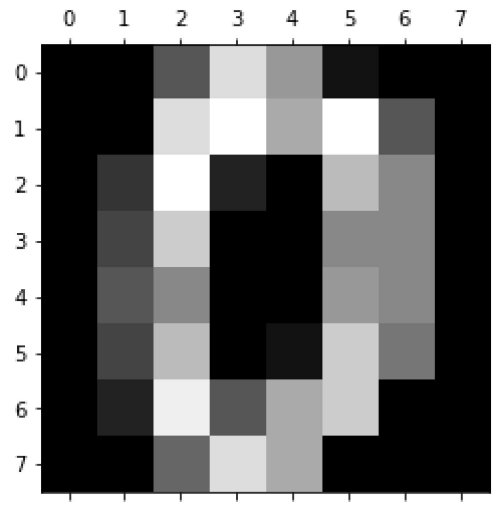
	0	1	2	3	4	5	6	7	8	9	...	55	56	57	58	59	60	61	6
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	3.0	11.0	16.0	9.
3	0.0	0.0	7.0	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	9.0	0.
4	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	4.0	0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
1792	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	9.0	0.
1793	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	6.0	0.
1794	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	6.0	0.
1795	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	12.0	0.
1796	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	12.0	1.

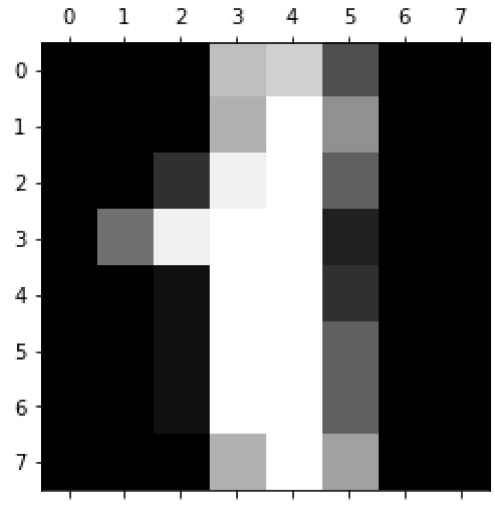
1797 rows × 65 columns

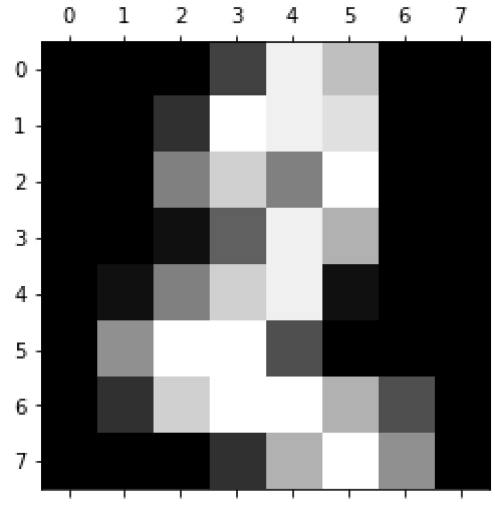
In [13]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

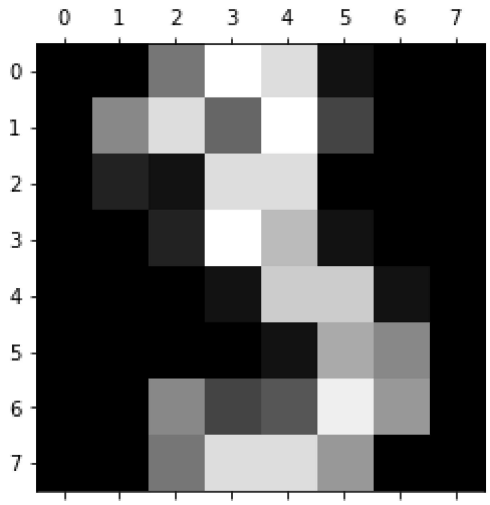
In [14]:
```
plt.gray()
for i in range (4):
    plt.matshow(digits.images[i])
```

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









In [15]:
```
x = df.drop('target' , axis = 'columns')
y = df.target
```

In [19]:
```
from sklearn.model_selection import train_test_split
```

In [21]:
```
x_train, x_test, y_train, y_test = train_test_split( x, y , test_size = 0.2)
```

In [ ]:
```

```

In [16]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [55]:
```
model = RandomForestClassifier(n_estimators =20)
```

In [56]:
```
model.fit(x_train , y_train)
```

Out[56]:
```
RandomForestClassifier(n_estimators=20)
```

In [57]:
```
model.score(x_test , y_test)
```

Out[57]:
```
0.9638888888888889
```

In [25]:
```
model.predict(x_test)
```

Out[25]:
```
array([1, 5, 4, 3, 9, 0, 6, 1, 1, 0, 1, 5, 2, 2, 4, 6, 5, 3, 5, 8, 7, 9,
       6, 6, 4, 8, 7, 7, 8, 5, 1, 7, 8, 2, 1, 1, 0, 8, 7, 3, 4, 7, 6, 9,
       8, 4, 5, 7, 1, 7, 7, 1, 7, 9, 6, 3, 1, 8, 7, 9, 4, 9, 7, 4, 2, 4,
       2, 7, 8, 0, 0, 5, 5, 9, 7, 2, 7, 1, 8, 9, 1, 9, 0, 1, 3, 4, 1, 6,
       4, 7, 2, 5, 9, 8, 2, 0, 4, 3, 7, 7, 1, 6, 3, 5, 1, 9, 7, 1, 5, 2,
       8, 3, 0, 4, 2, 8, 7, 8, 3, 9, 7, 5, 5, 6, 8, 8, 4, 1, 9, 9, 1, 4,
       2, 3, 0, 6, 0, 6, 6, 3, 1, 2, 0, 9, 1, 5, 7, 6, 5, 3, 9, 9, 3, 0,
       7, 1, 8, 5, 6, 0, 2, 2, 5, 8, 2, 1, 4, 9, 2, 8, 9, 7, 2, 4, 9, 8,
       8, 2, 7, 0, 3, 8, 5, 9, 6, 0, 0, 0, 0, 3, 5, 1, 0, 4, 9, 4, 4, 5,
       0, 5, 8, 2, 3, 7, 4, 3, 5, 2, 1, 4, 3, 4, 8, 6, 5, 4, 0, 4, 4, 0,
       1, 3, 5, 8, 8, 8, 8, 5, 4, 7, 5, 5, 2, 2, 2, 8, 3, 9, 3, 4, 1, 8,
       4, 7, 6, 1, 2, 1, 8, 8, 0, 0, 7, 0, 5, 4, 6, 3, 3, 8, 4, 4, 7, 5,
       9, 4, 5, 8, 8, 5, 0, 6, 8, 3, 7, 1, 0, 7, 7, 1, 2, 3, 6, 1, 2, 7,
       4, 9, 6, 9, 1, 8, 1, 3, 8, 4, 3, 3, 5, 2, 8, 2, 9, 8, 3, 7, 2, 8,
       9, 4, 2, 7, 0, 7, 5, 6, 8, 6, 3, 3, 7, 4, 7, 9, 3, 5, 1, 3, 2, 8,
       6, 3, 0, 1, 6, 1, 8, 1, 3, 2, 8, 8, 1, 5, 4, 2, 3, 0, 9, 8, 8, 5,
       4, 5, 0, 1, 4, 8, 6, 5])
```

In [28]:
```
y_predicted = model.predict(x_test)
```

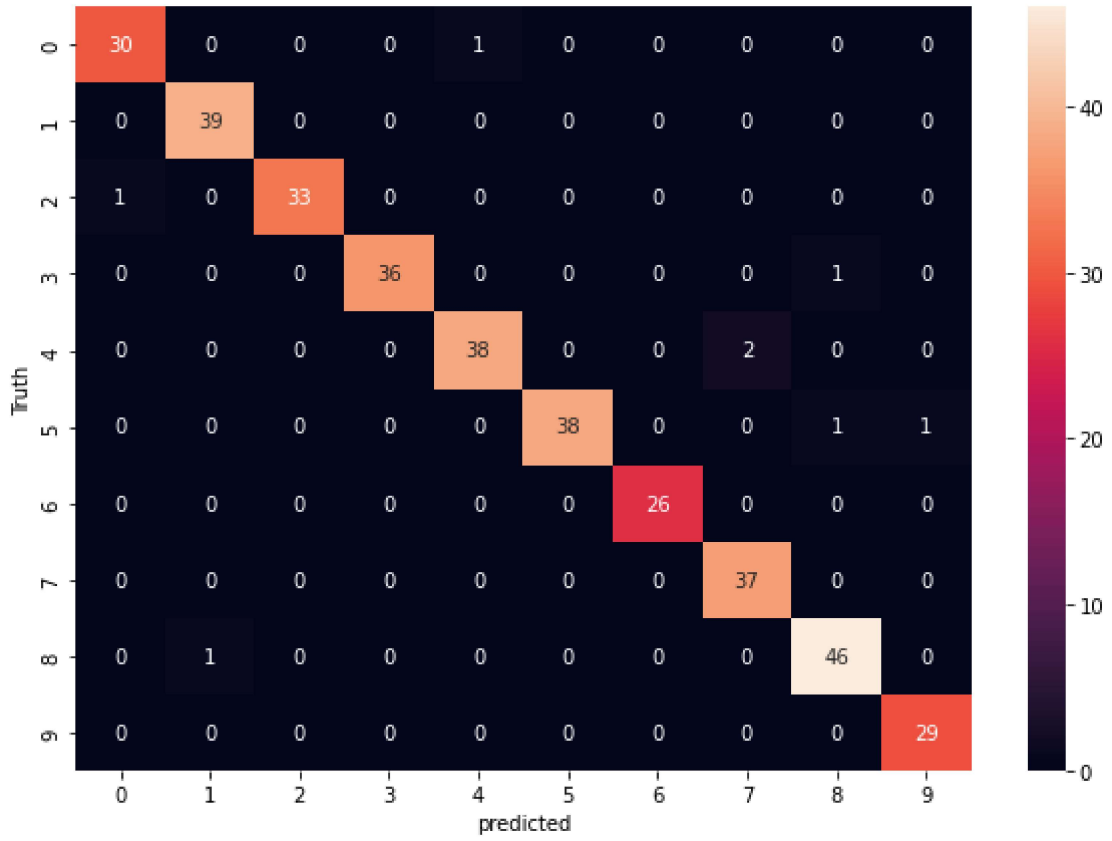
In [27]:
```
from sklearn.metrics import confusion_matrix
```

In [29]:
```
cm = confusion_matrix(y_test , y_predicted)
cm
```

Out[29]:
```
array([[30,  0,  0,  0,  1,  0,  0,  0,  0,  0],
       [ 0, 39,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 1,  0, 33,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 36,  0,  0,  0,  0,  1,  0],
       [ 0,  0,  0,  0, 38,  0,  0,  2,  0,  0],
       [ 0,  0,  0,  0,  0, 38,  0,  0,  1,  1],
       [ 0,  0,  0,  0,  0,  0, 26,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 37,  0,  0],
       [ 0,  1,  0,  0,  0,  0,  0,  0, 46,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 29]], dtype=int64)
```

In [31]:
```
%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[31]: Text(69.0, 0.5, 'Truth')



In [ ]:
```
```