In [1]:
```
'''
NAME: MANE SHIVRAJ PANDURANG
ROLL NO.37
COURSE: AI&DS, SUB:ML(Machine Learning)
CLASS: BE
'''
```

Out[1]:
```
'\nNAME: MANE SHIVRAJ PANDURANG\nROLL NO.37\nCOURSE: AI&DS, SUB:ML(Machine Learning)\nCLASS: BE \n'
```

In [2]:
```
'''
PRACTICAL.NO:05 (B)
    Use different voting mechanism and Apply AdaBoost (Adaptive Boosting), Gradient Tree
Boosting (GBM), XGBoost classification on Iris dataset and compare the performance of three
models using different evaluation measures.

'''
```

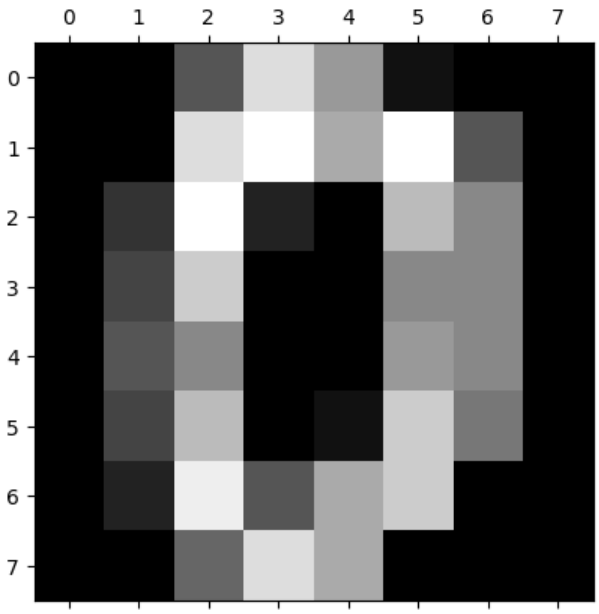
Out[2]:
```
'\nPRACTICAL.NO:05 (B)\n    Use different voting mechanism and Apply AdaBoost (Adaptive Boosting), Gradient Tree\nBo
osting (GBM), XGBoost classification on Iris dataset and compare the performance of three\nmodels using different ev
aluation measures.\n\n'
```

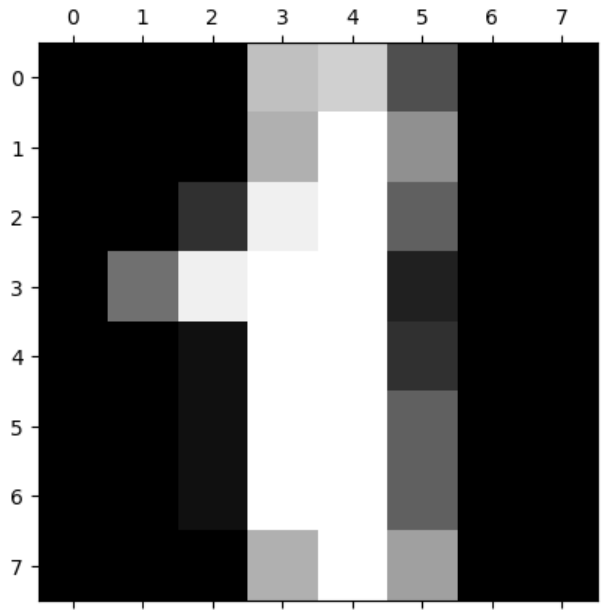
In [3]:
```
import pandas as pd
from sklearn.datasets import load_digits
```

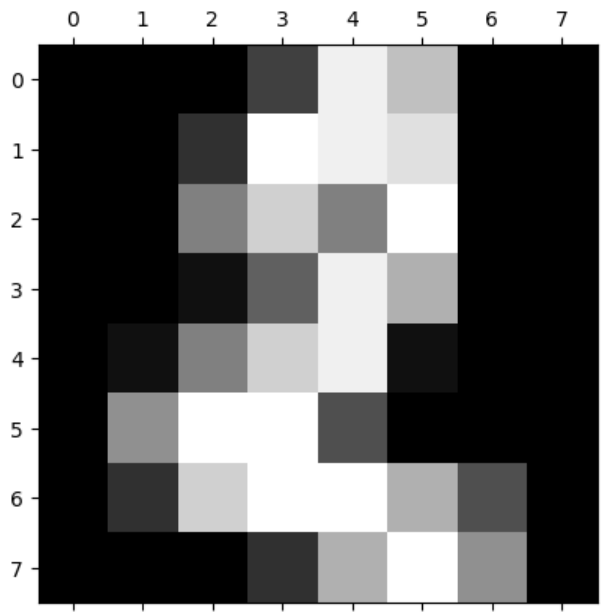
In [4]:
```
digits = load_digits()
dir(digits)
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
%matplotlib inline
import matplotlib.pyplot as plt
```

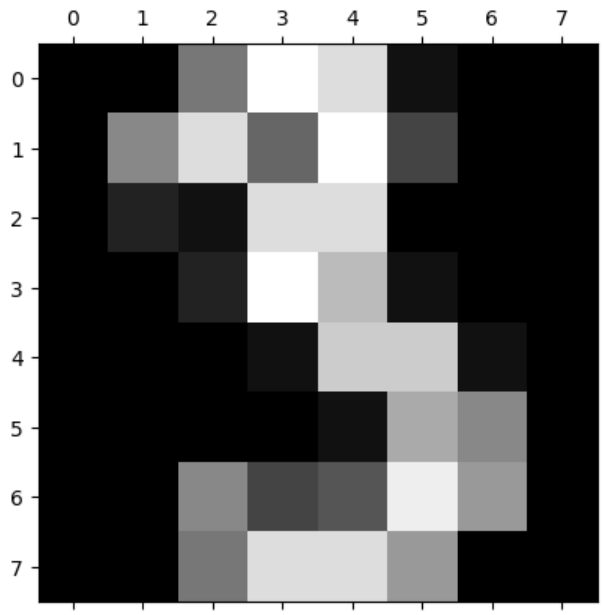
In [5]:
```
plt.gray()
for i in range(4):
 plt.matshow(digits.images[i])
```

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




0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7


0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7


0 1 2 3 4 5 6 7
0 1 2 3 4 5 6 7

In [6]:
```
df = pd.DataFrame(digits.data)
df.head()
```

Out[6]:

	0	1	2	3	4	5	6	7	8	9	...	54	55	56	57	58	59	60	61	62	63
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	0.0	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	0.0	11.0	16.0	10.0	0.0	0.0
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	16.0	9.0	0.0
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	9.0	0.0	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	0.0	2.0	16.0	4.0	0.0	0.0

5 rows × 64 columns

In [7]:
```
df['target'] = digits.target
df[0:12]
```

Out[7]:

	0	1	2	3	4	5	6	7	8	9	...	55	56	57	58	59	60	61	62	63	target
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	0.0	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	0.0	0.0	1
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	9.0	0.0	2
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	0.0	0.0	3
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	0.0	0.0	4
5	0.0	0.0	12.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	9.0	16.0	16.0	10.0	0.0	0.0	5
6	0.0	0.0	0.0	12.0	13.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	9.0	15.0	11.0	3.0	0.0	6
7	0.0	0.0	7.0	8.0	13.0	16.0	15.0	1.0	0.0	0.0	...	0.0	0.0	0.0	13.0	5.0	0.0	0.0	0.0	0.0	7
8	0.0	0.0	9.0	14.0	8.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	11.0	16.0	15.0	11.0	1.0	0.0	8
9	0.0	0.0	11.0	12.0	0.0	0.0	0.0	0.0	0.0	2.0	...	0.0	0.0	0.0	9.0	12.0	13.0	3.0	0.0	0.0	9
10	0.0	0.0	1.0	9.0	15.0	11.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	10.0	13.0	3.0	0.0	0.0	0
11	0.0	0.0	0.0	0.0	14.0	13.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	13.0	16.0	1.0	0.0	1

12 rows × 65 columns

In [8]:
```
# Train and the model and prediction
X = df.drop('target',axis='columns')
y = df.target
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)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=20)
model.fit(X_train, y_train)
RandomForestClassifier(n_estimators=20)
model.score(X_test, y_test)
```

Out[8]: 0.9472222222222222

In [9]:
```
y_predicted = model.predict(X_test)
```

In [10]:
```
# Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted)
cm
```

Out[10]:
```
array([[35,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 37,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 36,  1,  0,  0,  0,  1,  0,  1],
       [ 0,  0,  0, 42,  0,  0,  0,  0,  0,  1],
       [ 0,  0,  0,  0, 35,  0,  0,  0,  1,  0],
       [ 0,  0,  0,  0,  0, 34,  1,  0,  1,  0],
       [ 2,  0,  0,  0,  0,  0, 28,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 35,  0,  0],
       [ 0,  2,  1,  1,  0,  1,  0,  1, 21,  1],
       [ 0,  0,  0,  0,  0,  2,  0,  0,  1, 38]], dtype=int64)
```

In [11]:
```
%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[11]:
```
Text(95.72222222222221, 0.5, 'Truth')
```