Installing Libraries

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_digits
```

Importing the Dataset

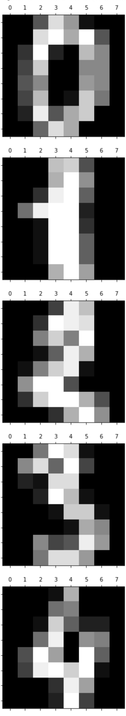
In [5]:
```
digits=load_digits()
dir(digits)
```

Out[5]:
```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

In [6]:
```
#shows the pixels of the image
digits.data[0]
```

Out[6]:
```
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.])
```

In [11]:
```
plt.gray()
for i in range(5):
    plt.matshow(digits.images[i])
```



In [13]:
```
digits.target[0:5]
```

Out[13]:
```
array([0, 1, 2, 3, 4])
```

In [5]:
```
len(digits.data)
```

Out[5]:
```
1797
```

In [7]:
```
#splitting the data into training and test set
xtrain,xtest,ytrain,ytest=train_test_split(digits.data,digits.target,test_
```

In [8]:
```
len(xtrain)
```

Out[8]:
```
1437
```

In [9]:
```
len(xtest)
```

Out[9]:
```
360
```

In [24]:
```
model=linear_model.LogisticRegression()
```

In [25]:
```
model.fit(xtrain,ytrain)
```

Out[25]:
```
C:\Users\Hp\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
LogisticRegression()
```

Checking the accuracy of the model

In [26]:
```
model.score(xtest,ytest)
```

Out[26]:
```
0.9722222222222222
```

In [28]:
```
plt.matshow(digits.images[95])
```

Out[28]:
```
<matplotlib.image.AxesImage at 0x18bd16daa30>
```

In [29]:
```
digits.target[95]
```

Out[29]:
```
6
```

In [30]:
```
model.predict([digits.data[95]])
```

Out[30]:
```
array([6])
```

In [32]:
```
model.predict(digits.data[0:5])
```

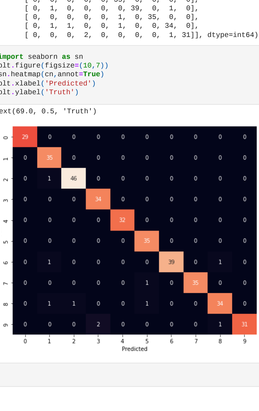
Out[32]:
```
array([0, 1, 2, 3, 4])
```

In [33]:
```
ypredicted=model.predict(xtest)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(ytest,ypredicted)
cm
```

Out[33]:
```
array([[29,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 35,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  1, 46,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 34,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0, 32,  0,  0,  0,  0,  0],
       [ 0,  0,  0,  0,  0, 35,  0,  0,  0,  0],
       [ 0,  1,  0,  0,  0,  0, 39,  0,  1,  0],
       [ 0,  0,  0,  0,  0,  1,  0, 35,  0,  0],
       [ 0,  1,  1,  0,  0,  1,  0,  0, 34,  0],
       [ 0,  0,  0,  2,  0,  0,  0,  0,  1, 31]], dtype=int64)
```

In [35]:
```
import seaborn as sn
plt.figure(figsize=(10,7))
sn.heatmap(cm,annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[35]:
```
Text(69.0, 0.5, 'Truth')
```



In [ ]:
```

```