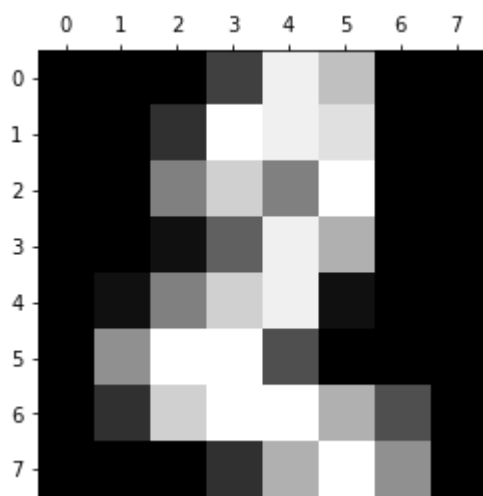
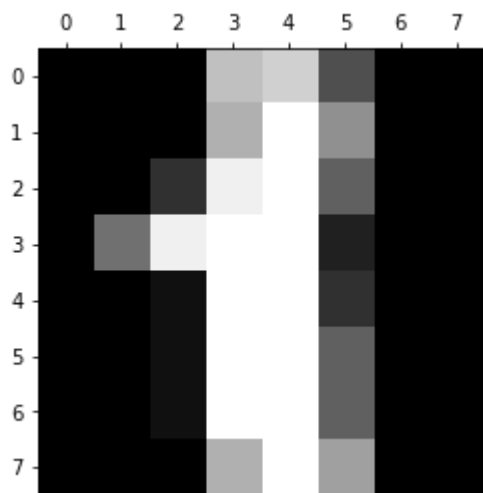
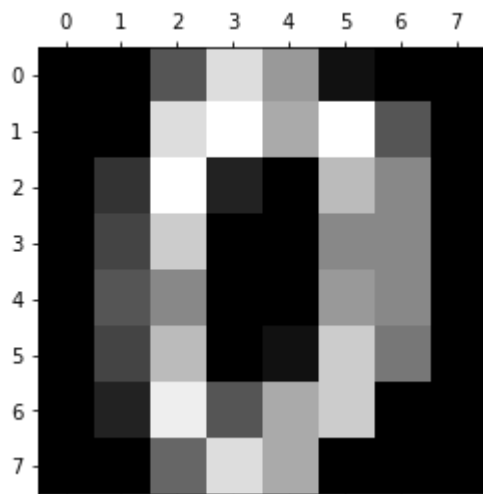
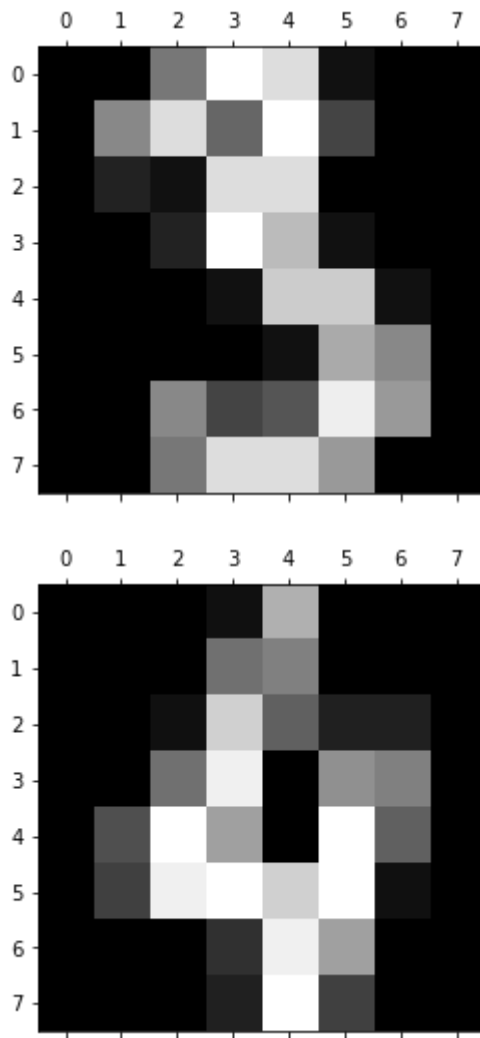


```
In [1]: from sklearn.datasets import load_digits
%matplotlib inline
import matplotlib.pyplot as plt
digits = load_digits()
```

```
In [2]: plt.gray()  
for i in range(5):  
    plt.matshow(digits.images[i])
```

<Figure size 432x288 with 0 Axes>





```
In [3]: dir(digits)
```

```
Out[3]: ['DESCR', 'data', 'images', 'target', 'target_names']
```

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

```
Out[4]: 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 [5]: from sklearn.linear_model import LogisticRegression
        model = LogisticRegression()
```

```
In [6]: from sklearn.model_selection import train_test_split
```

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target, test_size=0.2, random_state=42)
```

```
In [8]: model.fit(X_train, y_train)
```

```
C:\Users\aziz\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433:
FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a sol
ver to silence this warning.
```

```
FutureWarning)
```

```
C:\Users\aziz\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:460:
FutureWarning: Default multi_class will be changed to 'auto' in 0.22. Specify t
he multi_class option to silence this warning.
```

```
"this warning.", FutureWarning)
```

```
Out[8]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='warn',
n_jobs=None, penalty='l2', random_state=None, solver='warn',
tol=0.0001, verbose=0, warm_start=False)
```

```
In [9]: model.score(X_test, y_test)
```

```
Out[9]: 0.9666666666666667
```

```
In [10]: model.predict(digits.data[0:5])
```

```
Out[10]: array([0, 1, 2, 3, 4])
```

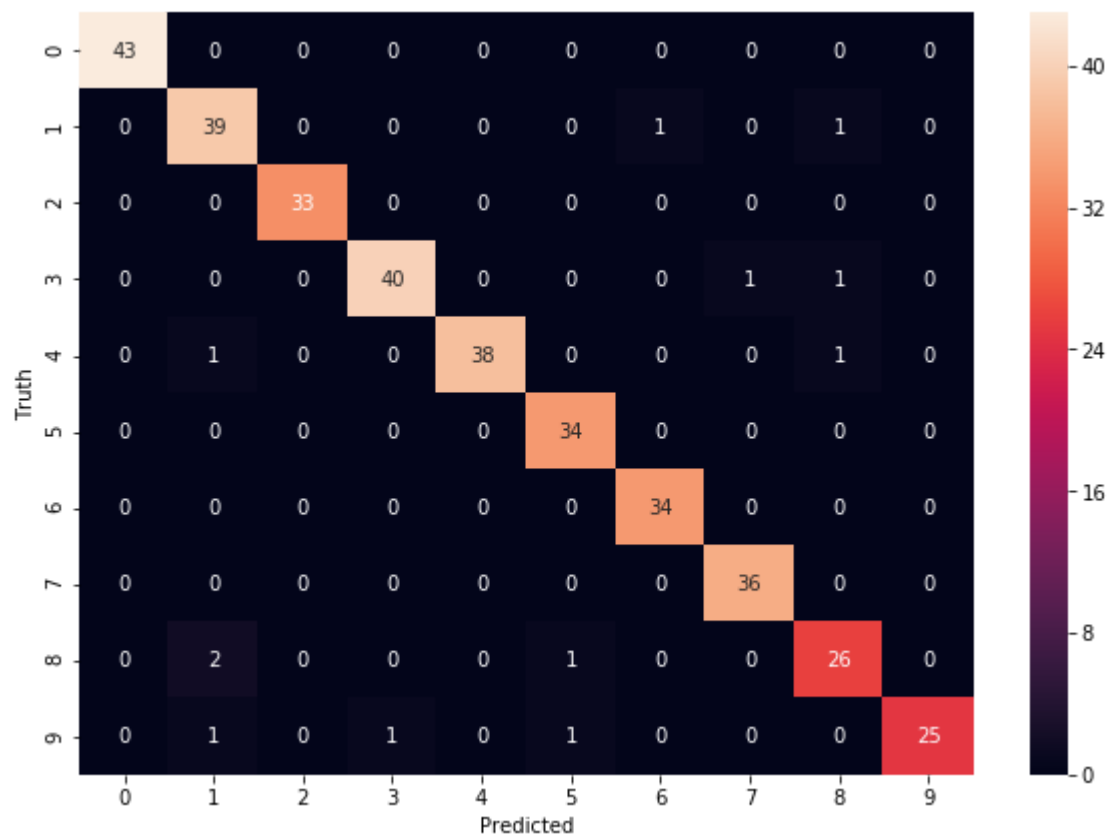
```
In [11]: y_predicted = model.predict(X_test)
```

```
In [12]: from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_predicted)
cm
```

```
Out[12]: array([[43,  0,  0,  0,  0,  0,  0,  0,  0,  0],
[ 0, 39,  0,  0,  0,  0,  1,  0,  1,  0],
[ 0,  0, 33,  0,  0,  0,  0,  0,  0,  0],
[ 0,  0,  0, 40,  0,  0,  0,  1,  1,  0],
[ 0,  1,  0,  0, 38,  0,  0,  0,  1,  0],
[ 0,  0,  0,  0,  0, 34,  0,  0,  0,  0],
[ 0,  0,  0,  0,  0,  0, 34,  0,  0,  0],
[ 0,  0,  0,  0,  0,  0,  0, 36,  0,  0],
[ 0,  2,  0,  0,  0,  1,  0,  0, 26,  0],
[ 0,  1,  0,  1,  0,  1,  0,  0,  0, 25]], dtype=int64)
```

```
In [15]: import seaborn as sn
plt.figure(figsize = (10,7))
sn.heatmap(cm, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

Out[15]: Text(69.0, 0.5, 'Truth')



In []: