# ml

October 8, 2023

```
[1]: %matplotlib inline
     import matplotlib.pyplot as plt
     from sklearn.datasets import load_digits
```

```
[2]: digits = load_digits()
```

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

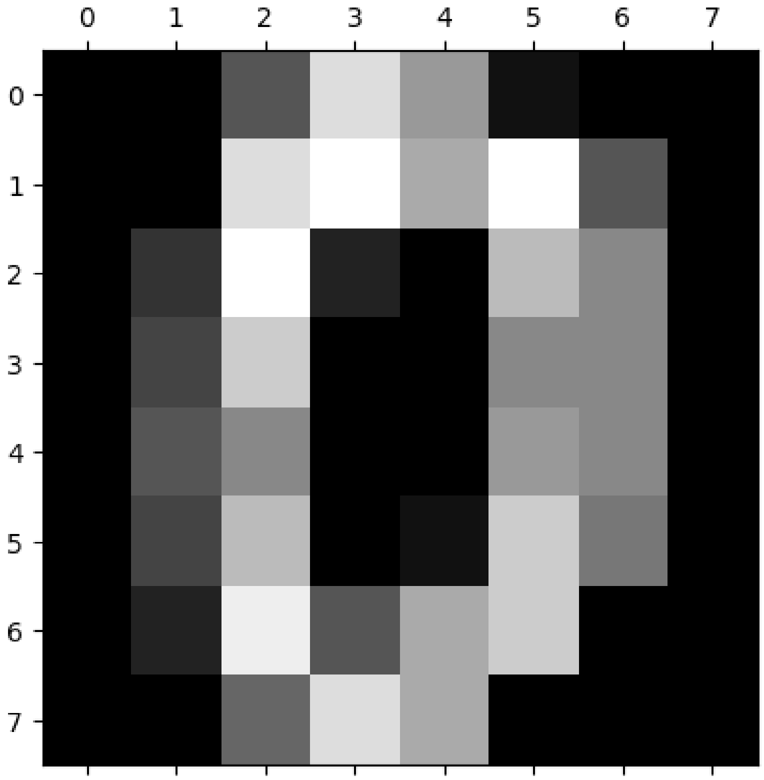
```
[3]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

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

```
[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.])
```

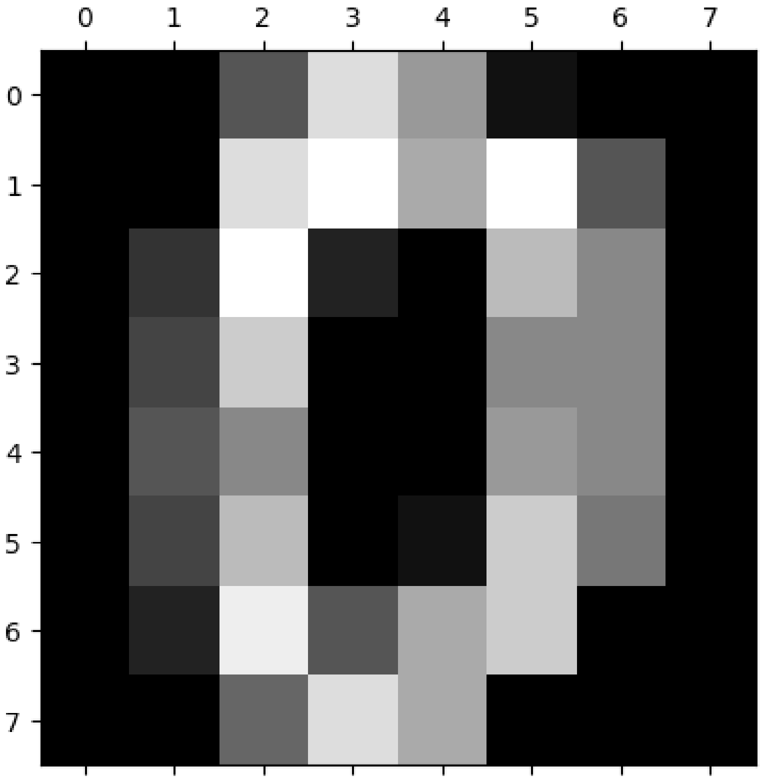
```
[5]: plt.gray()
     plt.matshow(digits.images[0])
```

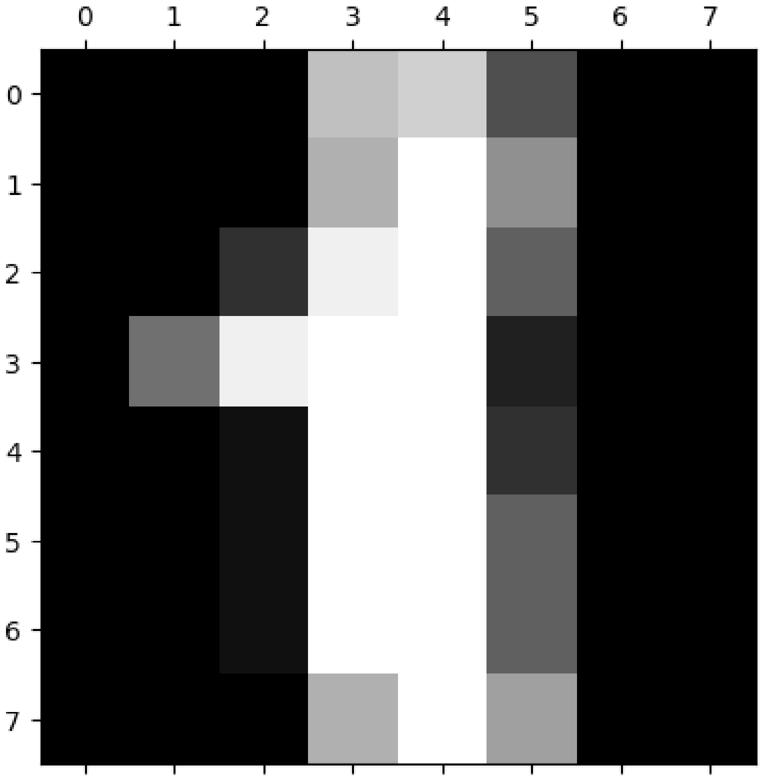
```
[5]: <matplotlib.image.AxesImage at 0x25ec9ccf100>

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

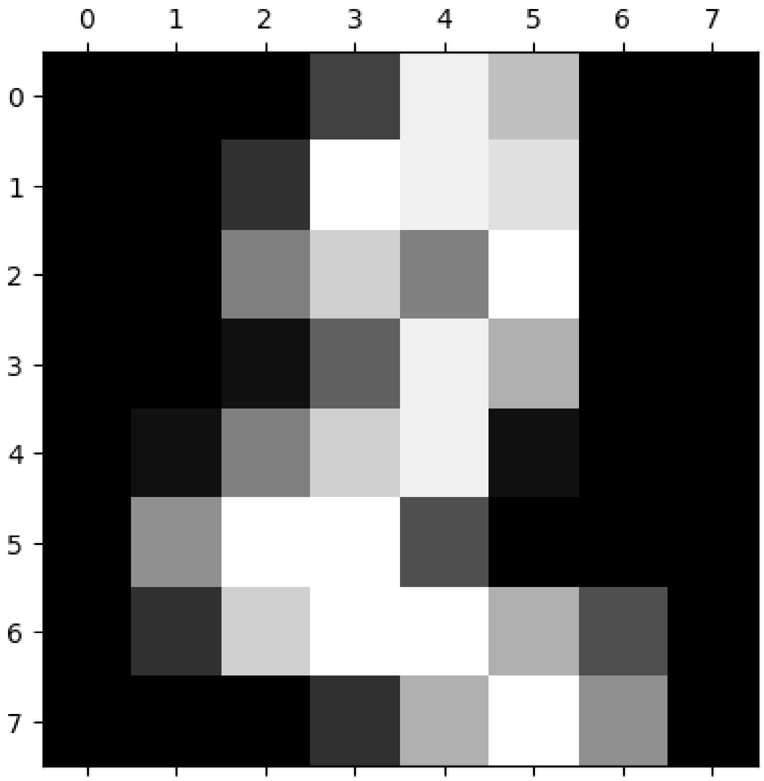


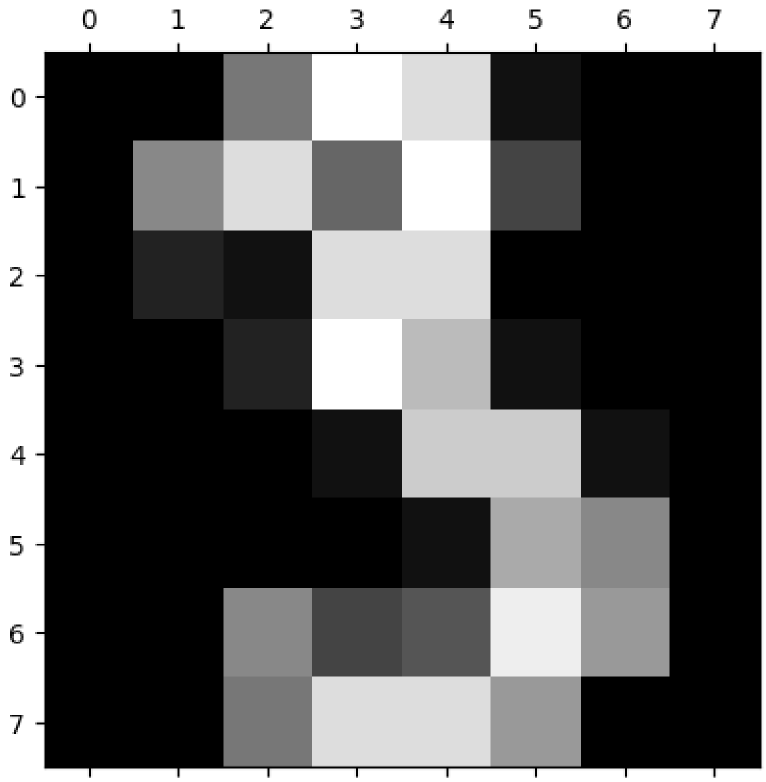
[8]:
```
plt.gray()
for i in range(5):
    plt.matshow(digits.images[i])
```

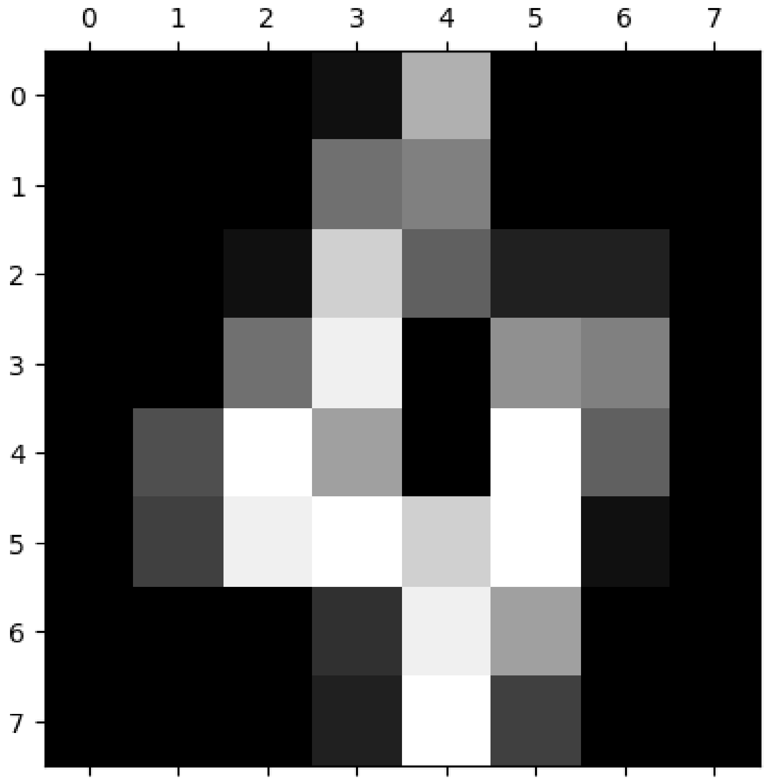
```
<Figure size 640x480 with 0 Axes>
```


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


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

```
[10]: digits.target[0:5]
```

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

```
[11]: from sklearn.model_selection import train_test_split
```

```
[12]: X_train,X_test,y_train,y_test = train_test_split(digits.data,digits.target)
```

```
[13]: len(X_train)
```

```
[13]: 1347
```

```
[14]: len(X_test)
```

```
[14]: 450
```

```
[31]: from sklearn.linear_model import LogisticRegression
      model=LogisticRegression()
```

```
[30]: Xt,Xv,yt,yv = train_test_split(X_train,y_train,test_size=0.25)
```

```
[33]: Xt.shape
```

```
[33]: (1010, 64)
```

```
[34]: Xv.shape
```

```
[34]: (337, 64)
```

```
[35]: model.fit(Xt,yt)
```

```
C:\Users\nandini sharma\anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:458: 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(
```

```
[35]: LogisticRegression()
```

[ ]:

[18]:

```
model.score(X_test,y_test)
```

[18]:

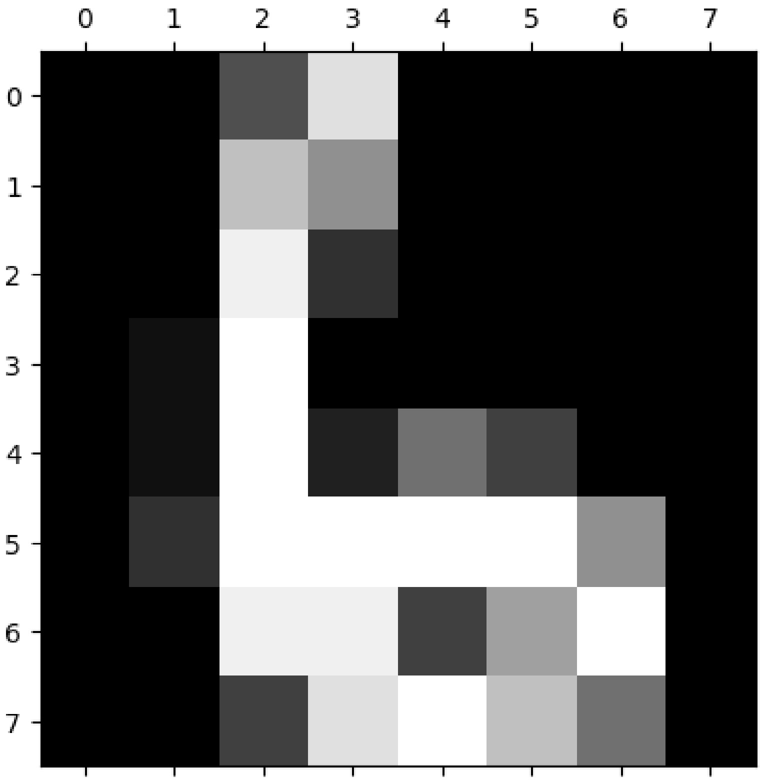
```
0.9488888888888889
```

[19]:

```
plt.matshow(digits.images[67])
```

[19]:

```
<matplotlib.image.AxesImage at 0x25ecb53e500>
```



[21]:

```
digits.target[67]
```

[21]:

```
6
```

[22]:

```
model.predict([digits.data[67]])
```

[22]:

```
array([6])
```

[23]:

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

[23]:

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

[24]:

```
y_predicted = model.predict(X_test)
from sklearn.metrics import confusion_matrix

cm= confusion_matrix(y_test,y_predicted)
cm
```

[24]:

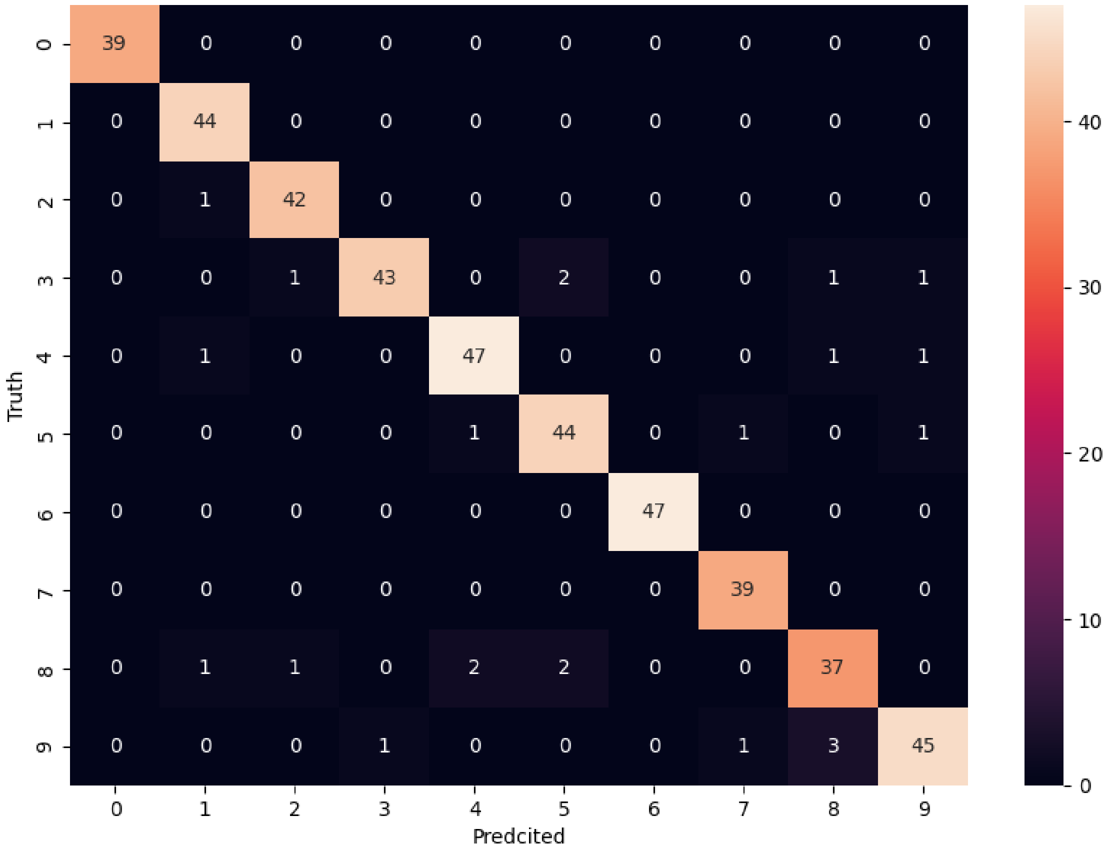
```
array([[39,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 44,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  1, 42,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  1, 43,  0,  2,  0,  0,  1,  1],
       [ 0,  1,  0,  0, 47,  0,  0,  0,  1,  1],
       [ 0,  0,  0,  0,  1, 44,  0,  1,  0,  1],
       [ 0,  0,  0,  0,  0,  0, 47,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 39,  0,  0],
       [ 0,  1,  1,  0,  2,  2,  0,  0, 37,  0],
       [ 0,  0,  0,  1,  0,  0,  0,  1,  3, 45]], dtype=int64)
```

[26]:

```
import seaborn as sns
plt.figure(figsize = (10,7))
sns.heatmap(cm,annot=True)
plt.xlabel('Predcited')
plt.ylabel('Truth')
```

[26]:

```
Text(95.72222222222221, 0.5, 'Truth')
```


0 39 0 0 0 0 0 0 0 0 0
1 0 44 0 0 0 0 0 0 0 0
2 0 1 42 0 0 0 0 0 0 0
3 0 0 1 43 0 2 0 0 1 1
4 0 1 0 0 47 0 0 0 1 1
5 0 0 0 0 1 44 0 1 0 1
6 0 0 0 0 0 0 47 0 0 0
7 0 0 0 0 0 0 0 39 0 0
8 0 1 1 0 2 2 0 0 37 0
9 0 0 0 1 0 0 0 1 3 45
Truth
0 1 2 3 4 5 6 7 8 9
Predcited
40
30
20
10
0

[ ]: