

Log Reg - Handwritten Digits prediction

December 18, 2022

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

from sklearn.datasets import load_digits

[2]: digits = load_digits()

[3]: dir(digits)

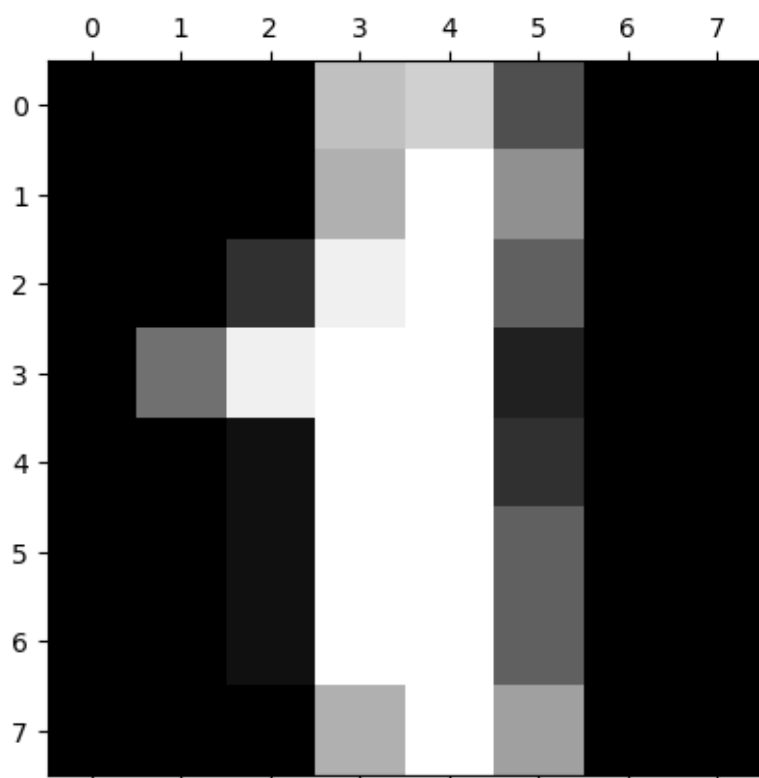
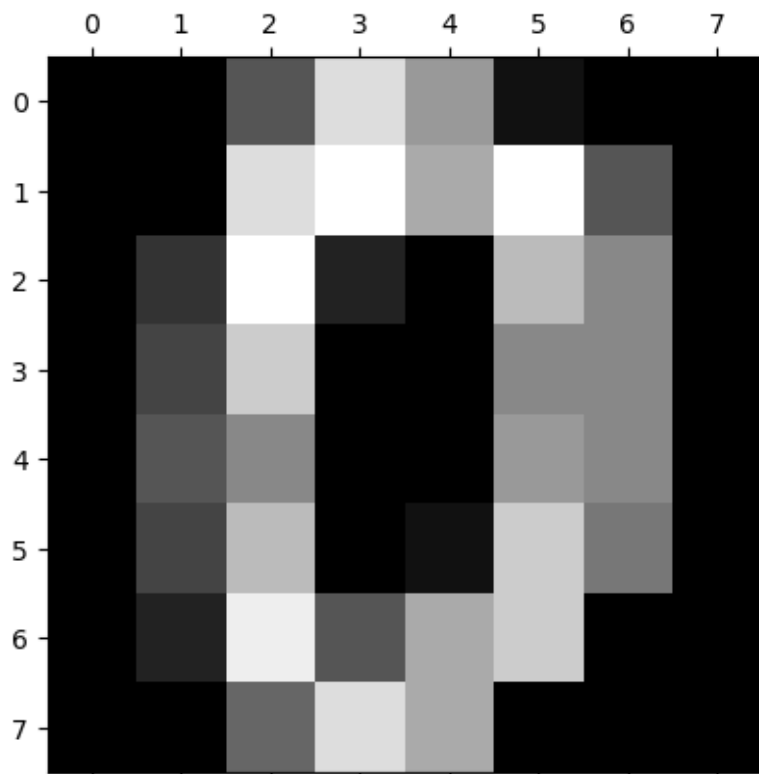
[3]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']

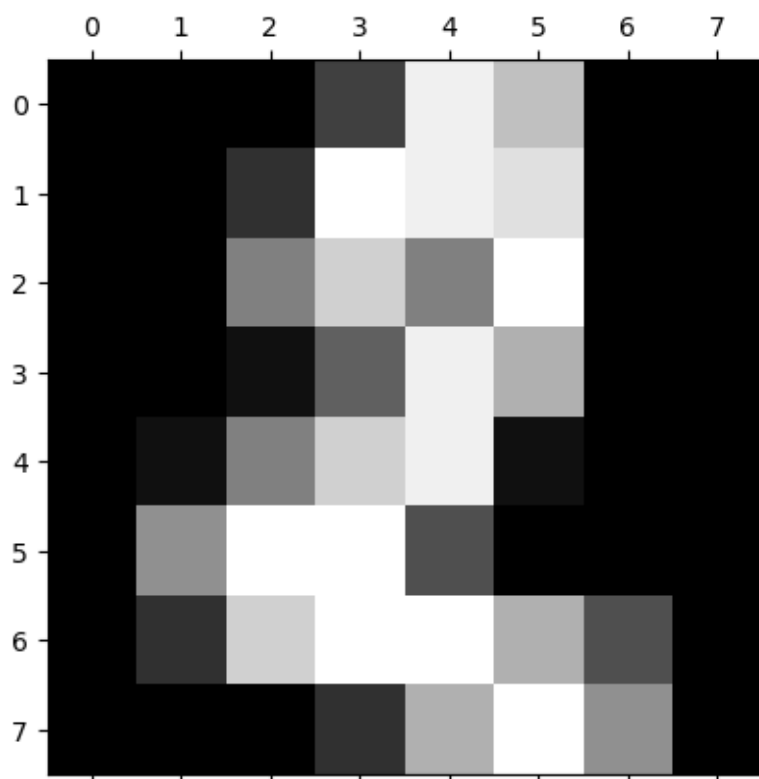
[4]: digits.data[0]

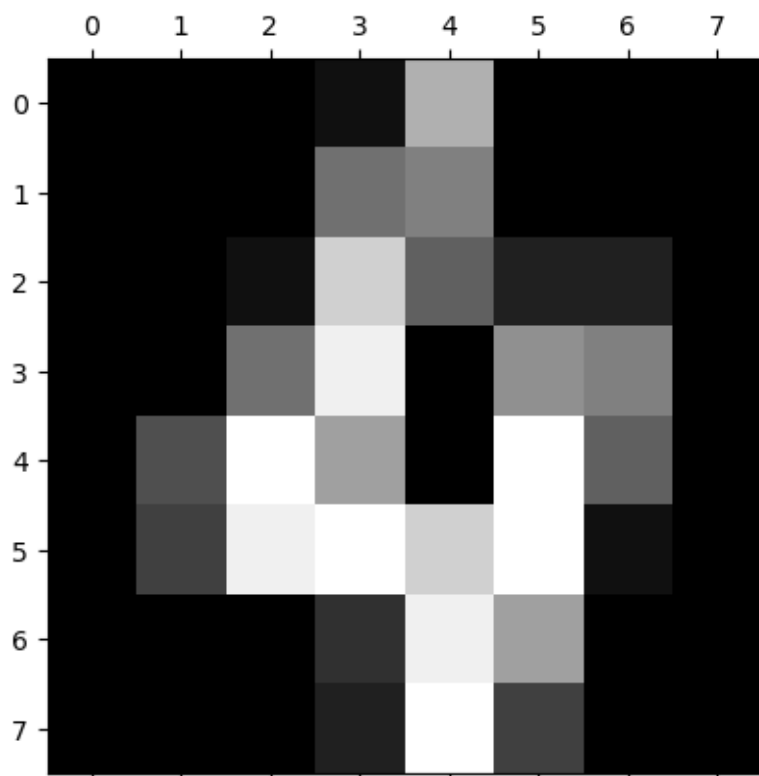
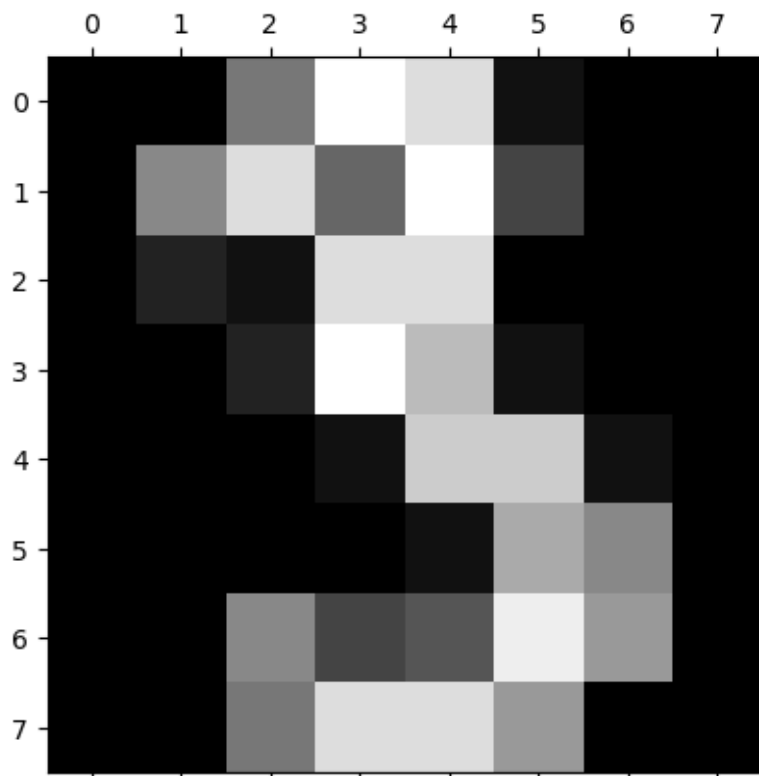
[4]: array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
          15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
          12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
           0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
          10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])

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

<Figure size 640x480 with 0 Axes>







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

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

[7]: from sklearn.model_selection import train_test_split

[8]: x_train,x_test,y_train,y_test = train_test_split(digits.data,digits.
↳target,test_size=0.2)

[9]: len(x_train)

[9]: 1437

[10]: len(x_test)

[10]: 360

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

[12]: model.fit(x_train,y_train)

C:\Users\Deepak\ana-conda-3\lib\site-
packages\sklearn\linear_model\_logistic.py:814: 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(

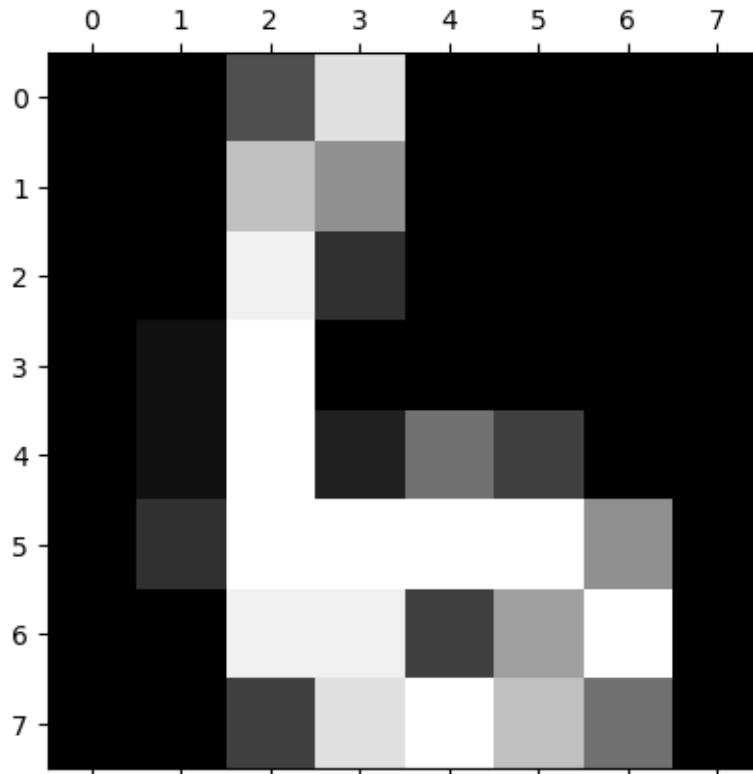
[12]: LogisticRegression()

[13]: model.score(x_test,y_test)

[13]: 0.9777777777777777

[14]: plt.matshow(digits.images[67])

[14]: <matplotlib.image.AxesImage at 0x1fa1d11abb0>
```



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

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

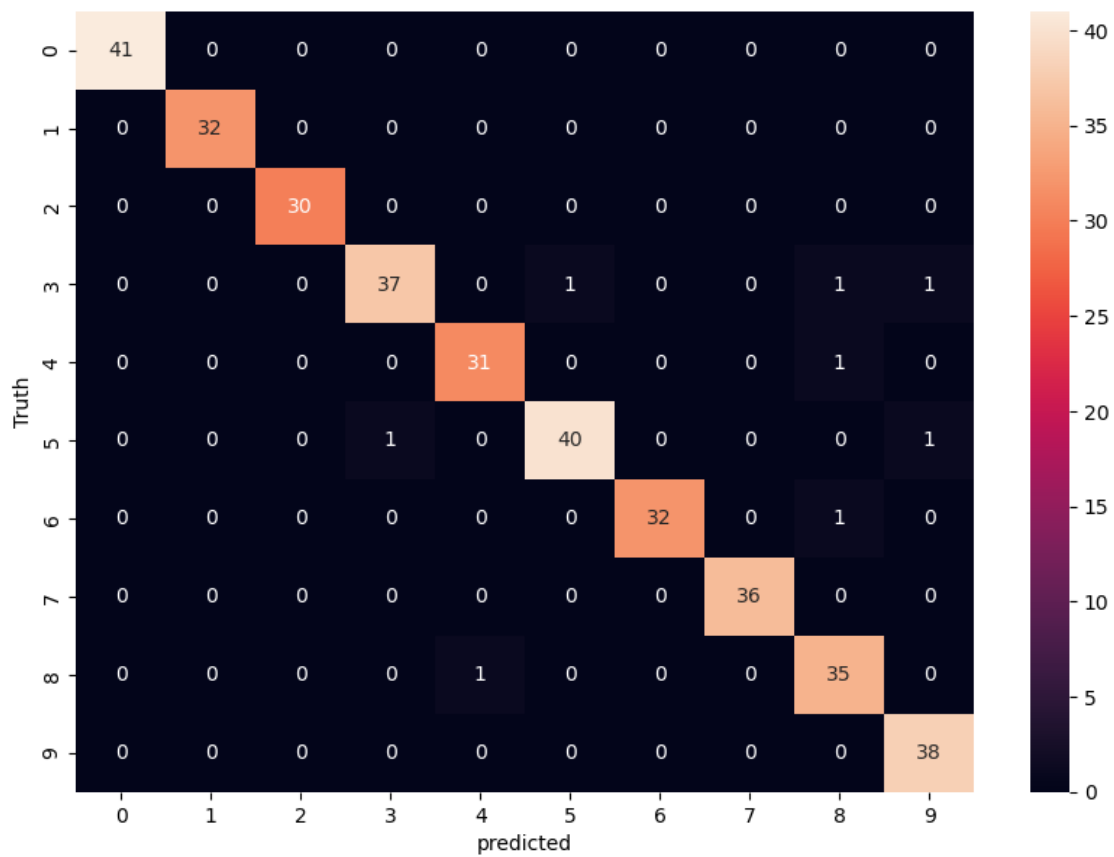
```
[16]: y_predicted = model.predict(x_test)
from sklearn.metrics import confusion_matrix

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

```
[16]: array([[41,  0,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 32,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0, 30,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0, 37,  0,  1,  0,  0,  1,  1],
 [ 0,  0,  0,  0, 31,  0,  0,  0,  1,  0],
 [ 0,  0,  0,  1,  0, 40,  0,  0,  0,  1],
 [ 0,  0,  0,  0,  0,  0, 32,  0,  1,  0],
 [ 0,  0,  0,  0,  0,  0,  0, 36,  0,  0],
 [ 0,  0,  0,  0,  1,  0,  0,  0, 35,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 38]], dtype=int64)
```

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

```
[17]: Text(95.72222222222221, 0.5, 'Truth')
```



```
[ ]:
```