

logistic regression multiple classification

May 11, 2023

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

```
[2]: from sklearn.datasets import load_digits
```

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

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

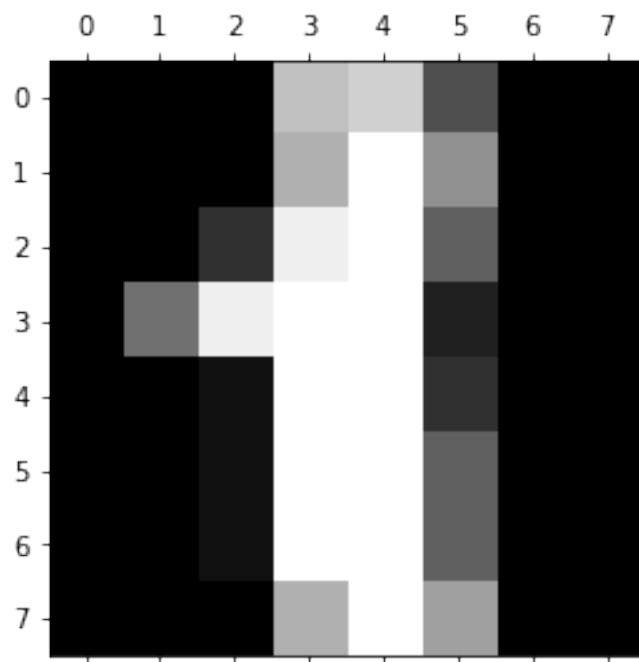
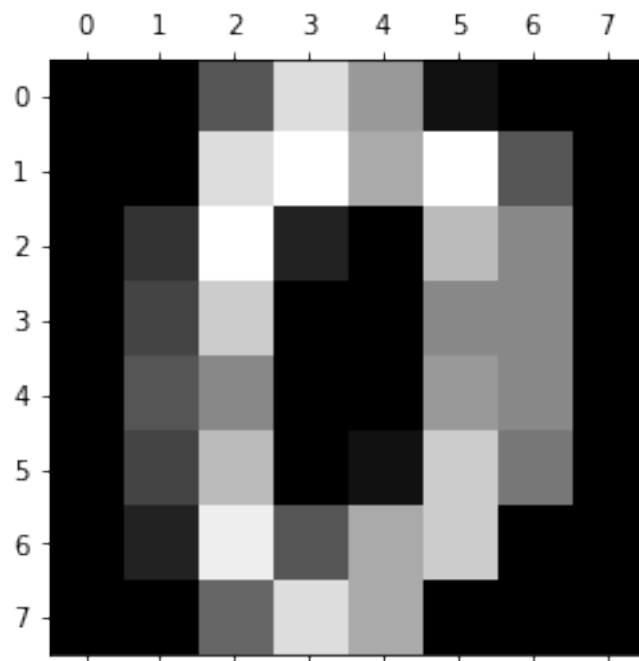
```
[4]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

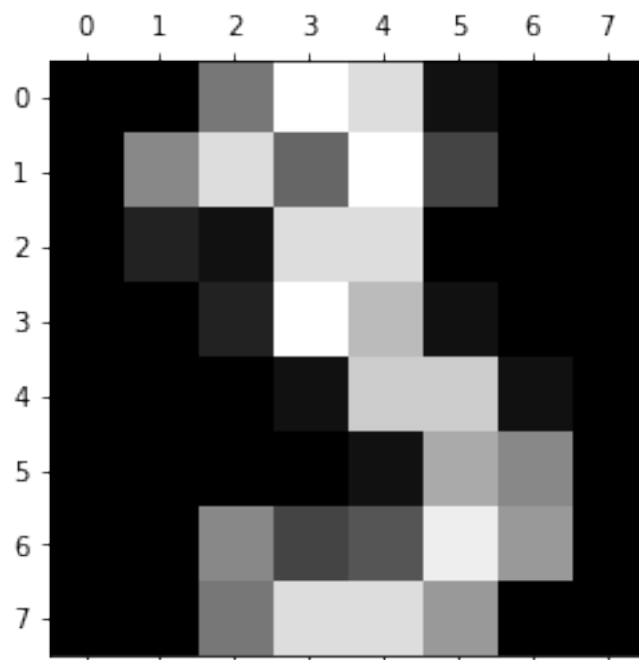
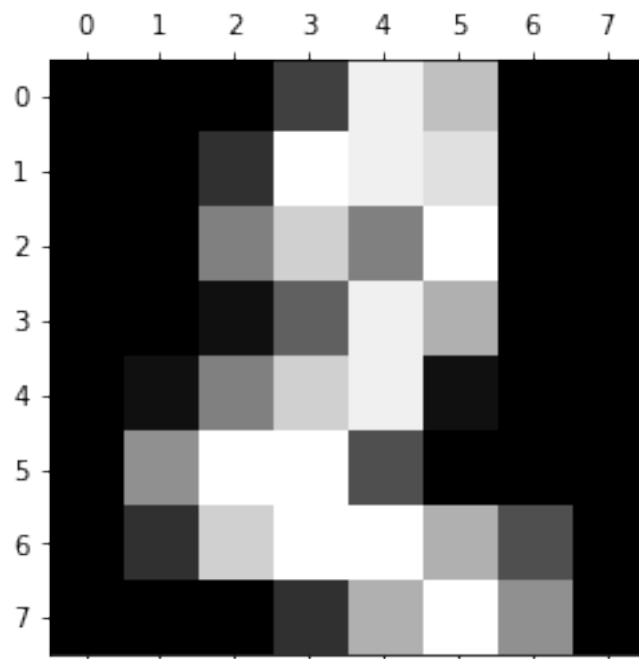
```
[5]: digits.data[0]
```

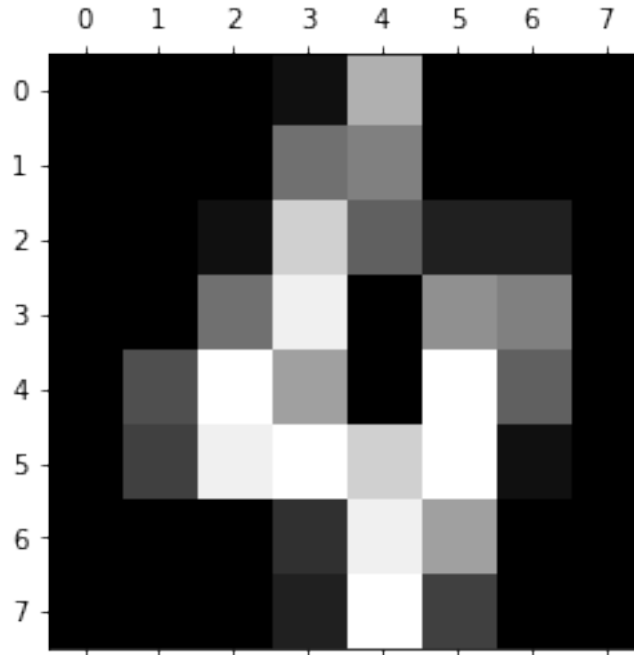
```
[5]: array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,  
          15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,  
          12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,  
           0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2.,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

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

<Figure size 432x288 with 0 Axes>







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

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

```
[13]: from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(digits.data,digits.
↪target,test_size=0.2)
```

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

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

C:\Users\Rakesh\anaconda3\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(
```

```
[19]: model.score(X_train,y_train)
```

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

```
[41]: digits.target[67]
```

```
[24]: digits.data[67]
```

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

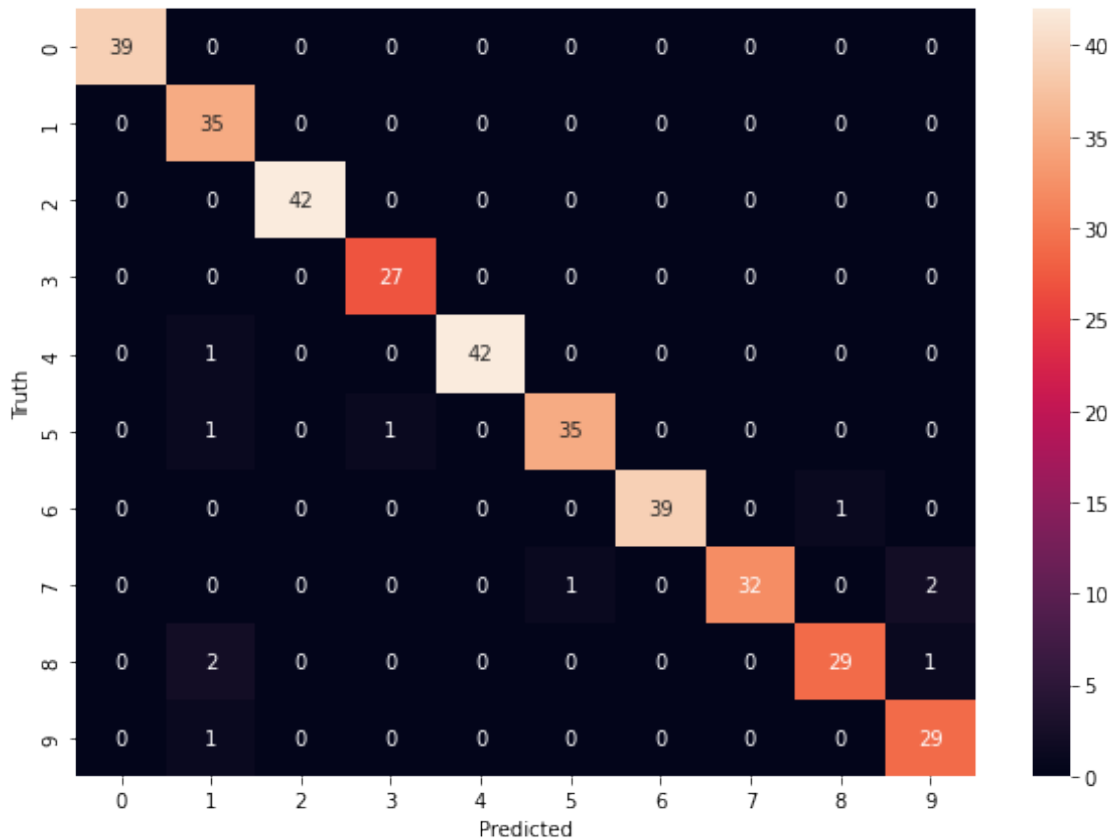
A 10x10 grayscale heatmap representing a handwritten digit '4'. The x and y axes are both labeled from 0 to 7. The digit is formed by a collection of gray and white pixels against a black background. The shape is a standard '4' with a vertical stroke on the left, a horizontal stroke in the middle, and a diagonal stroke on the right.

```
[34]: y_predicted=model.predict(X_test)
      from sklearn.metrics import confusion_matrix
      cm=confusion_matrix(y_test,y_predicted)
      cm
```

```
[34]: array([[39,  0,  0,  0,  0,  0,  0,  0,  0,  0],
            [ 0, 35,  0,  0,  0,  0,  0,  0,  0,  0],
            [ 0,  0, 42,  0,  0,  0,  0,  0,  0,  0],
            [ 0,  0,  0, 27,  0,  0,  0,  0,  0,  0],
            [ 0,  1,  0,  0, 42,  0,  0,  0,  0,  0],
            [ 0,  1,  0,  1,  0, 35,  0,  0,  0,  0],
            [ 0,  0,  0,  0,  0,  0, 39,  0,  1,  0],
            [ 0,  0,  0,  0,  0,  1,  0, 32,  0,  2],
            [ 0,  2,  0,  0,  0,  0,  0,  0, 29,  1],
            [ 0,  1,  0,  0,  0,  0,  0,  0,  0, 29]], dtype=int64)
```

```
[38]: import seaborn as sns
      plt.figure(figsize=(10,7))
      sns.heatmap(cm,annot=True)
      plt.xlabel("Predicted")
      plt.ylabel("Truth")
```

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



[]: