

Logistic

June 27, 2025

```
[16]: import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.datasets import load_digits
import matplotlib.pyplot as plt
```

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

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

```
[14]: digits.data[0]
digits.target[9]
```

```
[14]: 9
```

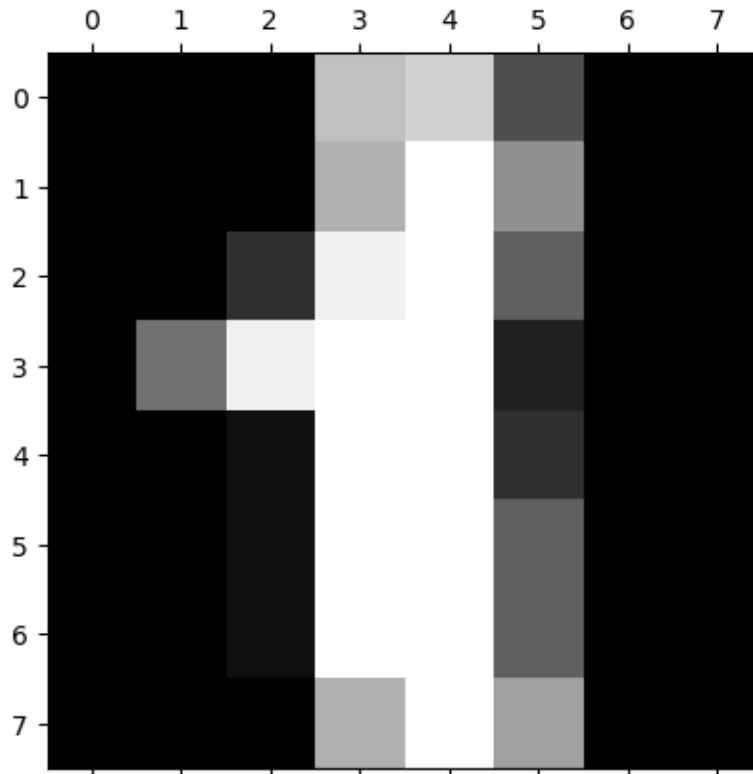
```
[15]: digits.images[0]
```

```
[15]: 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.]])
```

```
[18]: plt.gray()
plt.matshow(digits.images[1])
```

```
[18]: <matplotlib.image.AxesImage at 0x73d808b7b310>
```

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



```
[37]: X=digits['data']
      y=digits['target']
```

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

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

```
[38]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
      len(y_train)
```

```
[38]: 1437
```

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

```
/opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-
packages/sklearn/linear_model/_logistic.py:460: 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(
```

```
[39]: LogisticRegression()
```

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

```
[40]: 0.9694444444444444
```

```
[42]: model.predict([digits.data[54]])
```

```
[42]: array([2])
```

```
[43]: from sklearn.metrics import confusion_matrix
```

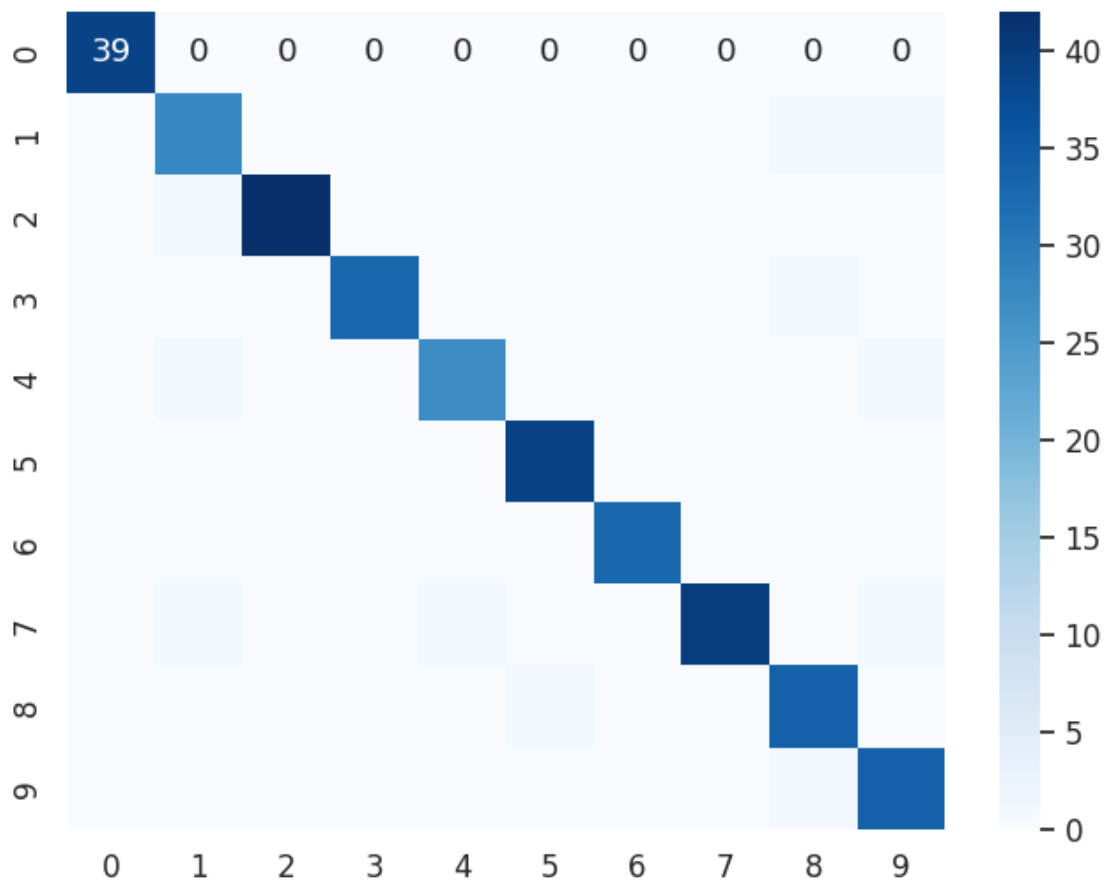
```
[51]: y_predict=model.predict(X_test)
```

```
[52]: cm=confusion_matrix(y_test,y_predict)
cm
```

```
[52]: array([[39,  0,  0,  0,  0,  0,  0,  0,  0,  0],
        [ 0, 28,  0,  0,  0,  0,  0,  0,  1,  1],
        [ 0,  1, 42,  0,  0,  0,  0,  0,  0,  0],
        [ 0,  0,  0, 33,  0,  0,  0,  0,  1,  0],
        [ 0,  1,  0,  0, 27,  0,  0,  0,  0,  1],
        [ 0,  0,  0,  0,  0, 39,  0,  0,  0,  0],
        [ 0,  0,  0,  0,  0,  0, 33,  0,  0,  0],
        [ 0,  1,  0,  0,  1,  0,  0, 40,  0,  1],
        [ 0,  0,  0,  0,  0,  1,  0,  0, 34,  0],
        [ 0,  0,  0,  0,  0,  0,  0,  0,  1, 34]])
```

```
[64]: plt.figure(figsize=(8,6),dpi=100)
sns.set(font_scale=1.1)
sns.heatmap(cm,annot=True,fmt='g',cmap='Blues')
```

```
[64]: <Axes: >
```



[]: