

In the previous Logistic Regression 1, we discuss and saw how to work on a Binary classification where output is either 0 or 1

In this one, we will see MultiClass Classification, for example which party a person is going to vote for? Democratic, Republican or Independent.

The problem we are going to solve today is to Recognize the Hand written digits from 0 to 9.

We will use a dataset for digits which is already present in SKLearn, then try to identify it using Logical Regression.

This dataset contains approx 1797 8x8 images of digits from 0 to 9, and also is numeric data

```
In [39]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_digits
```

```
In [40]: digits = load_digits()
```

Lets first explore what does this dataset contains

```
In [41]: dir(digits)
```

```
Out[41]: ['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

As you can see, it contains a 'data' from probably the numeric value of the image is present and the 'images' where the actual image of the digits are, also 'target' where it probably has the output or the actual identification of it

Printing the first value at index 0 in 'data', as you can see its an array, the numeric value of the actual images cuz we know our model can only understand numbers

```
In [42]: digits.data[0]
```

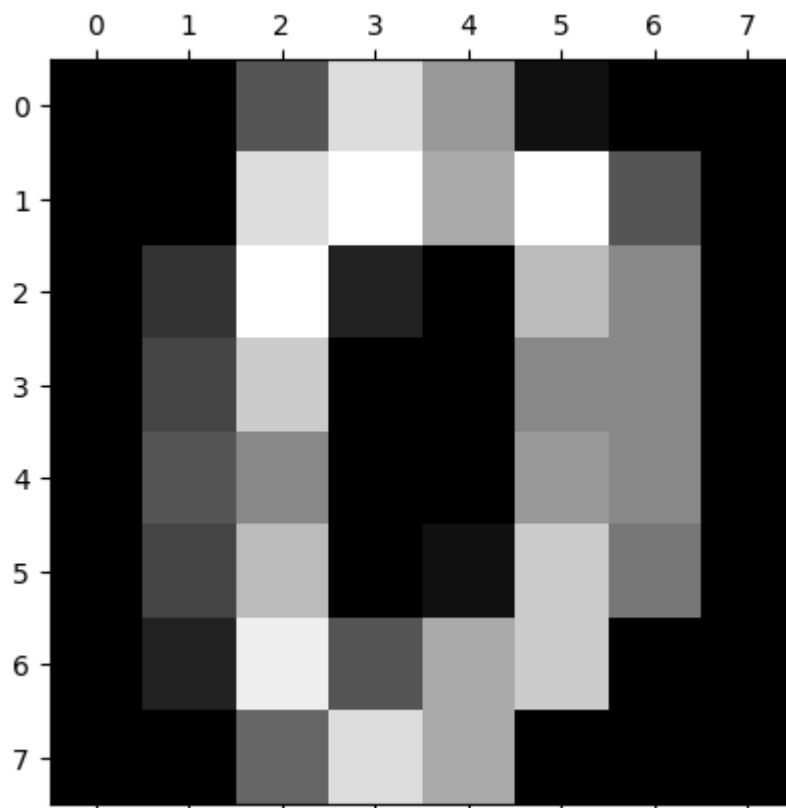
```
Out[42]: array([ 0.,  0.,  5., 13.,  9.,  1.,  0.,  0.,  0.,  0., 13., 15., 10.,
                15.,  5.,  0.,  0.,  3., 15.,  2.,  0., 11.,  8.,  0.,  0.,  4.,
                12.,  0.,  0.,  8.,  8.,  0.,  0.,  5.,  8.,  0.,  0.,  9.,  8.,
                 0.,  0.,  4., 11.,  0.,  1., 12.,  7.,  0.,  0.,  2., 14.,  5.,
                10., 12.,  0.,  0.,  0.,  0.,  6., 13., 10.,  0.,  0.,  0.])
```

Now, if you want to see this value's actual image, you can use matplotlib as follows in index 0

```
In [43]: plt.gray()
plt.matshow(digits.images[0])
```

```
Out[43]: <matplotlib.image.AxesImage at 0x25e7ed4a010>
```

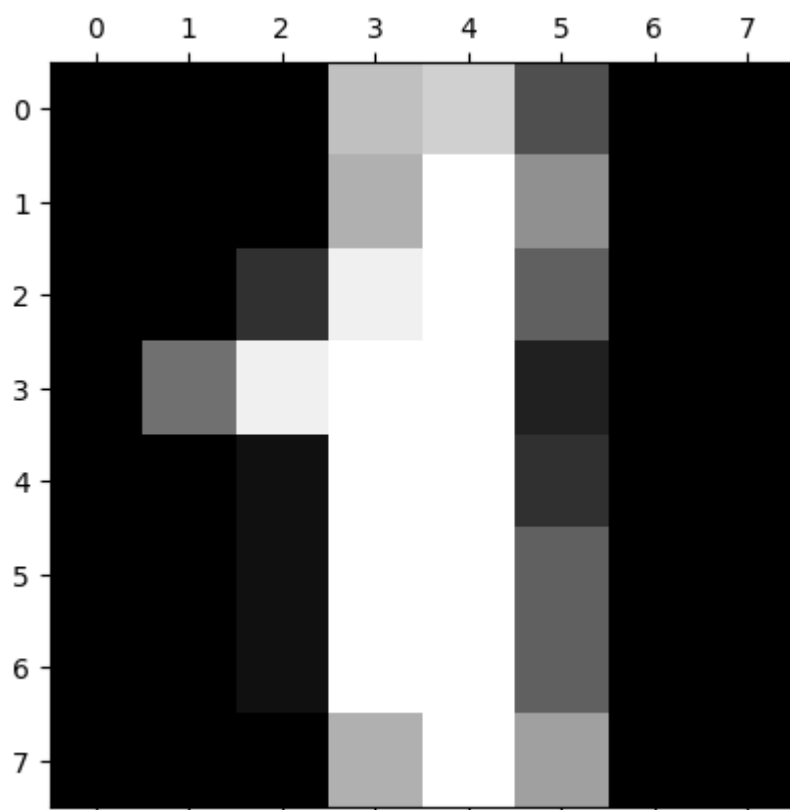
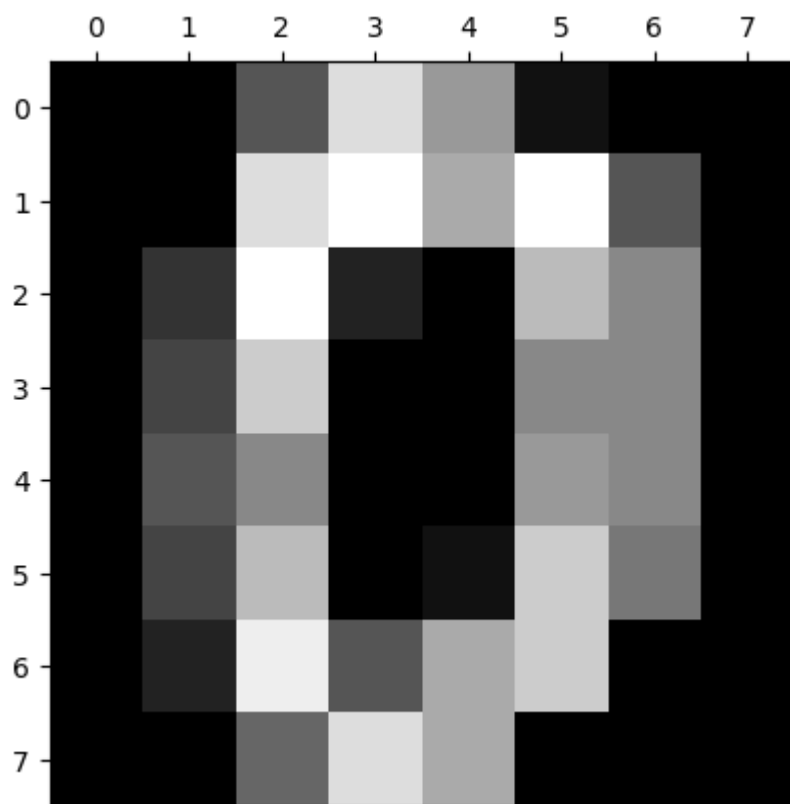
<Figure size 640x480 with 0 Axes>

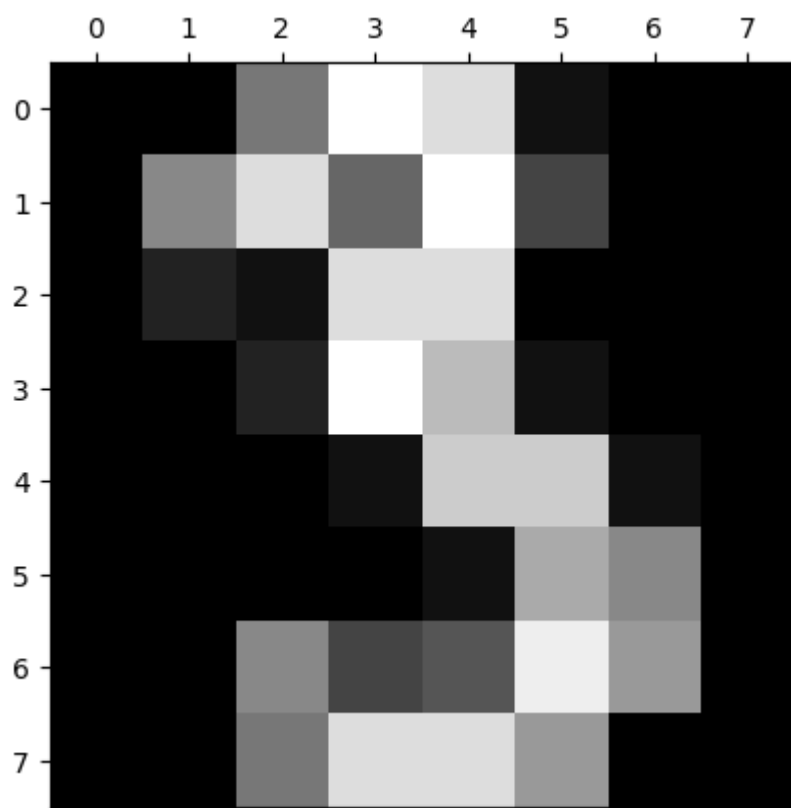
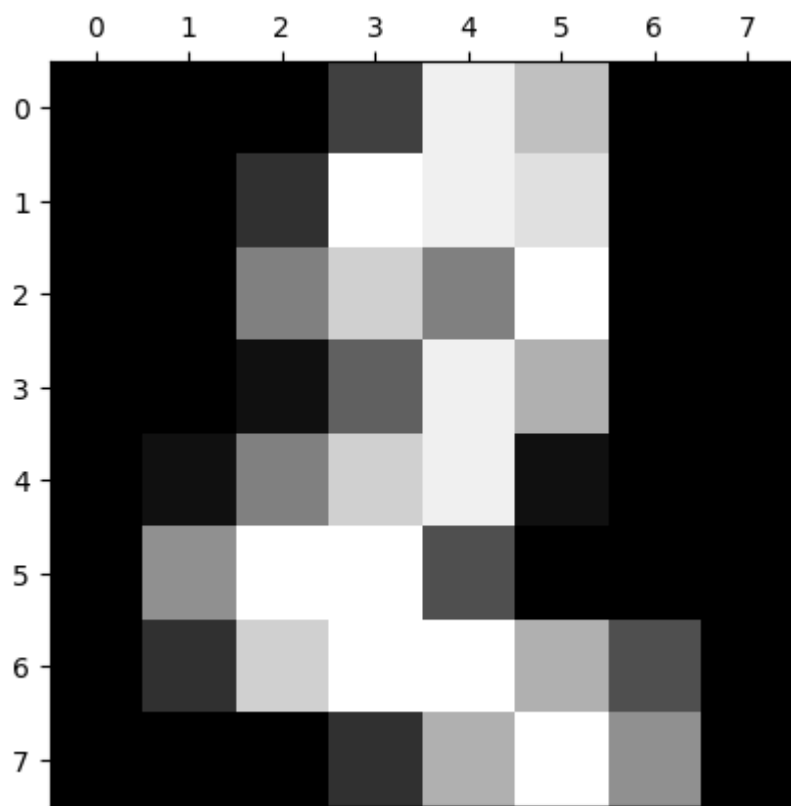


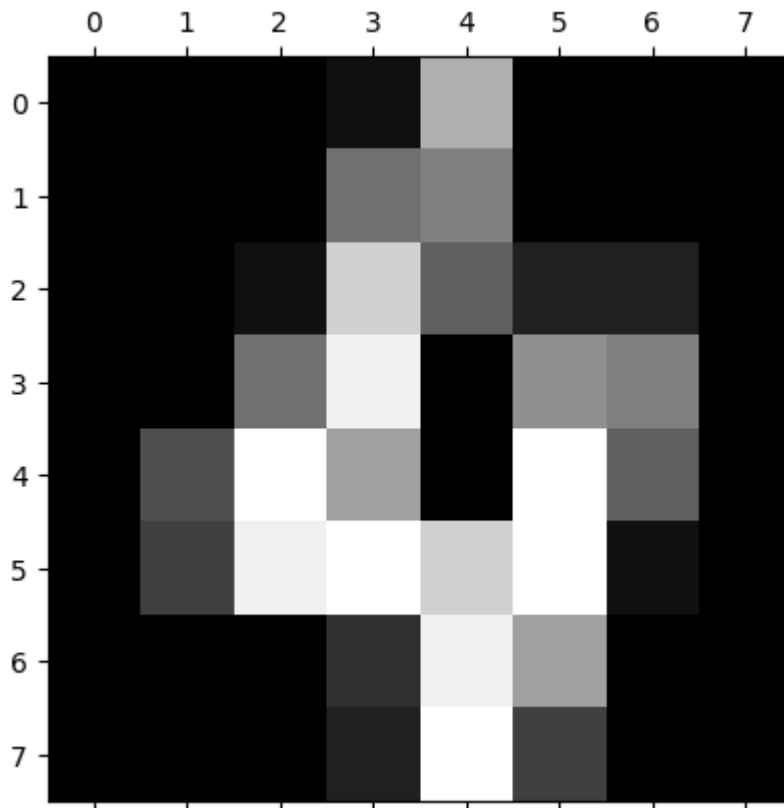
If you want to display lets say the first five sample, you can do this as well

```
In [44]: plt.gray()
for i in range(5):
    plt.matshow(digits.images[i])
```

<Figure size 640x480 with 0 Axes>







Now, before we start training our model, let's look at the corresponding 'target' of these images from 0 to 5

```
In [45]: digits.target[0:5]
```

```
Out[45]: array([0, 1, 2, 3, 4])
```

## Splitting the datasets to Train and Test

```
In [46]: from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(digits.data, digits.target)
```

```
In [47]: len(X_train)
```

```
Out[47]: 1437
```

```
In [48]: len(X_test)
```

```
Out[48]: 360
```

## Model Training and Prediction

```
In [49]: model = LogisticRegression()
```

```
In [50]: model.fit(X_train, y_train)
```

```
C:\Users\User\AppData\Local\Programs\Python\Python311\Lib\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](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter_i = _check_optimize_result(
```

```
Out[50]: ▼ LogisticRegression
LogisticRegression()
```

We have done training, Before predicting, lets calculate the score to see how accurate our model is

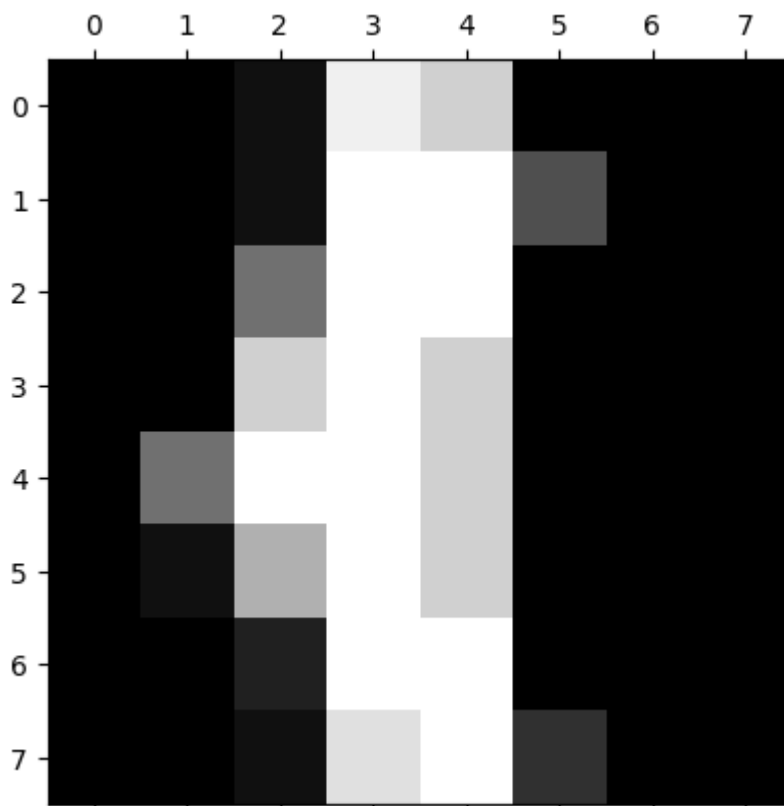
```
In [51]: model.score(X_test, y_test)
```

```
Out[51]: 0.975
```

Now, lets make an actual prediction, but first with a particular random value from the data

```
In [52]: plt.matshow(digits.images[99])
```

```
Out[52]: <matplotlib.image.AxesImage at 0x25e7f4cce10>
```



```
In [53]: digits.target[99]
```

```
Out[53]: 1
```

```
In [54]: model.predict([digits.data[99]])
```

```
Out[54]: array([1])
```

As you can see it guessed the digit at index 99 correctly, now lets Predict a digits from 0 to 5

```
In [59]: model.predict(digits.data[0:5])
```

```
Out[59]: array([0, 1, 2, 3, 4])
```

Now, lets use our actual Test data

```
In [60]: predict = model.predict(X_test)
predict
```

```
Out[60]: array([3, 9, 7, 6, 0, 3, 7, 5, 5, 1, 6, 8, 4, 7, 0, 0, 7, 4, 2, 0, 7, 4,
                3, 0, 2, 1, 8, 3, 4, 1, 9, 4, 6, 9, 2, 4, 0, 0, 0, 9, 6, 4, 0, 0,
                6, 7, 3, 9, 7, 8, 9, 2, 3, 1, 4, 4, 7, 7, 9, 6, 7, 1, 2, 5, 0, 3,
                6, 9, 9, 4, 1, 0, 5, 7, 2, 0, 8, 6, 2, 1, 8, 1, 1, 8, 0, 2, 7, 0,
                1, 7, 2, 3, 6, 0, 0, 8, 9, 3, 8, 6, 9, 7, 9, 1, 9, 6, 3, 8, 6, 6,
                4, 9, 8, 0, 0, 3, 2, 4, 1, 7, 2, 2, 7, 0, 5, 3, 6, 0, 8, 4, 9, 9,
                2, 1, 6, 5, 4, 1, 7, 5, 3, 2, 4, 2, 1, 1, 7, 7, 6, 4, 7, 7, 7, 7,
                1, 8, 1, 1, 3, 5, 8, 2, 0, 0, 6, 2, 6, 4, 6, 5, 5, 2, 3, 2, 0, 2,
                4, 8, 7, 4, 5, 1, 4, 3, 4, 3, 8, 6, 3, 3, 4, 5, 3, 3, 0, 6, 7, 2,
                4, 7, 4, 7, 1, 2, 6, 8, 8, 2, 2, 7, 4, 9, 5, 0, 4, 1, 5, 4, 0, 4,
                3, 4, 1, 2, 1, 4, 5, 0, 1, 9, 2, 1, 2, 5, 2, 0, 5, 3, 3, 3, 4, 8,
                7, 1, 3, 9, 6, 9, 6, 5, 3, 9, 8, 6, 8, 5, 2, 1, 7, 7, 5, 2, 5, 5,
                1, 6, 9, 9, 5, 1, 4, 6, 1, 7, 0, 7, 3, 7, 9, 7, 8, 0, 9, 9, 3, 9,
                0, 2, 9, 9, 6, 6, 7, 2, 3, 1, 9, 4, 2, 8, 2, 3, 4, 9, 2, 5, 7, 0,
                9, 1, 0, 5, 9, 6, 4, 5, 7, 7, 3, 5, 6, 5, 8, 0, 0, 0, 8, 1, 8, 8,
                8, 4, 7, 4, 9, 6, 8, 6, 3, 9, 6, 3, 9, 5, 7, 0, 5, 9, 7, 7, 8, 0,
                6, 7, 0, 1, 7, 9, 5, 2])
```

Checking Actual and Predicted value

```
In [61]: for i in range(len(predict)):
          print(f"Actual {y_test[i]}, Predict {predict[i]}")
```

Actual 3, Predict 3  
Actual 9, Predict 9  
Actual 7, Predict 7  
Actual 6, Predict 6  
Actual 0, Predict 0  
Actual 3, Predict 3  
Actual 7, Predict 7  
Actual 5, Predict 5  
Actual 5, Predict 5  
Actual 1, Predict 1  
Actual 6, Predict 6  
Actual 8, Predict 8  
Actual 4, Predict 4  
Actual 7, Predict 7  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 7, Predict 7  
Actual 4, Predict 4  
Actual 2, Predict 2  
Actual 0, Predict 0  
Actual 7, Predict 7  
Actual 4, Predict 4  
Actual 3, Predict 3  
Actual 0, Predict 0  
Actual 2, Predict 2  
Actual 1, Predict 1  
Actual 8, Predict 8  
Actual 3, Predict 3  
Actual 4, Predict 4  
Actual 1, Predict 1  
Actual 9, Predict 9  
Actual 4, Predict 4  
Actual 5, Predict 6  
Actual 9, Predict 9  
Actual 2, Predict 2  
Actual 4, Predict 4  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 4, Predict 4  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 6, Predict 6  
Actual 7, Predict 7  
Actual 3, Predict 3  
Actual 9, Predict 9  
Actual 7, Predict 7  
Actual 8, Predict 8  
Actual 5, Predict 9  
Actual 2, Predict 2  
Actual 3, Predict 3  
Actual 1, Predict 1  
Actual 4, Predict 4



Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 7, Predict 7  
Actual 1, Predict 1  
Actual 8, Predict 2  
Actual 5, Predict 5  
Actual 0, Predict 0  
Actual 3, Predict 3  
Actual 6, Predict 6  
Actual 5, Predict 9  
Actual 9, Predict 9  
Actual 4, Predict 4  
Actual 1, Predict 1  
Actual 0, Predict 0  
Actual 5, Predict 5  
Actual 7, Predict 7  
Actual 2, Predict 2  
Actual 0, Predict 0  
Actual 8, Predict 8  
Actual 6, Predict 6  
Actual 2, Predict 2  
Actual 1, Predict 1  
Actual 8, Predict 8  
Actual 1, Predict 1  
Actual 1, Predict 1  
Actual 8, Predict 8  
Actual 0, Predict 0  
Actual 2, Predict 2  
Actual 7, Predict 7  
Actual 0, Predict 0  
Actual 1, Predict 1  
Actual 7, Predict 7  
Actual 2, Predict 2  
Actual 3, Predict 3  
Actual 6, Predict 6  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 8, Predict 8  
Actual 9, Predict 9  
Actual 3, Predict 3  
Actual 8, Predict 8  
Actual 6, Predict 6  
Actual 9, Predict 9  
Actual 7, Predict 7  
Actual 9, Predict 9  
Actual 1, Predict 1  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 3, Predict 3  
Actual 9, Predict 8  
Actual 6, Predict 6  
Actual 6, Predict 6  
Actual 4, Predict 4

Actual 8, Predict 8  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 3, Predict 3  
Actual 2, Predict 2  
Actual 4, Predict 4  
Actual 4, Predict 1  
Actual 7, Predict 7  
Actual 2, Predict 2  
Actual 2, Predict 2  
Actual 7, Predict 7  
Actual 0, Predict 0  
Actual 5, Predict 5  
Actual 3, Predict 3  
Actual 6, Predict 6  
Actual 0, Predict 0  
Actual 8, Predict 8  
Actual 4, Predict 4  
Actual 9, Predict 9  
Actual 9, Predict 9  
Actual 2, Predict 2  
Actual 1, Predict 1  
Actual 6, Predict 6  
Actual 5, Predict 5  
Actual 4, Predict 4  
Actual 1, Predict 1  
Actual 7, Predict 7  
Actual 9, Predict 5  
Actual 3, Predict 3  
Actual 2, Predict 2  
Actual 4, Predict 4  
Actual 2, Predict 2  
Actual 1, Predict 1  
Actual 1, Predict 1  
Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 6, Predict 6  
Actual 4, Predict 4  
Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 1, Predict 1  
Actual 8, Predict 8  
Actual 1, Predict 1  
Actual 1, Predict 1  
Actual 3, Predict 3  
Actual 5, Predict 5  
Actual 8, Predict 8  
Actual 2, Predict 2  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 6, Predict 6  
Actual 2, Predict 2  
Actual 6, Predict 6

Actual 6, Predict 6  
Actual 5, Predict 5  
Actual 5, Predict 5  
Actual 2, Predict 2  
Actual 3, Predict 3  
Actual 2, Predict 2  
Actual 0, Predict 0  
Actual 2, Predict 2  
Actual 4, Predict 4  
Actual 8, Predict 8  
Actual 7, Predict 7  
Actual 4, Predict 4  
Actual 5, Predict 5  
Actual 1, Predict 1  
Actual 4, Predict 4  
Actual 3, Predict 3  
Actual 4, Predict 4  
Actual 3, Predict 3  
Actual 8, Predict 8  
Actual 6, Predict 6  
Actual 3, Predict 3  
Actual 3, Predict 3  
Actual 4, Predict 4  
Actual 5, Predict 5  
Actual 3, Predict 3  
Actual 3, Predict 3  
Actual 0, Predict 0  
Actual 6, Predict 6  
Actual 7, Predict 7  
Actual 2, Predict 2  
Actual 4, Predict 4  
Actual 7, Predict 7  
Actual 4, Predict 4  
Actual 7, Predict 7  
Actual 1, Predict 1  
Actual 2, Predict 2  
Actual 6, Predict 6  
Actual 8, Predict 8  
Actual 8, Predict 8  
Actual 2, Predict 2  
Actual 2, Predict 2  
Actual 7, Predict 7  
Actual 4, Predict 4  
Actual 9, Predict 9  
Actual 5, Predict 5  
Actual 0, Predict 0  
Actual 4, Predict 4  
Actual 1, Predict 1  
Actual 8, Predict 5  
Actual 4, Predict 4  
Actual 0, Predict 0  
Actual 4, Predict 4  
Actual 3, Predict 3  
Actual 4, Predict 4  
Actual 1, Predict 1  
Actual 2

Actual 1, Predict 1  
Actual 4, Predict 4  
Actual 5, Predict 5  
Actual 0, Predict 0  
Actual 1, Predict 1  
Actual 9, Predict 9  
Actual 2, Predict 2  
Actual 1, Predict 1  
Actual 2, Predict 2  
Actual 5, Predict 5  
Actual 2, Predict 2  
Actual 0, Predict 0  
Actual 6, Predict 5  
Actual 3, Predict 3  
Actual 3, Predict 3  
Actual 3, Predict 3  
Actual 4, Predict 4  
Actual 8, Predict 8  
Actual 7, Predict 7  
Actual 1, Predict 1  
Actual 3, Predict 3  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 5, Predict 5  
Actual 3, Predict 3  
Actual 9, Predict 9  
Actual 8, Predict 8  
Actual 6, Predict 6  
Actual 8, Predict 8  
Actual 5, Predict 5  
Actual 2, Predict 2  
Actual 1, Predict 1  
Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 5, Predict 5  
Actual 2, Predict 2  
Actual 5, Predict 5  
Actual 5, Predict 5  
Actual 1, Predict 1  
Actual 6, Predict 6  
Actual 9, Predict 9  
Actual 9, Predict 9  
Actual 5, Predict 5  
Actual 1, Predict 1  
Actual 4, Predict 4  
Actual 6, Predict 6  
Actual 1, Predict 1  
Actual 7, Predict 7  
Actual 0, Predict 0  
Actual 7, Predict 7  
Actual 3, Predict 3  
Actual 7, Predict 7  
Actual 9, Predict 9

Actual 8, Predict 8  
Actual 0, Predict 0  
Actual 9, Predict 9  
Actual 9, Predict 9  
Actual 3, Predict 3  
Actual 9, Predict 9  
Actual 0, Predict 0  
Actual 2, Predict 2  
Actual 9, Predict 9  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 6, Predict 6  
Actual 7, Predict 7  
Actual 2, Predict 2  
Actual 3, Predict 3  
Actual 1, Predict 1  
Actual 9, Predict 9  
Actual 4, Predict 4  
Actual 2, Predict 2  
Actual 8, Predict 8  
Actual 2, Predict 2  
Actual 3, Predict 3  
Actual 4, Predict 4  
Actual 9, Predict 9  
Actual 2, Predict 2  
Actual 5, Predict 5  
Actual 7, Predict 7  
Actual 0, Predict 0  
Actual 9, Predict 9  
Actual 1, Predict 1  
Actual 0, Predict 0  
Actual 5, Predict 5  
Actual 9, Predict 9  
Actual 6, Predict 6  
Actual 4, Predict 4  
Actual 5, Predict 5  
Actual 7, Predict 7  
Actual 7, Predict 7  
Actual 3, Predict 3  
Actual 5, Predict 5  
Actual 6, Predict 6  
Actual 5, Predict 5  
Actual 8, Predict 8  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 0, Predict 0  
Actual 8, Predict 8  
Actual 1, Predict 1  
Actual 8, Predict 8  
Actual 8, Predict 8  
Actual 8, Predict 8  
Actual 4, Predict 4  
Actual 7, Predict 7  
Actual 4, Predict 4  
Actual 9, Predict 9  
Actual 6, Predict 6

```

Actual 8, Predict 8
Actual 6, Predict 6
Actual 3, Predict 3
Actual 9, Predict 9
Actual 6, Predict 6
Actual 3, Predict 3
Actual 9, Predict 9
Actual 5, Predict 5
Actual 7, Predict 7
Actual 0, Predict 0
Actual 5, Predict 5
Actual 9, Predict 9
Actual 7, Predict 7
Actual 7, Predict 7
Actual 8, Predict 8
Actual 0, Predict 0
Actual 6, Predict 6
Actual 7, Predict 7
Actual 0, Predict 0
Actual 1, Predict 1
Actual 7, Predict 7
Actual 9, Predict 9
Actual 5, Predict 5
Actual 2, Predict 2

```

Now, since our accuracy score is pretty high, means it will get almost all the prediction right

So how do we know where our model is not doing well? since all the samples we tried, it did pretty well.

One of the ways of doing this is, Confusion Matrix.

In Confusion matrix, you give it `y_test`, which is the actual value and the predicted value, then you get confusion matrix back

```
In [62]: from sklearn.metrics import confusion_matrix
```

```
In [64]: cm = confusion_matrix(y_test, predict)
cm
```

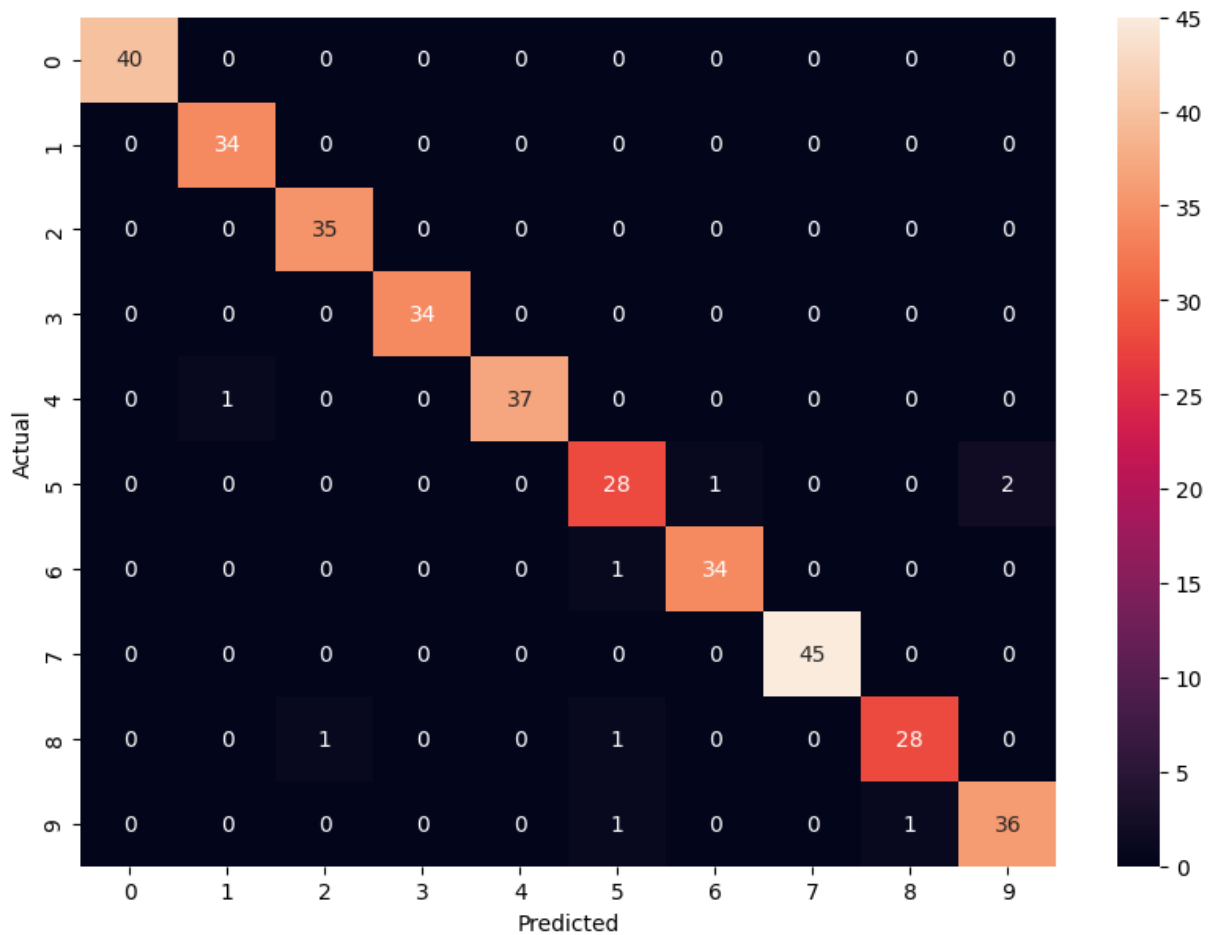
```
Out[64]: array([[40,  0,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 34,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0, 35,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0, 34,  0,  0,  0,  0,  0,  0],
 [ 0,  1,  0,  0, 37,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0, 28,  1,  0,  0,  2],
 [ 0,  0,  0,  0,  0,  1, 34,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0, 45,  0,  0],
 [ 0,  0,  1,  0,  0,  1,  0,  0, 28,  0],
 [ 0,  0,  0,  0,  0,  1,  0,  0,  1, 36]], dtype=int64)
```

As seen above, it returns an array, if you don't understand this, let's visualize it using Seaborn to understand better

```
In [67]: import seaborn as sn
```

```
In [68]: plt.figure(figsize=(10, 7))  
sn.heatmap(cm, annot=True)  
plt.xlabel('Predicted')  
plt.ylabel('Actual')
```

```
Out[68]: Text(95.7222222222221, 0.5, 'Actual')
```



Lets try to understand the above visual

For example, you see 40 highlighted from top left, it means that 40 times the actual value was 0 and the model predicted it 0 as well.

Then 1 means, 1 time the actual value was 4 but the model predicted it to be 1.

So basically, anywhere in the the black/dark blue area where you dont see 0, its where your model is not Predicting correctly!

Confusion matrix is just a nice way of Visualizing how well your model is doing.