```
In [5]: import re
        from sklearn.datasets import load_digits
        from sklearn.model_selection import train_test_split
        import numpy as np

        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn import metrics

        %matplotlib inline
        digits=load_digits()
```
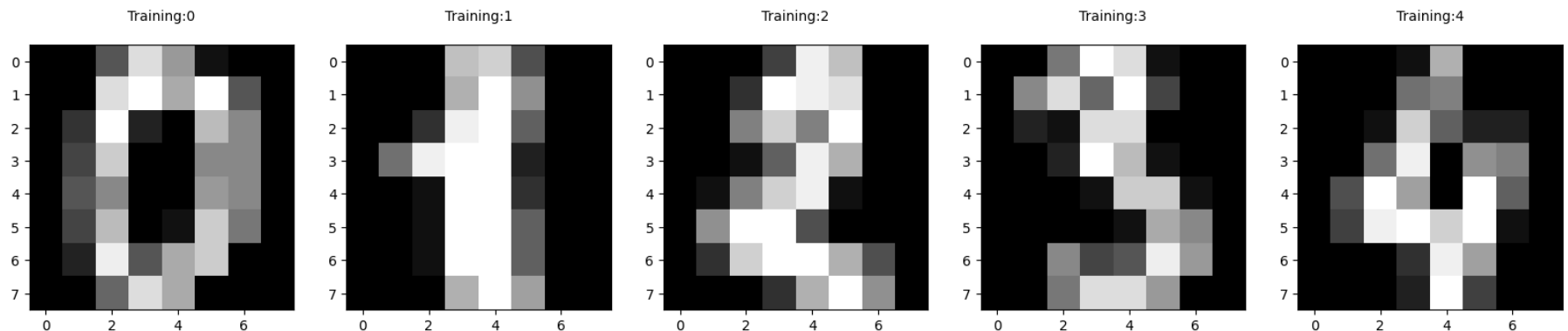
```
In [6]: print("Image Data Shape",digits.data.shape)
        print("Label Data Shape",digits.target.shape)
```

```
Image Data Shape (1797, 64)
Label Data Shape (1797,)
```

```
In [22]: plt.figure(figsize=(20,4))
         for index,(image,label)in enumerate(zip(digits.data[0:5],digits.target[0:5])):
             plt.subplot(1,5,index+1)
             plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
             plt.title('Training:%i\n'%label,fontsize=10)
```

```
In [24]:  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=0.30,random_state=2)
```

```
In [25]:  print(x_train.shape)
```

```
(1257, 64)
```

```
In [26]:  print(y_train.shape)
```

```
(1257,)
```

```
In [28]:  print(x_test.shape)
```

```
(540, 64)
```

```
In [29]:  print(y_test.shape)
```

```
(540,)
```

```
In [31]:  from sklearn.linear_model import LogisticRegression
```

```
In [34]:  logisticRegr=LogisticRegression(max_iter=10000)
          logisticRegr.fit(x_train,y_train)
```

Out[34]: LogisticRegression(max_iter=10000)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [35]: print(logisticRegr.predict(x_test))
```

```
[4 0 9 1 8 7 1 5 1 6 6 7 6 1 5 5 8 6 2 7 4 6 4 1 5 2 9 5 4 6 5 6 3 4 0 9 9
 8 4 6 8 8 5 7 9 8 9 6 1 7 0 1 9 7 3 3 1 8 8 8 9 8 5 8 4 9 3 5 8 4 3 1 3 8
 7 3 3 0 8 7 2 8 5 3 8 7 6 4 6 2 2 0 1 1 5 3 5 7 1 8 2 2 6 4 6 7 3 7 3 9 4
 7 0 3 5 1 5 0 3 9 2 7 3 2 0 8 1 9 2 1 5 1 0 3 4 3 0 8 3 2 2 7 3 1 6 7 2 8
 3 1 1 6 4 8 2 1 8 4 1 3 1 1 9 5 4 8 7 4 8 9 5 7 6 9 4 0 4 0 0 9 0 6 5 8 8
 3 7 9 2 0 8 2 7 3 0 2 1 9 2 7 0 6 9 3 1 1 3 5 2 5 5 2 1 2 9 4 6 5 5 5 9 7
 1 5 9 6 3 7 1 7 5 1 7 2 7 5 5 4 8 6 6 2 8 7 3 7 8 0 9 5 7 4 3 4 1 0 3 3 5
 4 1 3 1 2 5 1 4 0 3 1 5 5 7 4 0 1 0 9 5 5 5 4 0 1 8 6 2 1 1 1 7 9 6 7 9 7
 0 4 9 6 9 2 7 2 1 0 8 2 8 6 5 7 8 4 5 7 8 6 4 2 6 9 3 0 0 8 0 6 6 7 1 4 5
 6 9 7 2 8 5 1 2 4 1 8 8 7 6 0 8 0 6 1 5 7 8 0 4 1 4 5 9 2 2 3 9 1 3 9 3 2
 8 0 6 5 6 2 5 2 3 2 6 1 0 7 6 0 6 2 7 0 3 2 4 2 3 6 9 7 7 0 3 5 4 1 2 2 1
 2 7 7 0 4 9 8 5 6 1 6 5 2 0 8 2 4 3 3 2 9 3 8 9 9 5 9 0 3 4 7 9 8 5 7 5 0
 5 3 5 0 2 7 3 0 4 3 6 6 1 9 6 3 4 6 4 6 7 2 7 6 3 0 3 0 1 3 6 1 0 4 3 8 4
 3 3 4 8 6 9 6 3 3 0 5 7 8 9 1 5 3 2 5 1 7 6 0 6 9 5 2 4 4 7 2 0 5 6 2 0 8
 4 4 4 7 1 0 4 1 9 2 1 3 0 5 3 9 8 2 6 0 0 4]
```

```
In [37]: score=logisticRegr.score(x_test,y_test)
         print(score)
```

```
0.9537037037037037
```