In [1]:
```
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 [2]:
```
print("Image Data Shape",digits.data.shape)
print("Label Data Shape",digits.target.shape)
```

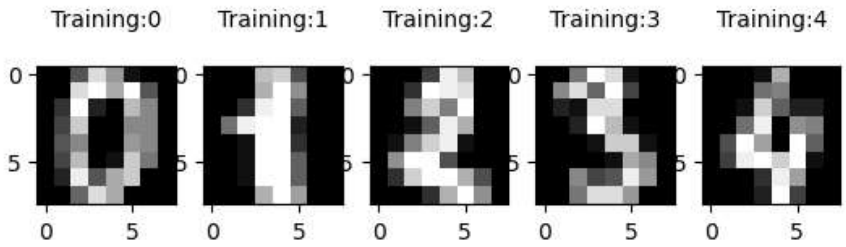
```
Image Data Shape (1797, 64)
Label Data Shape (1797,)
```

In [3]:
```
plt.figure(figsize=(20,4))
```

Out[3]:
```
<Figure size 2000x400 with 0 Axes>
```

```
<Figure size 2000x400 with 0 Axes>
```

In [5]:
```
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 [6]:
```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.30,random_state=42)
```

In [7]:
```
print(x_train.shape)
```

```
(1257, 64)
```

In [8]:
```
print(y_train.shape)
```

```
(1257,)
```

In [9]:
```
print(x_test.shape)
```

```
(540, 64)
```

In [10]:
```
print(y_test.shape)
```

```
(540,)
```

In [11]:
```
from sklearn.linear_model import LogisticRegression
logisticRegr=LogisticRegression(max_iter=10000)
```

In [12]:
```
logisticRegr.fit(x_train,y_train)
```

Out[12]:
```
LogisticRegression
LogisticRegression(max_iter=10000)
```

In [13]:
```
print(logisticRegr.predict(x_test))
```

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

In [14]:
```
score=logisticRegr.score(x_test,y_test)
print(score)
```

```
0.9722222222222222
```