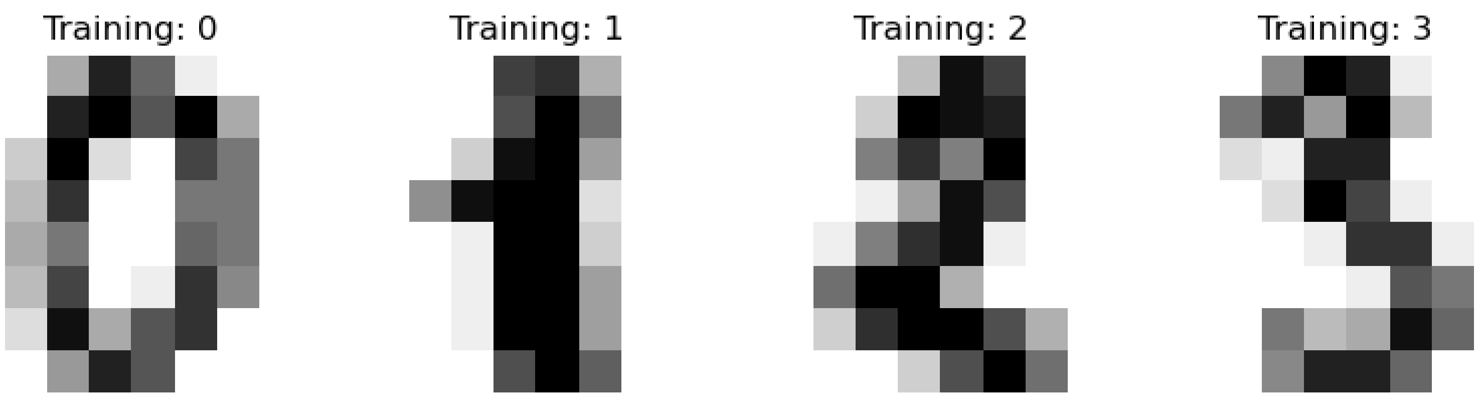
```
In [1]:  import pandas as pd

In [2]:  import numpy as np

In [3]:  import matplotlib.pyplot as plt

In [4]:  from sklearn.datasets import load_digits

In [5]:  df=load_digits()

In [6]:  _,axes=plt.subplots(nrows=1,ncols=4,figsize=(10,3))
         for ax,image,lable in zip(axes,df.images,df.target):
             ax.set_axis_off()
             ax.imshow(image,cmap=plt.cm.gray_r,interpolation='nearest')
             ax.set_title('Training: %i'%lable)
```



```
In [7]:  df.images.shape

Out[7]:  (1797, 8, 8)

In [8]:  df.images[0]

Out[8]:  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.]])

In [9]:  df.images[0].shape

Out[9]:  (8, 8)

In [10]: len(df.images)

Out[10]: 1797

In [12]: n_samples=len(df.images)
         data=df.images.reshape((n_samples,-1))

In [13]: data[0]

Out[13]: 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.])

In [14]: data[0].shape

Out[14]: (64,)

In [15]: data.shape

Out[15]: (1797, 64)

In [16]: data.min()

Out[16]: 0.0

In [17]: data.max()

Out[17]: 16.0

In [18]: data=data/16

In [20]: data.min()

Out[20]: 0.0

In [21]: data.max()

Out[21]: 1.0

In [22]: data[0]

Out[22]: array([0.    , 0.    , 0.3125, 0.8125, 0.5625, 0.0625, 0.    , 0.    ,
                0.    , 0.    , 0.8125, 0.9375, 0.625 , 0.9375, 0.3125, 0.    ,
                0.    , 0.1875, 0.9375, 0.125 , 0.    , 0.6875, 0.5   , 0.    ,
                0.    , 0.25  , 0.75  , 0.    , 0.    , 0.5   , 0.5   , 0.    ,
                0.    , 0.3125, 0.5   , 0.    , 0.    , 0.5625, 0.5   , 0.    ,
                0.    , 0.25  , 0.6875, 0.    , 0.0625, 0.75  , 0.4375, 0.    ,
                0.    , 0.125 , 0.875 , 0.3125, 0.625 , 0.75  , 0.    , 0.    ,
                0.    , 0.    , 0.375 , 0.8125, 0.625 , 0.    , 0.    , 0.    ])

In [23]: from sklearn.model_selection import train_test_split

In [28]: x_train,x_test,y_train,y_test=train_test_split(data, df.target, test_size=0.3)

In [29]: x_train.shape,x_test.shape,y_train.shape,y_test.shape

Out[29]: ((1257, 64), (540, 64), (1257,), (540,))

In [30]: from sklearn.ensemble import RandomForestClassifier

In [31]: rf=RandomForestClassifier()

In [32]: rf.fit(x_train,y_train)

Out[32]: ▾ RandomForestClassifier
         RandomForestClassifier()

In [33]: y_pred=rf.predict(x_test)

In [34]: y_pred

Out[34]: array([5, 8, 8, 3, 7, 7, 7, 6, 6, 5, 9, 6, 7, 7, 6, 9, 3, 5, 7, 9, 8, 3,
                2, 7, 4, 1, 7, 2, 7, 6, 5, 4, 2, 5, 5, 2, 7, 5, 7, 0, 0, 4, 7, 1,
                8, 3, 0, 4, 7, 5, 5, 2, 8, 5, 8, 2, 3, 2, 9, 6, 5, 2, 0, 8, 1, 1,
                8, 9, 2, 1, 9, 2, 9, 0, 4, 3, 1, 5, 9, 7, 0, 8, 1, 0, 6, 4, 8, 1,
                1, 3, 3, 5, 1, 1, 4, 7, 4, 2, 3, 8, 9, 5, 8, 6, 0, 7, 7, 6, 3, 3,
                6, 7, 5, 8, 3, 1, 6, 9, 1, 6, 6, 2, 2, 4, 3, 2, 3, 0, 1, 6, 7, 9,
                9, 5, 7, 7, 3, 0, 4, 3, 5, 0, 3, 8, 9, 7, 1, 5, 4, 9, 6, 2, 7, 5,
                4, 4, 0, 8, 2, 1, 3, 2, 6, 3, 4, 4, 6, 1, 8, 1, 7, 1, 7, 7, 6, 5,
                9, 6, 2, 4, 8, 0, 7, 9, 7, 4, 7, 7, 3, 3, 2, 6, 3, 1, 4, 9, 9, 8,
                8, 1, 0, 1, 1, 6, 3, 7, 9, 4, 7, 7, 2, 8, 4, 7, 2, 4, 1, 1, 4, 5,
                7, 0, 2, 5, 0, 9, 2, 3, 9, 9, 4, 4, 1, 1, 8, 6, 8, 3, 1, 8, 6, 8,
                7, 5, 9, 8, 5, 3, 5, 4, 7, 4, 5, 3, 5, 5, 4, 4, 4, 1, 2, 9, 5, 4,
                9, 6, 5, 5, 2, 7, 2, 7, 3, 7, 3, 5, 7, 0, 6, 2, 0, 1, 3, 5, 1, 2,
                2, 6, 9, 8, 5, 3, 9, 9, 0, 2, 0, 1, 5, 2, 7, 1, 8, 0, 4, 5, 3, 0,
                2, 7, 6, 9, 5, 3, 5, 4, 1, 1, 8, 0, 5, 4, 5, 4, 0, 7, 1, 4, 9, 9,
                1, 8, 9, 7, 9, 6, 8, 6, 1, 5, 5, 7, 8, 4, 6, 2, 8, 1, 5, 6, 1, 2,
                7, 7, 8, 5, 3, 1, 6, 4, 1, 8, 7, 5, 2, 7, 4, 3, 4, 9, 6, 7, 9, 4,
                0, 3, 4, 5, 2, 8, 2, 7, 8, 6, 4, 3, 3, 7, 3, 2, 1, 8, 7, 1, 4, 8,
                1, 4, 0, 0, 2, 6, 9, 9, 6, 9, 4, 8, 7, 2, 8, 7, 2, 3, 7, 7, 1, 6,
                2, 5, 0, 9, 6, 6, 6, 6, 4, 7, 7, 9, 4, 2, 3, 1, 1, 3, 0, 4, 2, 4,
                9, 4, 2, 0, 2, 1, 6, 6, 3, 3, 4, 3, 6, 2, 7, 0, 2, 0, 5, 3, 6, 3,
                0, 6, 3, 7, 8, 1, 8, 2, 1, 2, 5, 3, 6, 0, 7, 5, 2, 0, 7, 6, 2, 7,
                5, 0, 1, 6, 1, 4, 0, 7, 4, 0, 7, 8, 0, 3, 4, 9, 1, 1, 9, 9, 7, 6,
                8, 4, 5, 3, 5, 9, 4, 8, 0, 3, 0, 6, 6, 8, 9, 7, 3, 5, 3, 0, 8, 1,
                7, 9, 5, 3, 9, 0, 3, 5, 5, 5, 4, 7])

In [35]: from sklearn.metrics import confusion_matrix,classification_report

In [36]: confusion_matrix(y_test,y_pred)

Out[36]: array([[43,  0,  0,  0,  0,  0,  0,  0,  0,  0],
                [ 0, 51,  0,  0,  0,  0,  0,  0,  0,  0],
                [ 0,  1, 52,  0,  0,  0,  0,  0,  0,  0],
                [ 0,  1,  0, 56,  0,  2,  0,  0,  0,  0],
                [ 0,  0,  0,  0, 55,  0,  0,  3,  0,  0],
                [ 0,  0,  0,  0,  0, 55,  0,  0,  0,  1],
                [ 0,  0,  0,  0,  2,  0, 52,  0,  0,  0],
                [ 0,  0,  0,  0,  0,  0,  0, 65,  0,  1],
                [ 0,  3,  0,  0,  0,  1,  0,  1, 46,  0],
                [ 0,  0,  0,  0,  0,  0,  0,  2,  1, 46]], dtype=int64)

In [37]: y_pred

Out[37]: array([5, 8, 8, 3, 7, 7, 7, 6, 6, 5, 9, 6, 7, 7, 6, 9, 3, 5, 7, 9, 8, 3,
                2, 7, 4, 1, 7, 2, 7, 6, 5, 4, 2, 5, 5, 2, 7, 5, 7, 0, 0, 4, 7, 1,
                8, 3, 0, 4, 7, 5, 5, 2, 8, 5, 8, 2, 3, 2, 9, 6, 5, 2, 0, 8, 1, 1,
                8, 9, 2, 1, 9, 2, 9, 0, 4, 3, 1, 5, 9, 7, 0, 8, 1, 0, 6, 4, 8, 1,
                1, 3, 3, 5, 1, 1, 4, 7, 4, 2, 3, 8, 9, 5, 8, 6, 0, 7, 7, 6, 3, 3,
                6, 7, 5, 8, 3, 1, 6, 9, 1, 6, 6, 2, 2, 4, 3, 2, 3, 0, 1, 6, 7, 9,
                9, 5, 7, 7, 3, 0, 4, 3, 5, 0, 3, 8, 9, 7, 1, 5, 4, 9, 6, 2, 7, 5,
                4, 4, 0, 8, 2, 1, 3, 2, 6, 3, 4, 4, 6, 1, 8, 1, 7, 1, 7, 7, 6, 5,
                9, 6, 2, 4, 8, 0, 7, 9, 7, 4, 7, 7, 3, 3, 2, 6, 3, 1, 4, 9, 9, 8,
                8, 1, 0, 1, 1, 6, 3, 7, 9, 4, 7, 7, 2, 8, 4, 7, 2, 4, 1, 1, 4, 5,
                7, 0, 2, 5, 0, 9, 2, 3, 9, 9, 4, 4, 1, 1, 8, 6, 8, 3, 1, 8, 6, 8,
                7, 5, 9, 8, 5, 3, 5, 4, 7, 4, 5, 3, 5, 5, 4, 4, 4, 1, 2, 9, 5, 4,
                9, 6, 5, 5, 2, 7, 2, 7, 3, 7, 3, 5, 7, 0, 6, 2, 0, 1, 3, 5, 1, 2,
                2, 6, 9, 8, 5, 3, 9, 9, 0, 2, 0, 1, 5, 2, 7, 1, 8, 0, 4, 5, 3, 0,
                2, 7, 6, 9, 5, 3, 5, 4, 1, 1, 8, 0, 5, 4, 5, 4, 0, 7, 1, 4, 9, 9,
                1, 8, 9, 7, 9, 6, 8, 6, 1, 5, 5, 7, 8, 4, 6, 2, 8, 1, 5, 6, 1, 2,
                7, 7, 8, 5, 3, 1, 6, 4, 1, 8, 7, 5, 2, 7, 4, 3, 4, 9, 6, 7, 9, 4,
                0, 3, 4, 5, 2, 8, 2, 7, 8, 6, 4, 3, 3, 7, 3, 2, 1, 8, 7, 1, 4, 8,
                1, 4, 0, 0, 2, 6, 9, 9, 6, 9, 4, 8, 7, 2, 8, 7, 2, 3, 7, 7, 1, 6,
                2, 5, 0, 9, 6, 6, 6, 6, 4, 7, 7, 9, 4, 2, 3, 1, 1, 3, 0, 4, 2, 4,
                9, 4, 2, 0, 2, 1, 6, 6, 3, 3, 4, 3, 6, 2, 7, 0, 2, 0, 5, 3, 6, 3,
                0, 6, 3, 7, 8, 1, 8, 2, 1, 2, 5, 3, 6, 0, 7, 5, 2, 0, 7, 6, 2, 7,
                5, 0, 1, 6, 1, 4, 0, 7, 4, 0, 7, 8, 0, 3, 4, 9, 1, 1, 9, 9, 7, 6,
                8, 4, 5, 3, 5, 9, 4, 8, 0, 3, 0, 6, 6, 8, 9, 7, 3, 5, 3, 0, 8, 1,
                7, 9, 5, 3, 9, 0, 3, 5, 5, 5, 4, 7])

In [39]: from sklearn.metrics import confusion_matrix,classification_report

In [40]: confusion_matrix(y_test,y_pred)

Out[40]: array([[43,  0,  0,  0,  0,  0,  0,  0,  0,  0],
                [ 0, 51,  0,  0,  0,  0,  0,  0,  0,  0],
                [ 0,  1, 52,  0,  0,  0,  0,  0,  0,  0],
                [ 0,  1,  0, 56,  0,  2,  0,  0,  0,  0],
                [ 0,  0,  0,  0, 55,  0,  0,  3,  0,  0],
                [ 0,  0,  0,  0,  0, 55,  0,  0,  0,  1],
                [ 0,  0,  0,  0,  2,  0, 52,  0,  0,  0],
                [ 0,  0,  0,  0,  0,  0,  0, 65,  0,  1],
                [ 0,  3,  0,  0,  0,  1,  0,  1, 46,  0],
                [ 0,  0,  0,  0,  0,  0,  0,  2,  1, 46]], dtype=int64)

In [41]: print(classification_report(y_test,y_pred))

                       precision    recall  f1-score   support

                    0       1.00      1.00      1.00        43
                    1       0.91      1.00      0.95        51
                    2       1.00      0.98      0.99        53
                    3       1.00      0.95      0.97        59
                    4       0.96      0.95      0.96        58
                    5       0.95      0.98      0.96        56
                    6       1.00      0.96      0.98        54
                    7       0.92      0.98      0.95        66
                    8       0.98      0.90      0.94        51
                    9       0.96      0.94      0.95        49

             accuracy                           0.96       540
            macro avg       0.97      0.96      0.97       540
         weighted avg       0.97      0.96      0.96       540

In [ ]:
```