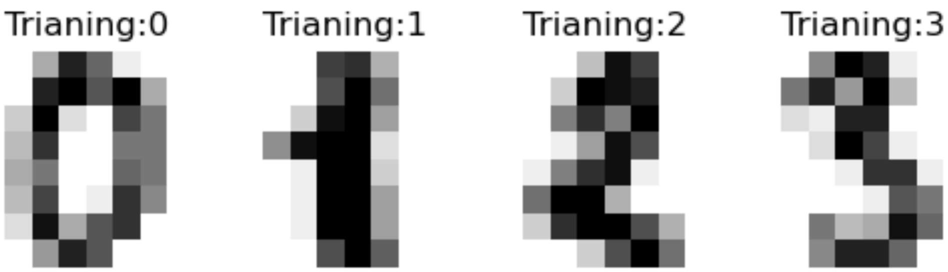
In [1]:
```
%matplotlib inline
import matplotlib
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn import datasets,tree,metrics
```

In [2]:
```
pip install seaborn --upgrade
```

```
Requirement already satisfied: seaborn in d:\app\anaconda\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in d:\app\anaconda\lib\site-pack
ages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=1.2 in d:\app\anaconda\lib\site-packages (fro
m seaborn) (2.1.4)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in d:\app\anaconda\lib\site-p
ackages (from seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in d:\app\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.2.0)
Requirement already satisfied: cycler>=0.10 in d:\app\anaconda\lib\site-packages (fr
om matplotlib!=3.6.1,>=3.4->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in d:\app\anaconda\lib\site-package
s (from matplotlib!=3.6.1,>=3.4->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in d:\app\anaconda\lib\site-package
s (from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in d:\app\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in d:\app\anaconda\lib\site-packages (f
rom matplotlib!=3.6.1,>=3.4->seaborn) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in d:\app\anaconda\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in d:\app\anaconda\lib\site-pack
ages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in d:\app\anaconda\lib\site-packages (fr
om pandas>=1.2->seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in d:\app\anaconda\lib\site-packages
(from pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in d:\app\anaconda\lib\site-packages (from p
ython-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
digits = datasets.load_digits()
```

In [4]:
```
_, axes =plt.subplots(1,4)
images_and_labels = list(zip(digits.images,digits.target))
for ax,(image,label) in zip(axes,images_and_labels[:4]):
    ax.set_axis_off()
    ax.imshow(image,cmap=plt.cm.gray_r,interpolation='nearest')
    ax.set_title('Trianing:%i'% label)
plt.show()
```


Trianing:0 Trianing:1 Trianing:2 Trianing:3

In [5]:
```
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))
```

In [6]:
```
classifier = tree.DecisionTreeClassifier(criterion="entropy",random_state=33)
```

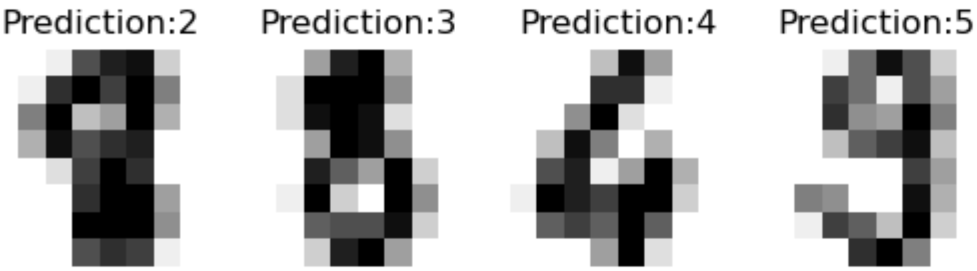
In [7]:
```
x_train,x_test,y_train,y_test = train_test_split(data,digits.target,test_size=0.2,s
```

In [8]:
```
classifier.fit(x_train, y_train)
```

Out[8]:
```
▾                     DecisionTreeClassifier

DecisionTreeClassifier(criterion='entropy', random_state=33)
```

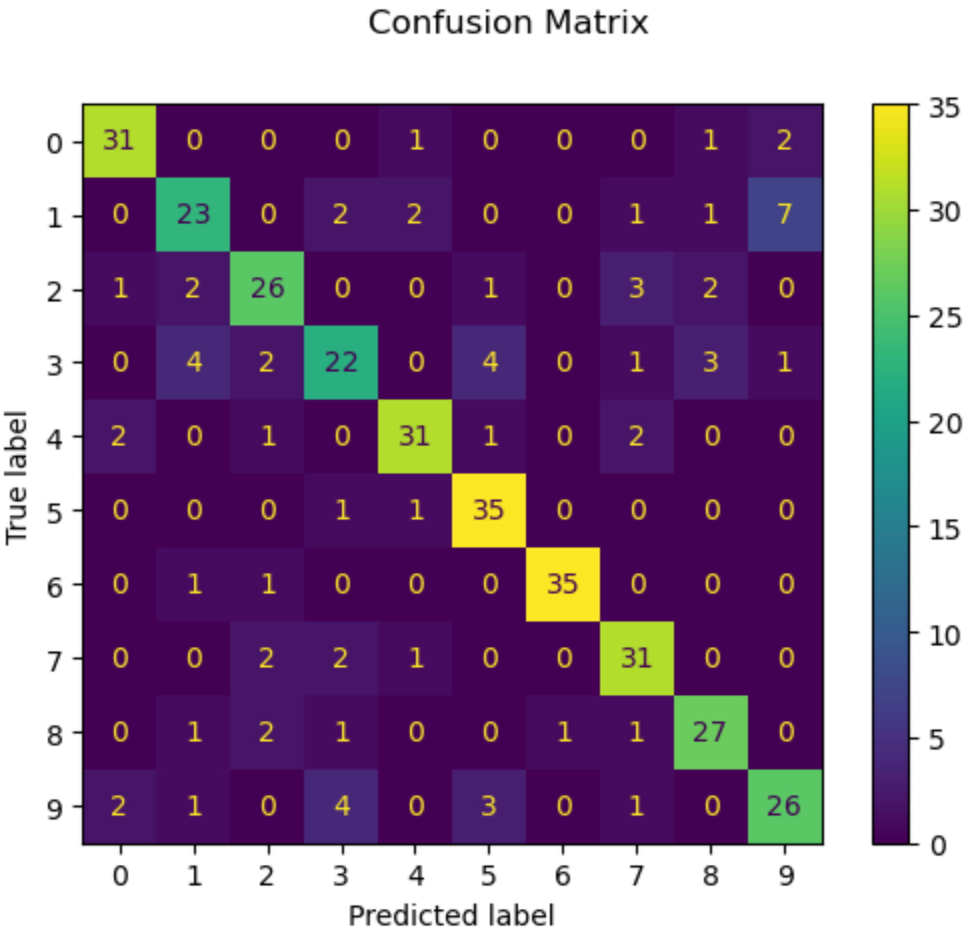
In [9]:
```
predicted = classifier.predict(x_test)
```

In [10]:
```
_, axes =plt.subplots(1,4)
images_and_predictions = list(zip(digits.images[n_samples//2:],predicted))
for ax,(image,prediction) in zip(axes,images_and_predictions[:4]):
    ax.set_axis_off()
    ax.imshow(image,cmap=plt.cm.gray_r,interpolation='nearest')
    ax.set_title('Prediction:%i'% prediction)
plt.show()
```


Prediction:2 Prediction:3 Prediction:4 Prediction:5

In [12]:
```
print("Classificaion report for classifier %s:\n%s\n"
      %(classifier,metrics.classification_report(y_test,predicted)))
disp = metrics.plot_confusion_matrix(classifier,x_test,y_test)
disp.figure_.suptitle("Confusion Matrix")
print("Confustion matri:\m%s"% disp.confusion_matrix)
```

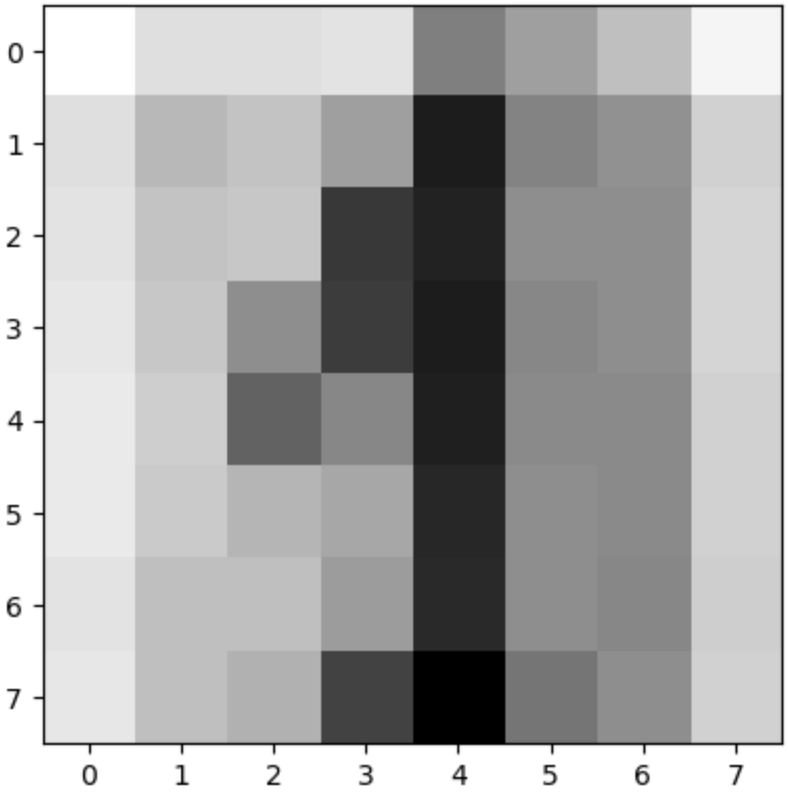
```
Classificaion report for classifier DecisionTreeClassifier(criterion='entropy', rand
om_state=33):
              precision    recall  f1-score   support

           0       0.86      0.89      0.87        35
           1       0.72      0.64      0.68        36
           2       0.76      0.74      0.75        35
           3       0.69      0.59      0.64        37
           4       0.86      0.84      0.85        37
           5       0.80      0.95      0.86        37
           6       0.97      0.95      0.96        37
           7       0.78      0.86      0.82        36
           8       0.79      0.82      0.81        33
           9       0.72      0.70      0.71        37

    accuracy                           0.80       360
   macro avg       0.80      0.80      0.79       360
weighted avg       0.80      0.80      0.79       360


Confustion matri:\m[[31  0  0  0  1  0  0  0  1  2]
 [ 0 23  0  2  2  0  0  1  1  7]
 [ 1  2 26  0  0  1  0  3  2  0]
 [ 0  4  2 22  0  4  0  1  3  1]
 [ 2  0  1  0 31  1  0  2  0  0]
 [ 0  0  0  1  1 35  0  0  0  0]
 [ 0  1  1  0  0  0 35  0  0  0]
 [ 0  0  2  2  1  0  0 31  0  0]
 [ 0  1  2  1  0  0  1  1 27  0]
 [ 2  1  0  4  0  3  0  1  0 26]]
D:\APP\Anaconda\Lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Fu
nction plot_confusion_matrix is deprecated; Function `plot_confusion_matrix` is depr
ecated in 1.0 and will be removed in 1.2. Use one of the class methods: ConfusionMat
rixDisplay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```



In [13]:
```python
from PIL import Image,ImageOps
import numpy as np
```

In [14]:
```python
img = Image.open('1.png').convert("L").resize((8,8))
img = ImageOps.invert(img)
im2arr = np.array(img)
plt.imshow(im2arr, cmap=plt.cm.gray_r, interpolation='nearest')
```

Out[14]:
```
<matplotlib.image.AxesImage at 0x20918ada750>
```

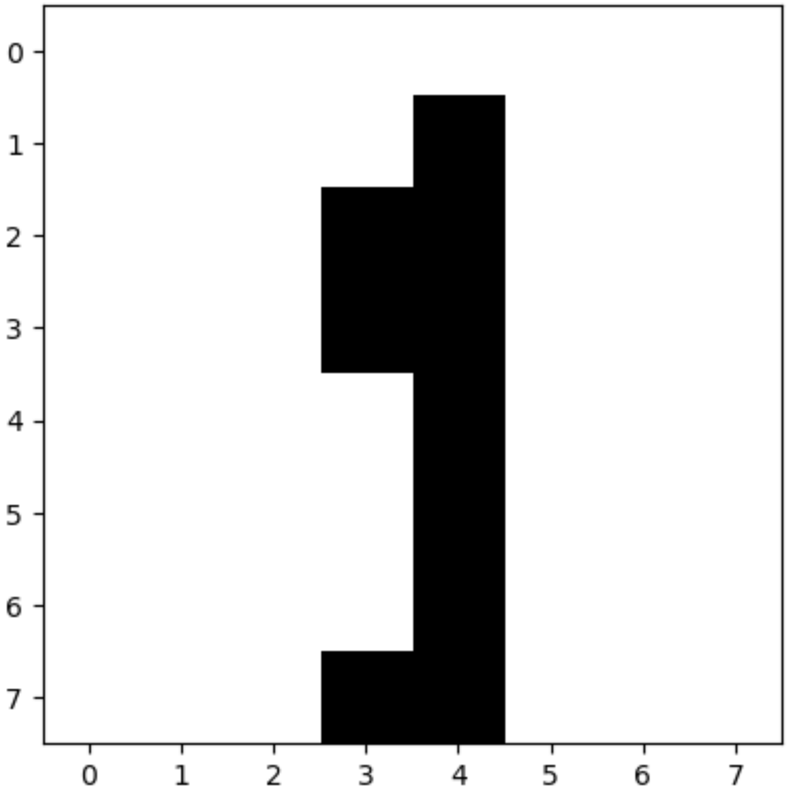


```
In [15]: img1d = im2arr.reshape([1,64])
         img1d[img1d > 109] = 155
         img1d[img1d < 110] = 0
         img1d
```

```
Out[15]: array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 155,
                  0,   0,   0,   0,   0,   0, 155, 155,   0,   0,   0,   0,   0,
                  0, 155, 155,   0,   0,   0,   0,   0,   0,   0, 155,   0,   0,
                  0,   0,   0,   0,   0, 155,   0,   0,   0,   0,   0,   0,   0,
                155,   0,   0,   0,   0,   0,   0, 155, 155,   0,   0,   0]],
               dtype=uint8)
```

```
In [16]: plt.imshow(im2arr, cmap=plt.cm.gray_r, interpolation='nearest')
```

```
Out[16]: <matplotlib.image.AxesImage at 0x20918b7c150>
```



In [17]:

```
y_pred = classifier.predict(img1d)
print(y_pred)
```

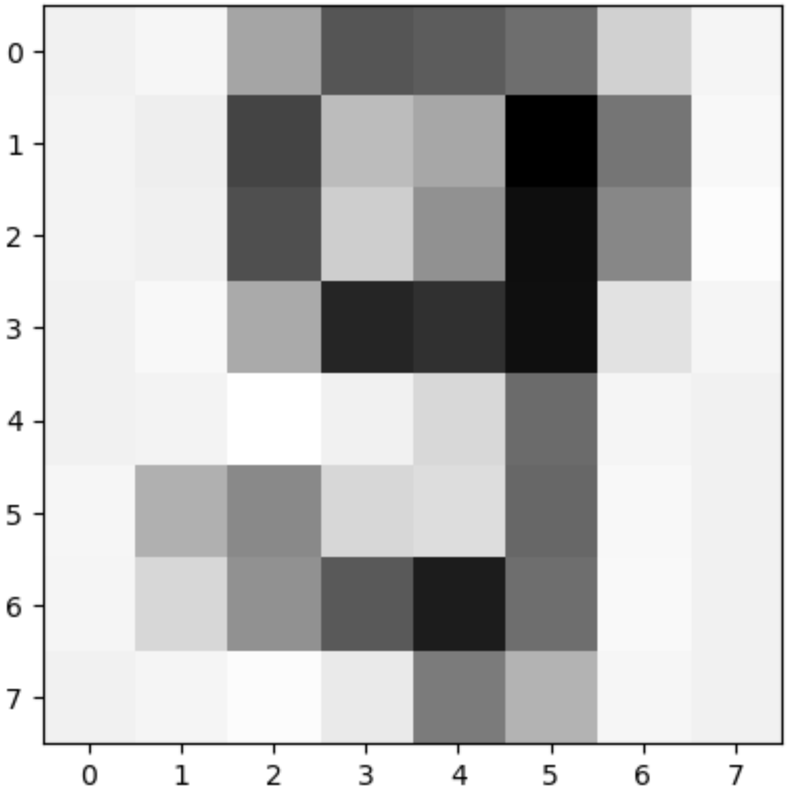
```
[1]
```

In [21]:

```
img = Image.open('9.jpg').convert("L").resize((8,8))
img = ImageOps.invert(img)
im2arr = np.array(img)
plt.imshow(im2arr, cmap=plt.cm.gray_r, interpolation='nearest')
```

Out[21]:

```
<matplotlib.image.AxesImage at 0x20918df2750>
```



In [22]:

```
img1d = im2arr.reshape([1,64])
img1d[img1d > 109] = 155
img1d[img1d < 110] = 0
img1d
```

Out[22]:

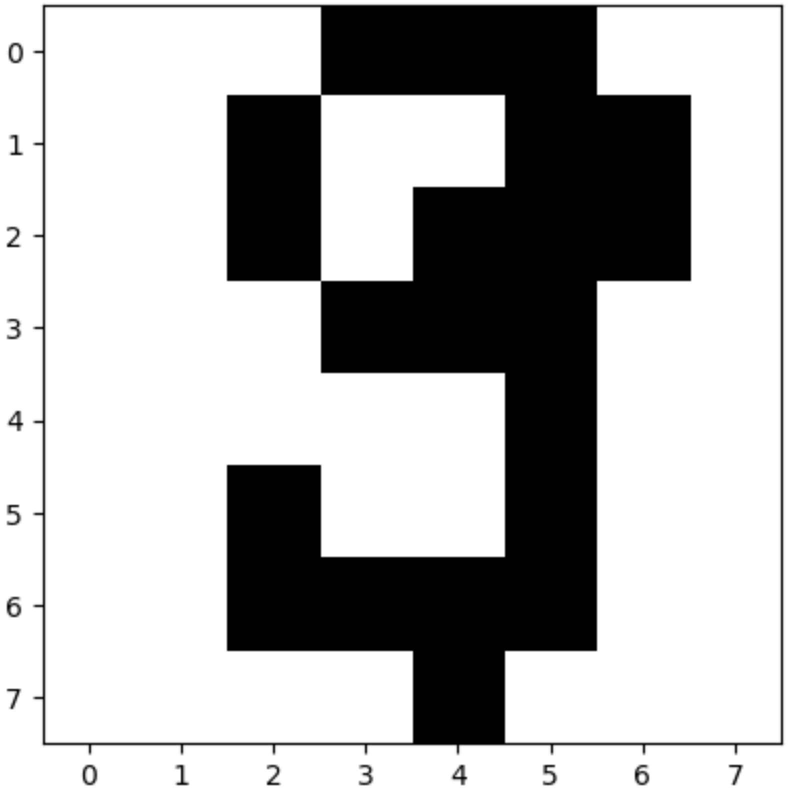
```
array([[  0,   0,   0, 155, 155, 155,   0,   0,   0,   0, 155,   0,   0,
        155, 155,   0,   0,   0, 155,   0, 155, 155, 155,   0,   0,   0,
          0, 155, 155, 155,   0,   0,   0,   0,   0,   0,   0, 155,   0,
          0,   0,   0, 155,   0,   0, 155,   0,   0,   0,   0, 155, 155,
        155, 155,   0,   0,   0,   0,   0,   0, 155,   0,   0,   0]],
      dtype=uint8)
```

In [23]:

```
plt.imshow(im2arr, cmap=plt.cm.gray_r, interpolation='nearest')
```

Out[23]:

```
<matplotlib.image.AxesImage at 0x20918e4a750>
```



In [24]:
```
y_pred = classifier.predict(img1d)
print(y_pred)
```

```
[8]
```