In [1]:
```
import numpy as np
import matplotlib.pyplot as plt

from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, plot_confusion_matrix

%matplotlib inline
```

In [2]:
```
data = load_digits()
```

In [3]:
```
features = data.images
features.shape
```

Out[3]: (1797, 8, 8)

In [4]:
```
labels = data.target
labels.shape
```

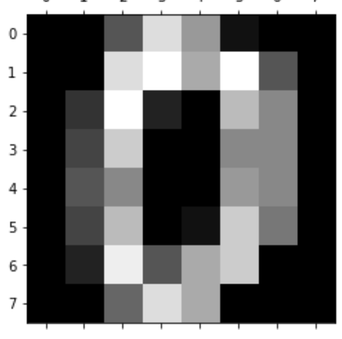
Out[4]: (1797,)

In [5]:
```
data.images[0].shape
```

Out[5]: (8, 8)

In [6]:
```
plt.gray()
plt.matshow(data.images[0])
plt.show()
```

<Figure size 432x288 with 0 Axes>



In [7]:
```
x_train, x_test, y_train, y_test = train_test_split(features, labels, random_state = 45, test_size = 0.2)
```

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

```
(1437, 8, 8)
(1437,)
(360, 8, 8)
(360,)
```

In [9]:
```
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1] * x_test.shape[2])
```

In [10]:
```
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1] * x_train.shape[2])
```

In [11]:
```
print(x_train.shape)
print(x_test.shape)
```

```
(1437, 64)
(360, 64)
```

In [12]:
```
model = KNeighborsClassifier(n_neighbors = 3)
```

In [13]:
```
model.fit(x_train, y_train)
```

Out[13]: KNeighborsClassifier(n_neighbors=3)

In [14]:
```
prediction = model.predict(x_test)
```

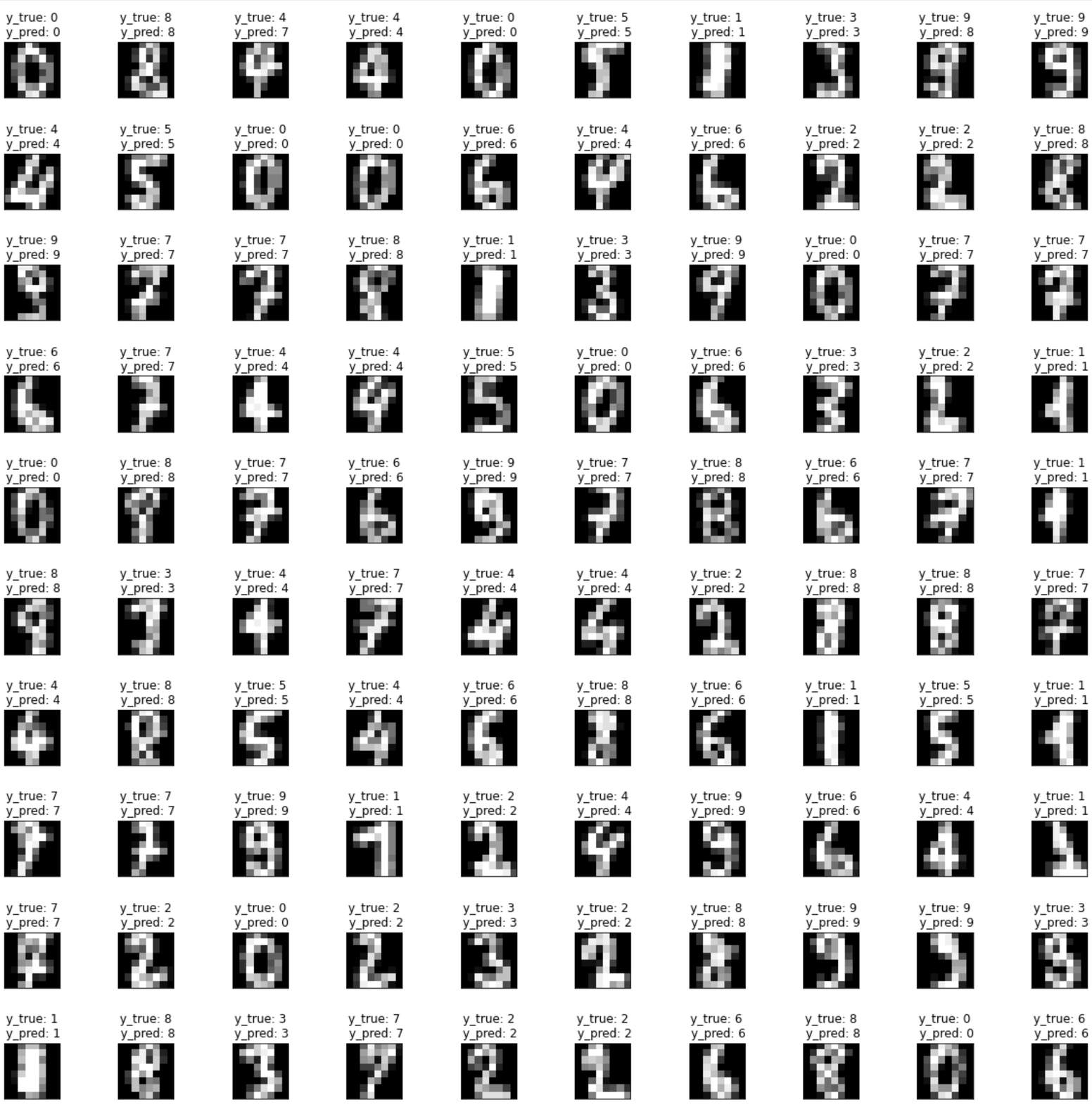
In [15]:
```
accuracy_score(y_test, prediction)
```

Out[15]: 0.9805555555555555

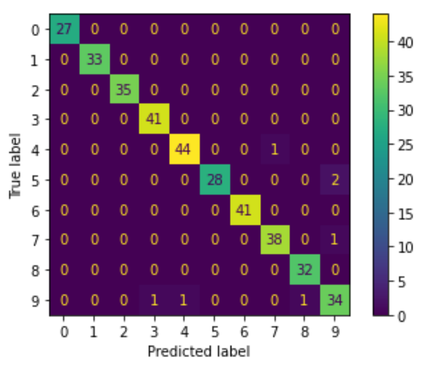
In [16]:
```
fig = plt.figure(figsize = (20, 20))
w = 10
h = 10

for i in range(w * h):
    ax = fig.add_subplot(w, h, i + 1)
    ax.matshow(x_test[i].reshape(8, 8))
    plt.xticks([])
    plt.yticks([])
    plt.subplots_adjust(hspace = 1, wspace = 1)
    plt.title(f'y_true: {y_test[i]}\n y_pred: {prediction[i]}')

plt.show()
```



In [17]:
```
plot_confusion_matrix(model, x_test, y_test)
plt.show()
```



In [ ]:
```

```