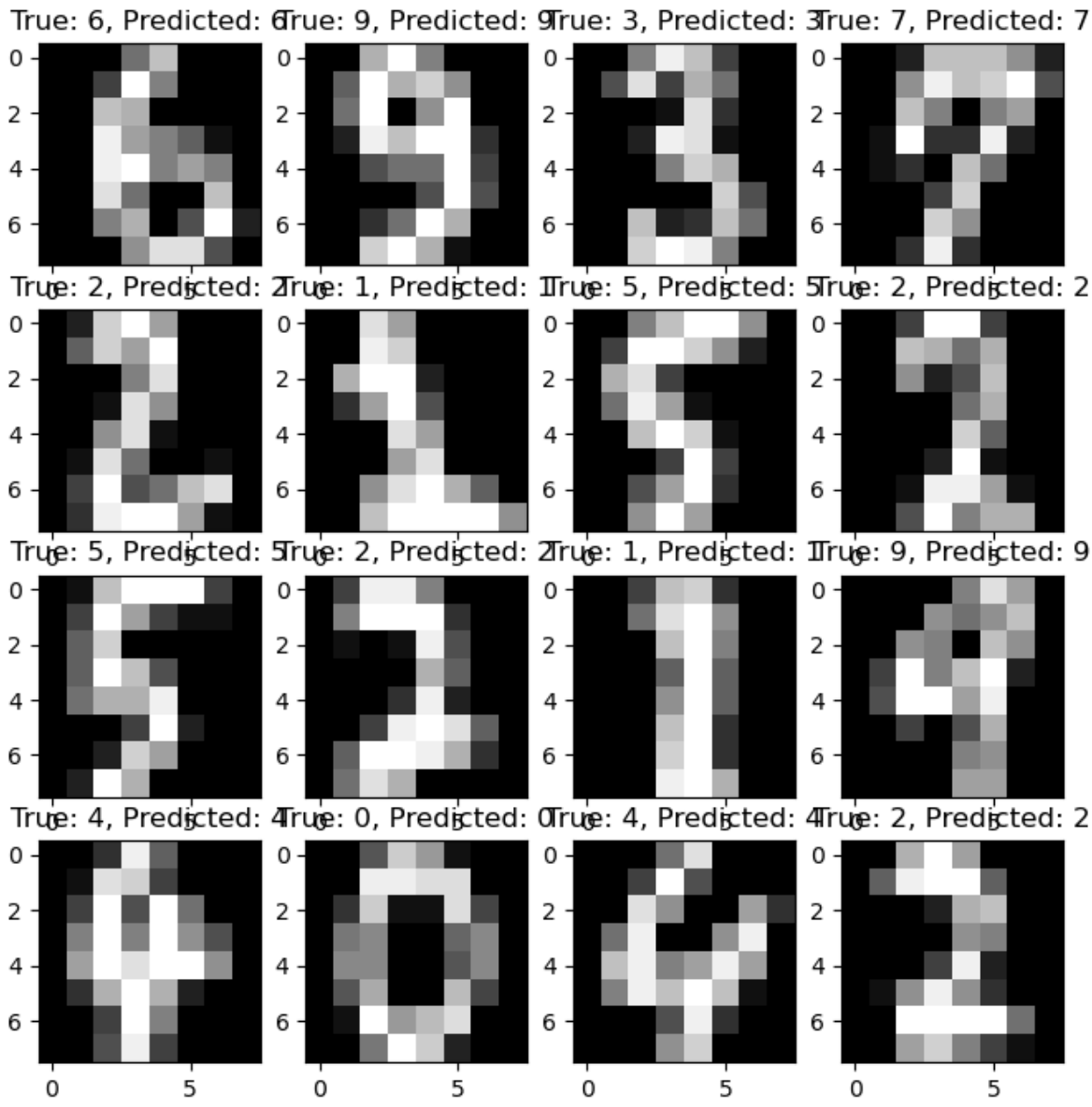
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics

# Load the MNIST dataset
digits = datasets.load_digits()

# Split the dataset into features (X) and target labels (y)
X = digits.images
y = digits.target

# Flatten the 2D images to 1D arrays
n_samples = len(X)
X = X.reshape((n_samples, -1))

# Split the dataset into a training and testing set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_sta

# Create a Support Vector Machine (SVM) classifier
clf = svm.SVC(gamma=0.001)

# Train the classifier on the training data
clf.fit(X_train, y_train)

# Make predictions on the test data
y_pred = clf.predict(X_test)

# Evaluate the classifier's performance
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# Visualize some test samples and their predictions
fig, axes = plt.subplots(4, 4, figsize=(8, 8))
for i, ax in enumerate(axes.flat):
    ax.imshow(X_test[i].reshape(8, 8), cmap=plt.cm.gray)
    ax.set(title=f"True: {y_test[i]}, Predicted: {y_pred[i]}")
plt.show()
```

```
Accuracy: 0.9866518353726362
```


True: 6, Predicted: 6
True: 9, Predicted: 9
True: 3, Predicted: 3
True: 7, Predicted: 7
True: 2, Predicted: 2
True: 1, Predicted: 1
True: 5, Predicted: 5
True: 2, Predicted: 2
True: 5, Predicted: 5
True: 2, Predicted: 2
True: 1, Predicted: 1
True: 9, Predicted: 9
True: 4, Predicted: 4
True: 0, Predicted: 0
True: 4, Predicted: 4
True: 2, Predicted: 2