print(\_\_doc\_\_)

import matplotlib.pyplot as plt

from sklearn import datasets, svm, metrics

from sklearn.model\_selection import train\_test\_split

digits = datasets.load\_digits()

\_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))

for ax, image, label in zip(axes, digits.images, digits.target):

ax.set\_axis\_off()

ax.imshow(image, cmap=plt.cm.gray\_r, interpolation='nearest')

ax.set\_title('Training: %i' % label)

n\_samples = len(digits.images)

data = digits.images.reshape((n\_samples, -1))

clf = svm.SVC(gamma=0.001)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

data, digits.target, test\_size=0.5, shuffle=False)

clf.fit(X\_train, y\_train)

predicted = clf.predict(X\_test)

\_, axes = plt.subplots(nrows=1, ncols=4, figsize=(10, 3))

for ax, image, prediction in zip(axes, X\_test, predicted):

ax.set\_axis\_off()

image = image.reshape(8, 8)

ax.imshow(image, cmap=plt.cm.gray\_r, interpolation='nearest')

ax.set\_title(f'Prediction: {prediction}')

print(f"Classification report for classifier {clf}:\n"

f"{metrics.classification\_report(y\_test, predicted)}\n")

disp = metrics.plot\_confusion\_matrix(clf, X\_test, y\_test)

disp.figure\_.suptitle("Confusion Matrix")

print(f"Confusion matrix:\n{disp.confusion\_matrix}")

plt.show()