# Import required libraries

from sklearn import datasets

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

from sklearn.preprocessing import StandardScaler

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, confusion\_matrix

import matplotlib.pyplot as plt

# Load MNIST digits dataset

digits = datasets.load\_digits()

# Features and labels

X = digits.data

y = digits.target

# Split into training and testing sets

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

X, y, test\_size=0.2, random\_state=42, stratify=y

)

# Feature scaling

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Train the SVM model

svm\_model = SVC(kernel='rbf', gamma=0.001, C=10)

svm\_model.fit(X\_train\_scaled, y\_train)

# Make predictions

y\_pred = svm\_model.predict(X\_test\_scaled)

# Evaluation

print("Classification Report:\n", classification\_report(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

# Display some predicted digits

plt.figure(figsize=(10, 4))

for index, (image, prediction) in enumerate(zip(X\_test[:8], y\_pred[:8])):

plt.subplot(2, 4, index + 1)

plt.imshow(image.reshape(8, 8), cmap=plt.cm.gray\_r)

plt.title(f"Pred: {prediction}")

plt.axis('off')

plt.tight\_layout()

plt.show()