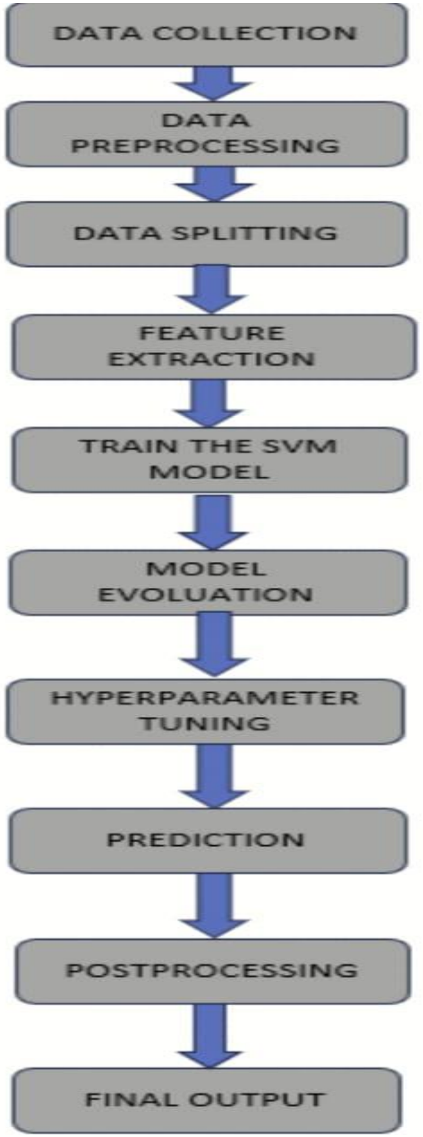
Handwritten Digit Recognition using SVM

Flow Chart



DATA COLLECTION

DATA PREPROCESSING

DATA SPLITTING

FEATURE EXTRACTION

TRAIN THE SVM MODEL

MODEL EVOLUATION

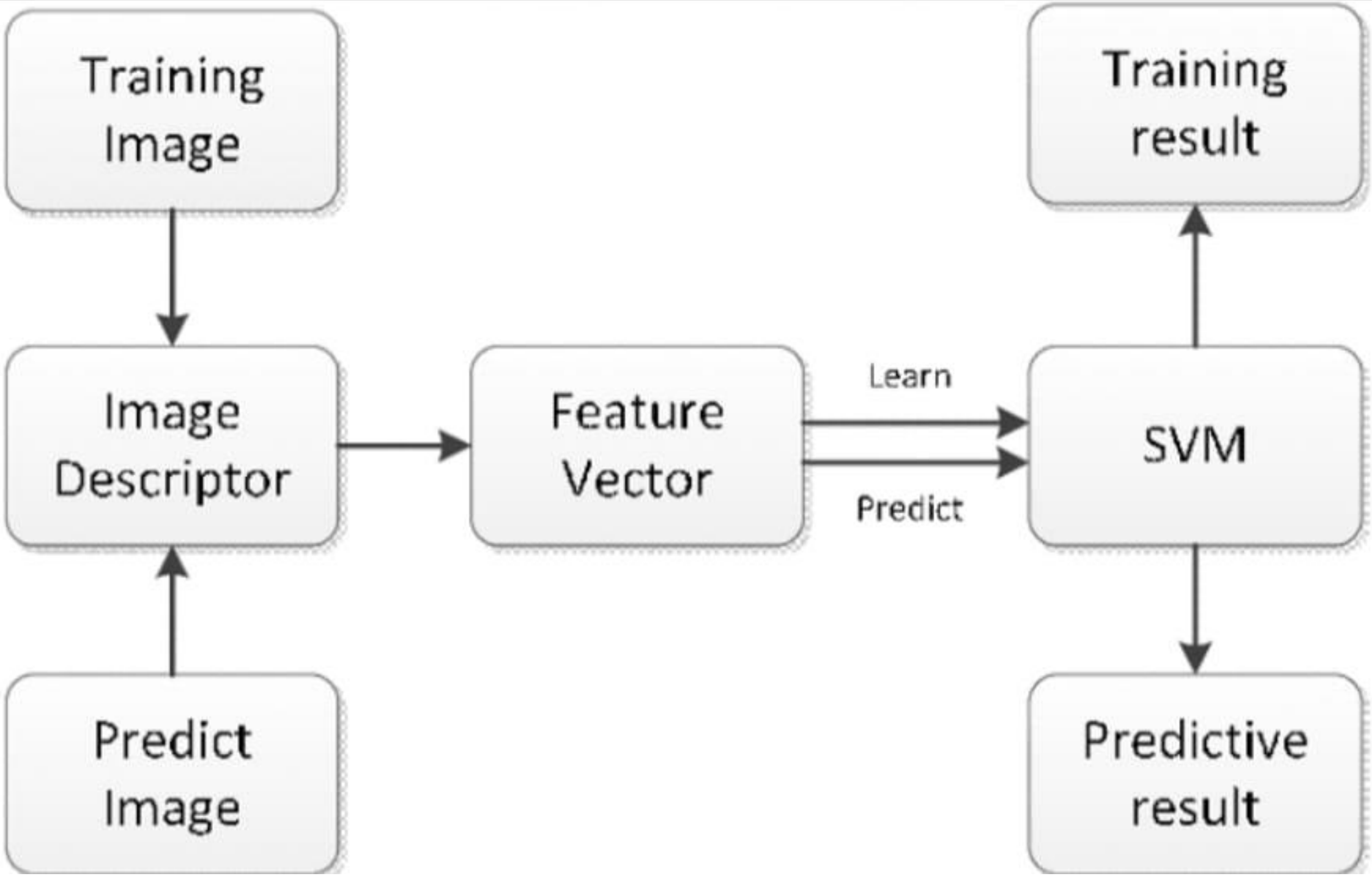
HYPERPARAMETER TUNING

PREDICTION

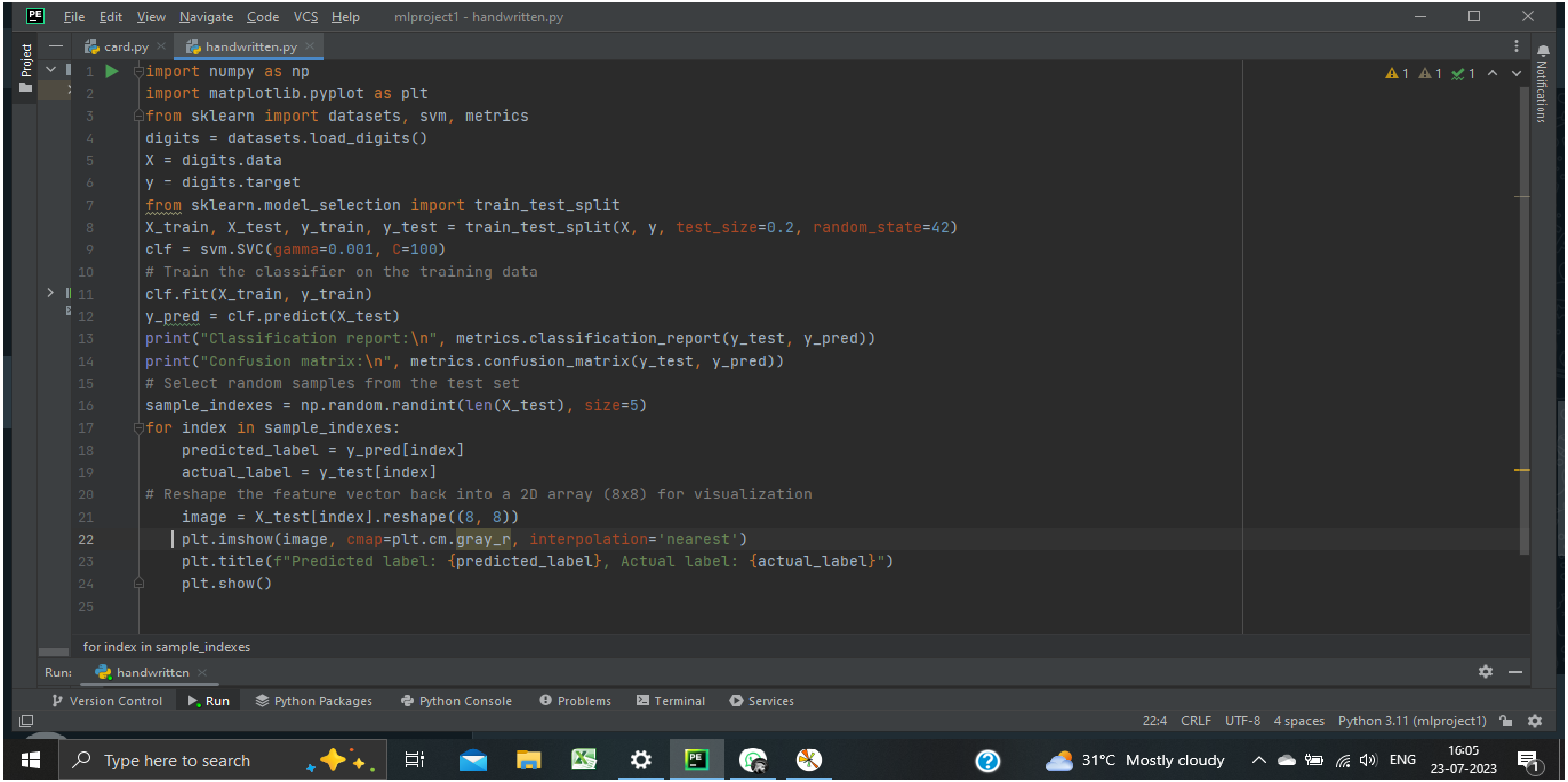
POSTPROCESSING

FINAL OUTPUT

Architecture Diagram

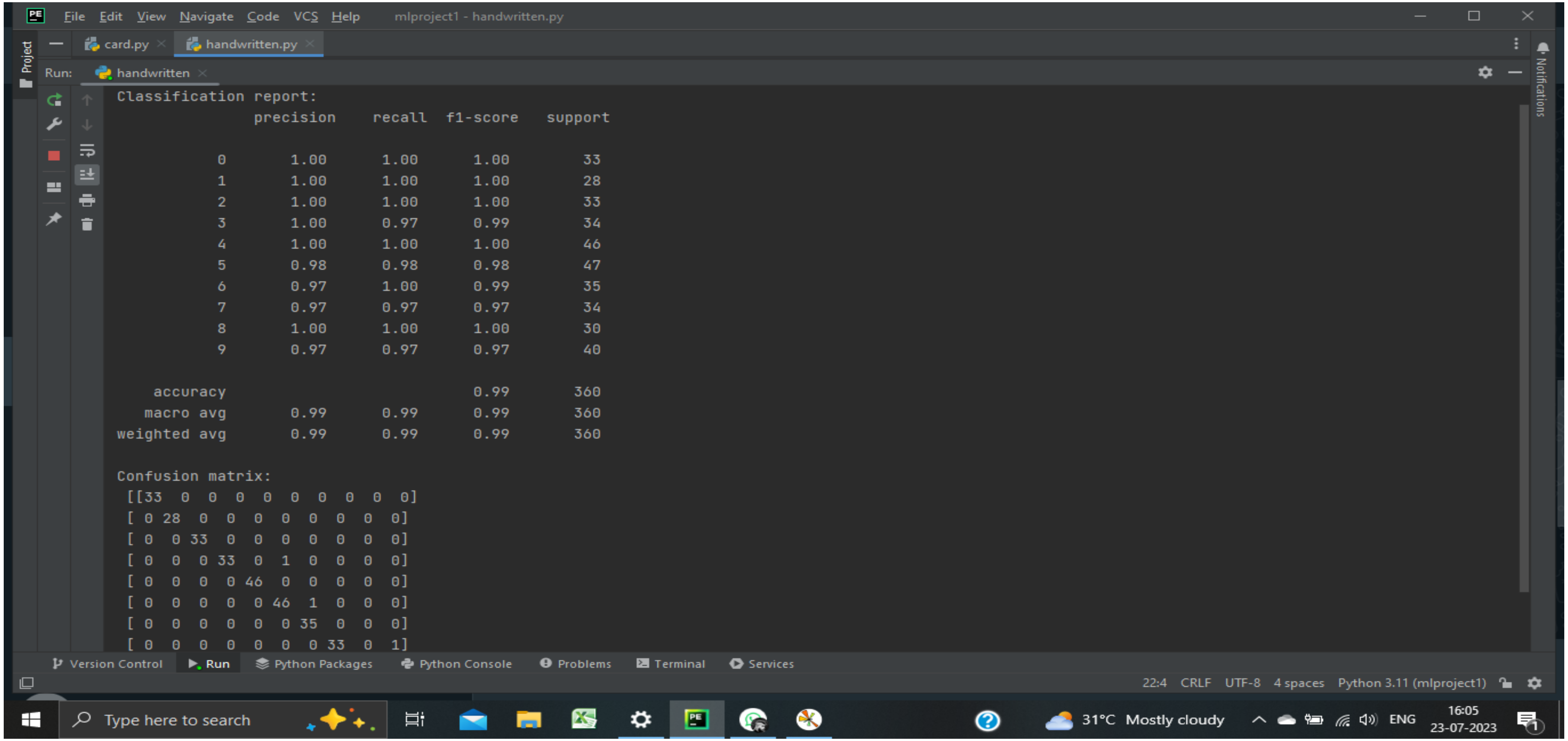


Program Code



```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets, svm, metrics
digits = datasets.load_digits()
X = digits.data
y = digits.target
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
clf = svm.SVC(gamma=0.001, C=100)
# Train the classifier on the training data
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Classification report:\n", metrics.classification_report(y_test, y_pred))
print("Confusion matrix:\n", metrics.confusion_matrix(y_test, y_pred))
# Select random samples from the test set
sample_indexes = np.random.randint(len(X_test), size=5)
for index in sample_indexes:
    predicted_label = y_pred[index]
    actual_label = y_test[index]
# Reshape the feature vector back into a 2D array (8x8) for visualization
    image = X_test[index].reshape((8, 8))
    plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    plt.title(f"Predicted label: {predicted_label}, Actual label: {actual_label}")
    plt.show()
```

Output

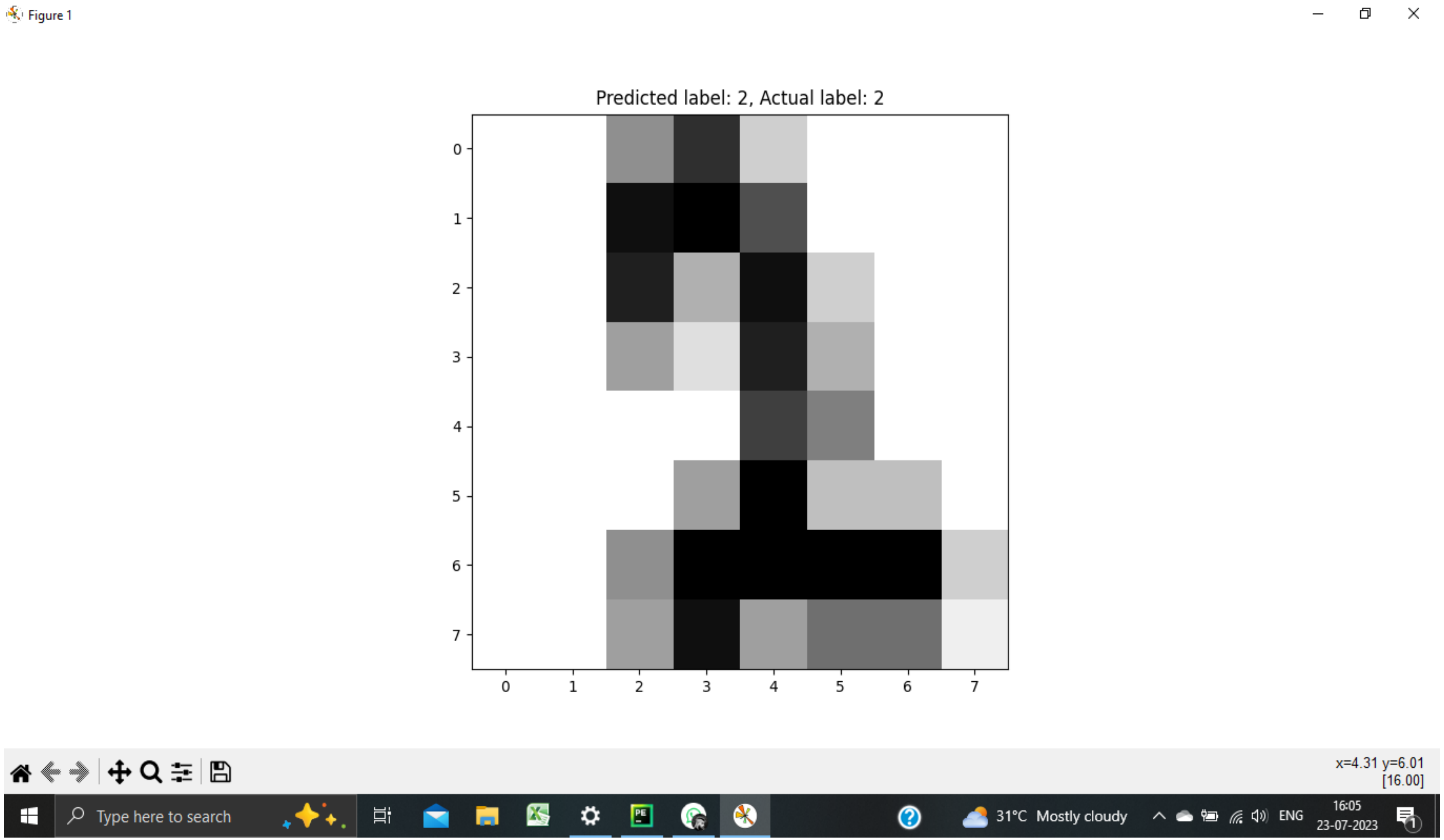


```
Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        33
           1       1.00      1.00      1.00        28
           2       1.00      1.00      1.00        33
           3       1.00      0.97      0.99        34
           4       1.00      1.00      1.00        46
           5       0.98      0.98      0.98        47
           6       0.97      1.00      0.99        35
           7       0.97      0.97      0.97        34
           8       1.00      1.00      1.00        30
           9       0.97      0.97      0.97        40

    accuracy                           0.99       360
   macro avg       0.99      0.99      0.99       360
weighted avg       0.99      0.99      0.99       360

Confusion matrix:
 [[33  0  0  0  0  0  0  0  0  0]
 [ 0 28  0  0  0  0  0  0  0  0]
 [ 0  0 33  0  0  0  0  0  0  0]
 [ 0  0  0 33  0  1  0  0  0  0]
 [ 0  0  0  0 46  0  0  0  0  0]
 [ 0  0  0  0  0 46  1  0  0  0]
 [ 0  0  0  0  0  0 35  0  0  0]
 [ 0  0  0  0  0  0  0 33  0  1]
```



Predicted label: 2, Actual label: 2