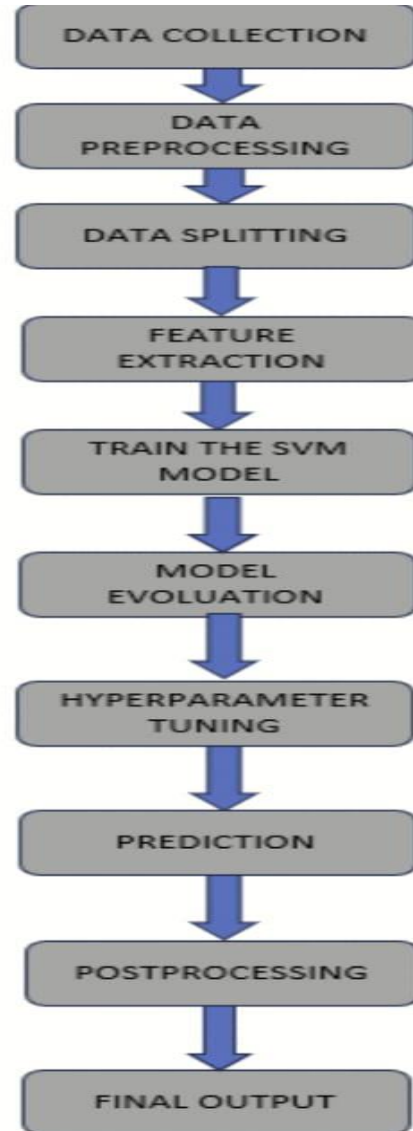
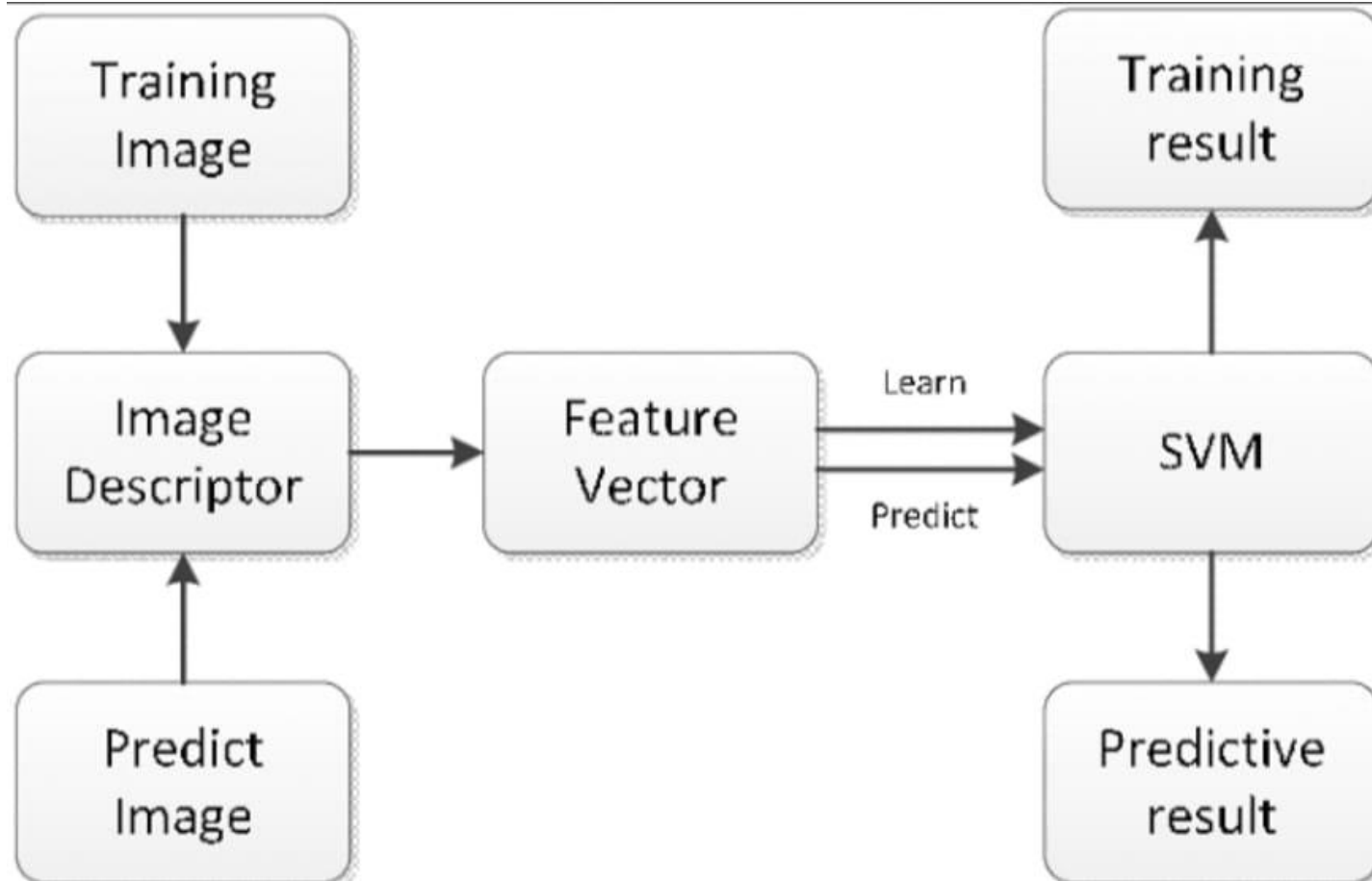


# Handwritten Digit Recognition using SVM

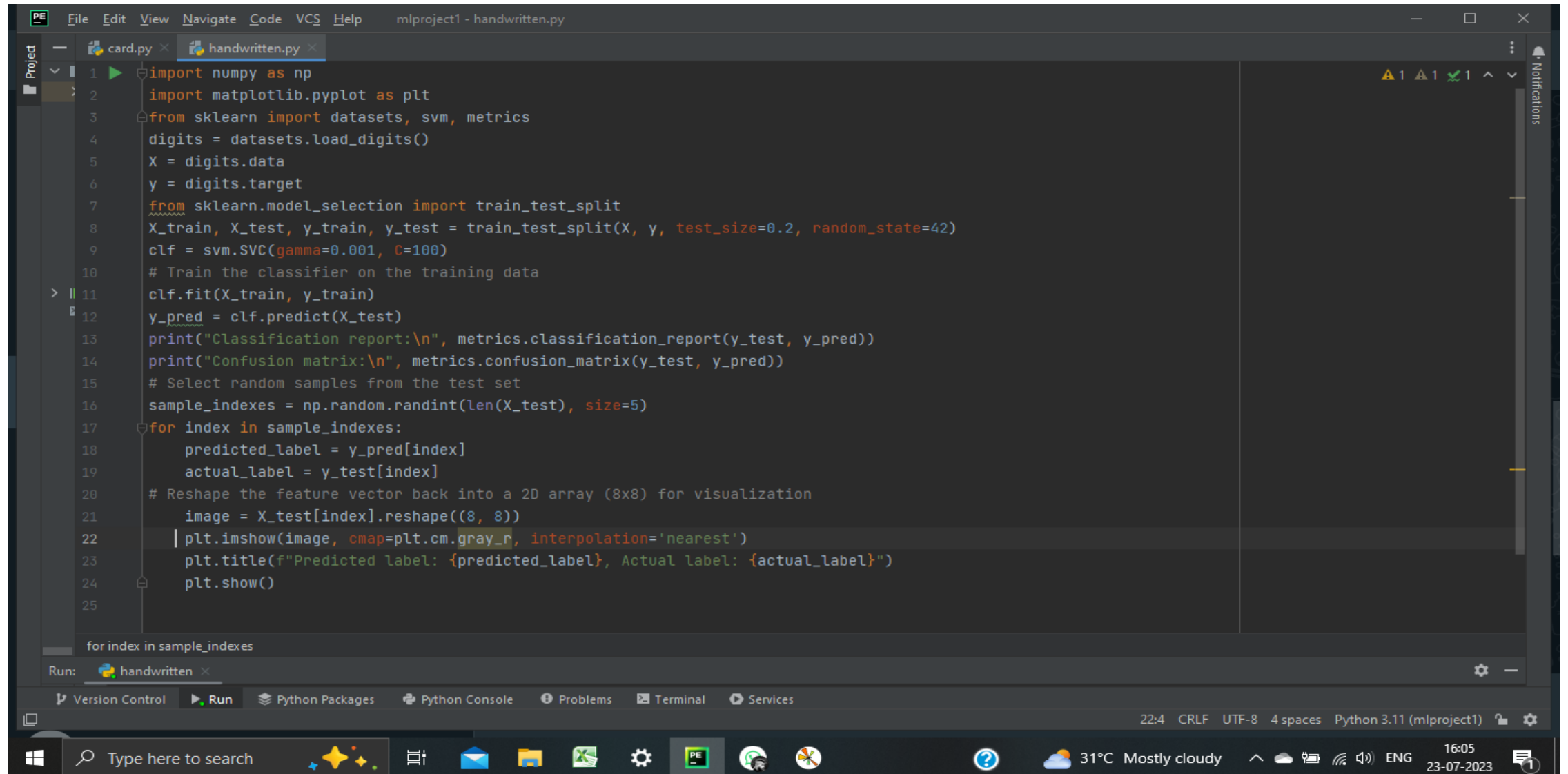
# Flow Chart



# Architecture Diagram



# Program Code



The screenshot shows a code editor window titled "mlproject1 - handwritten.py". The code is a Python script for digit classification using Support Vector Machine (SVM). It imports necessary libraries, loads the digits dataset, splits it into training and testing sets, trains an SVM classifier, and then evaluates its performance. The script also includes a loop to visualize five random samples from the test set, showing the predicted and actual labels for each.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn import datasets, svm, metrics
4 digits = datasets.load_digits()
5 X = digits.data
6 y = digits.target
7 from sklearn.model_selection import train_test_split
8 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
9 clf = svm.SVC(gamma=0.001, C=100)
10 # Train the classifier on the training data
11 clf.fit(X_train, y_train)
12 y_pred = clf.predict(X_test)
13 print("Classification report:\n", metrics.classification_report(y_test, y_pred))
14 print("Confusion matrix:\n", metrics.confusion_matrix(y_test, y_pred))
15 # Select random samples from the test set
16 sample_indexes = np.random.randint(len(X_test), size=5)
17 for index in sample_indexes:
18     predicted_label = y_pred[index]
19     actual_label = y_test[index]
20     # Reshape the feature vector back into a 2D array (8x8) for visualization
21     image = X_test[index].reshape((8, 8))
22     plt.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
23     plt.title(f"Predicted label: {predicted_label}, Actual label: {actual_label}")
24     plt.show()
25
```

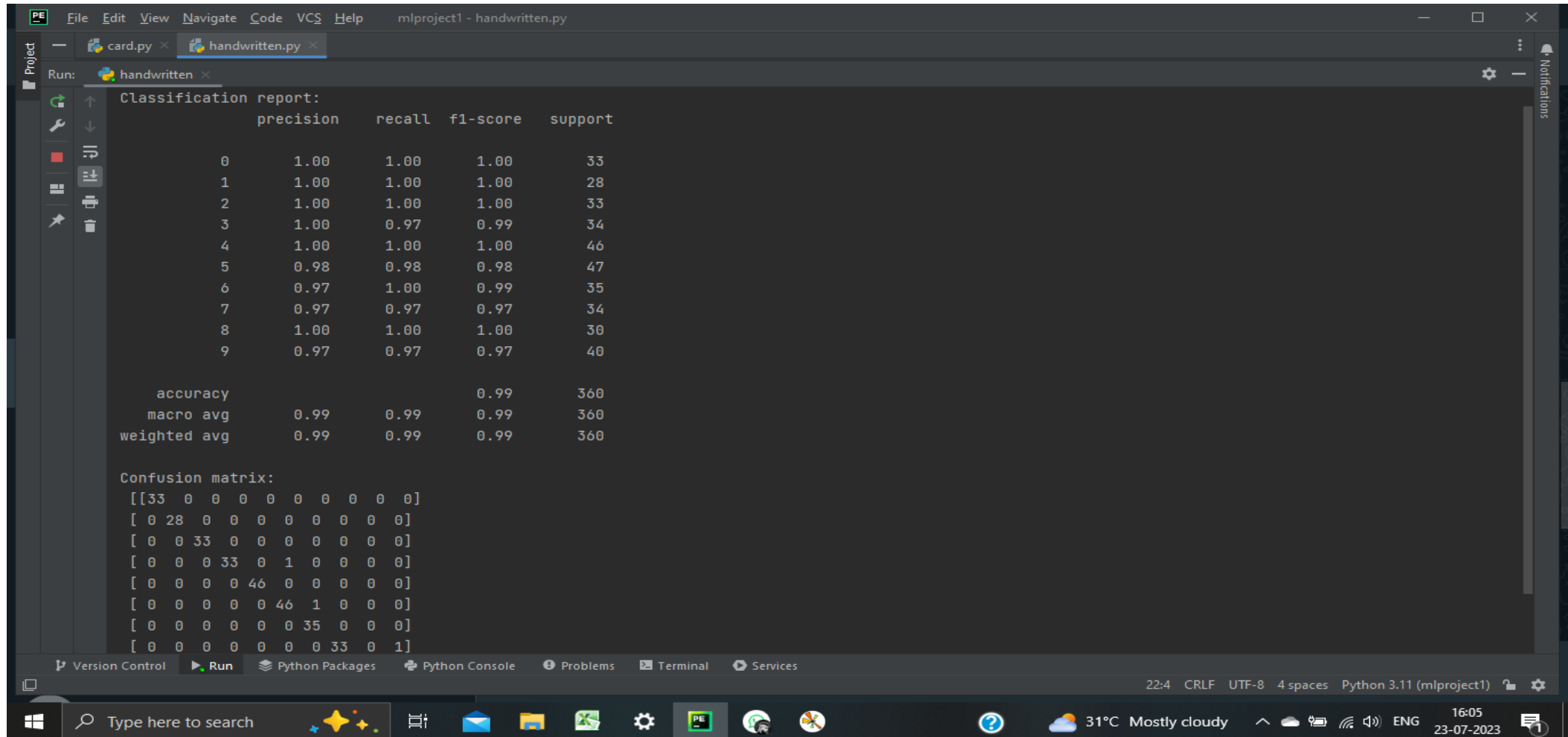
Run: handwritten

Version Control Run Python Packages Python Console Problems Terminal Services

22:4 CRLF UTF-8 4 spaces Python 3.11 (mlproject1)

Type here to search 31°C Mostly cloudy 16:05 23-07-2023

# Output



```
mlproject1 - handwritten.py
card.py x handwritten.py x
Run: handwritten x
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

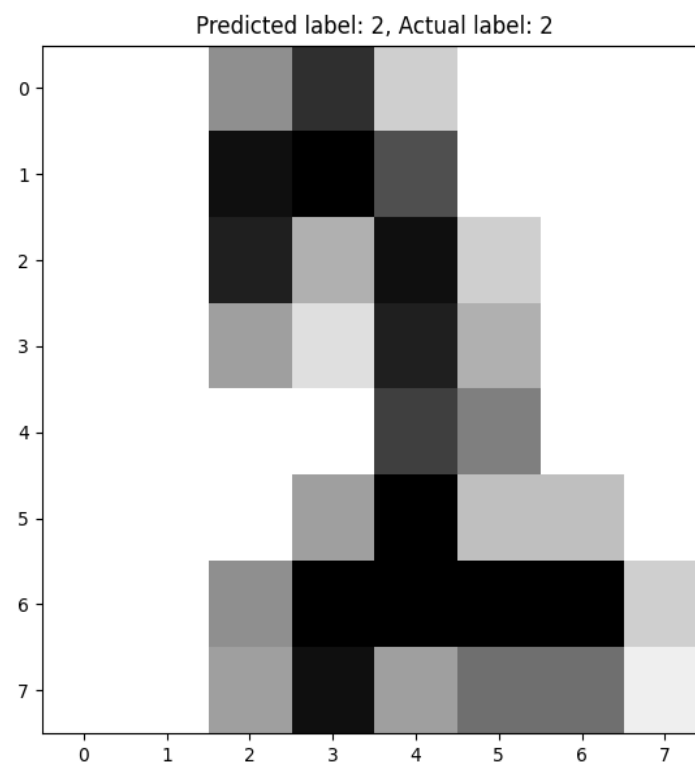
 accuracy          0.99      0.99      0.99      360
 macro avg         0.99      0.99      0.99      360
weighted avg         0.99      0.99      0.99      360

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]
```

22:4 CRLF UTF-8 4 spaces Python 3.11 (mlproject1)

Figure 1

— □ ×



x=4.31 y=6.01  
[16.00]