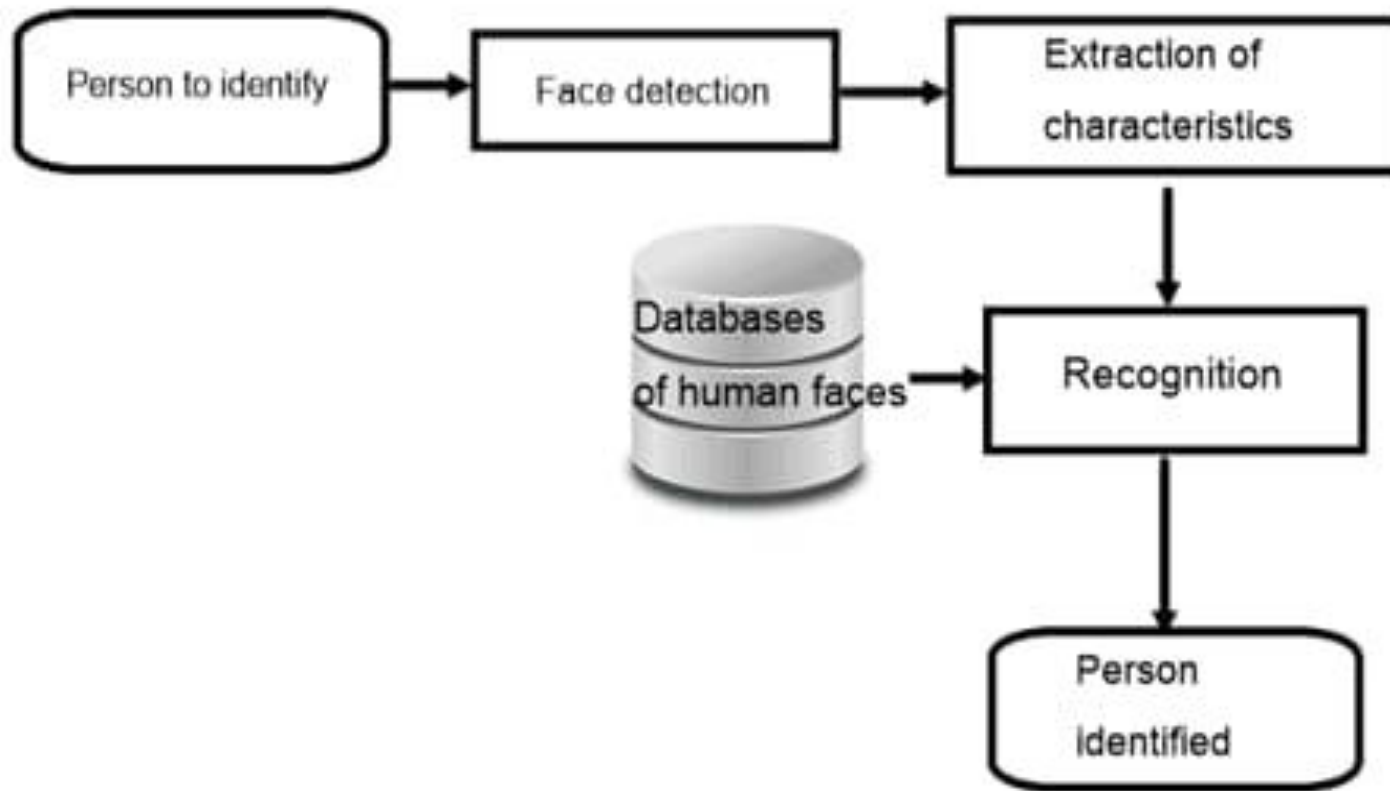


Face Recognition Using SVM

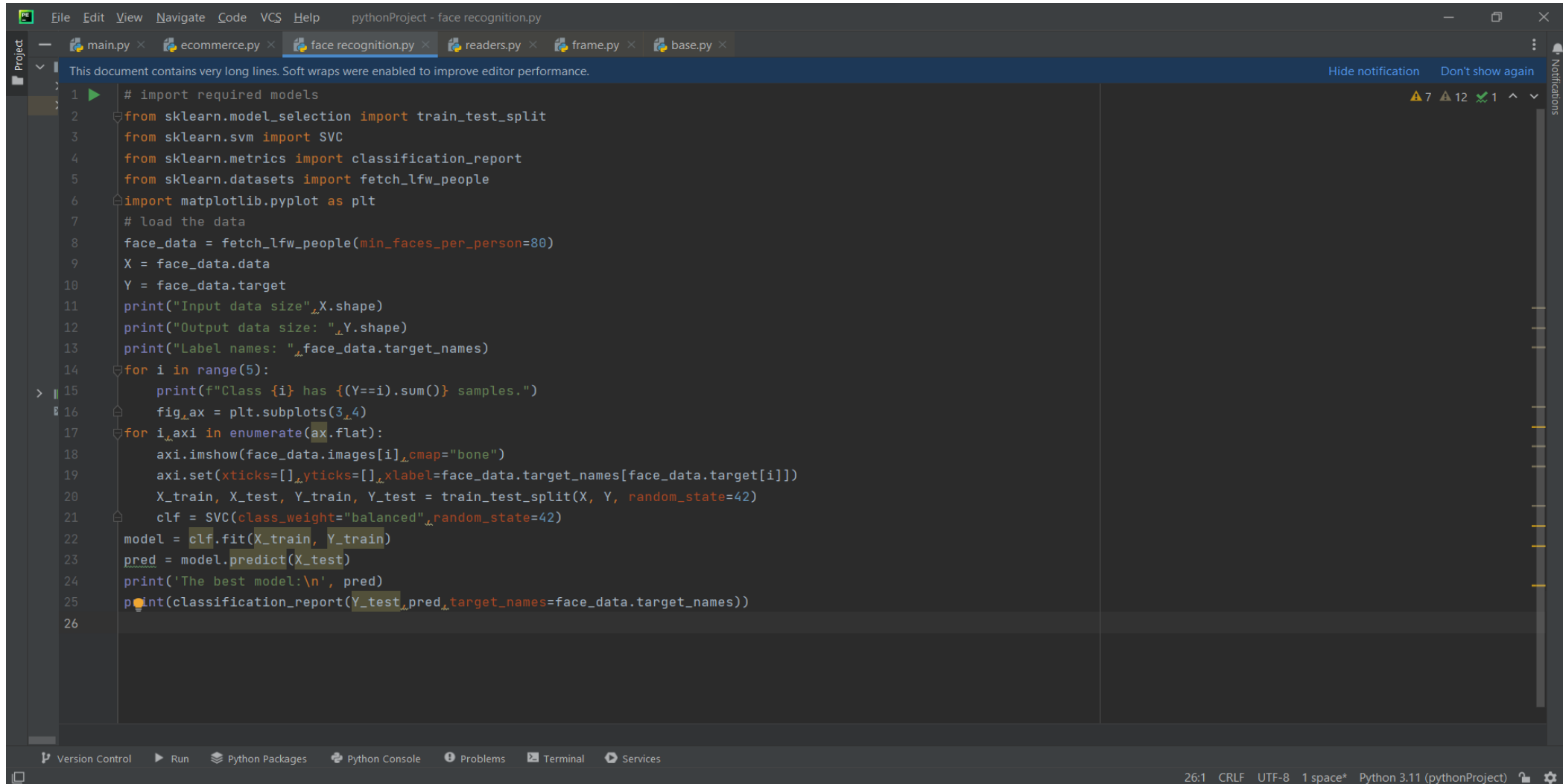
Flow Chart



Architecture Diagram



Program Code



The screenshot shows a code editor window titled 'pythonProject - face_recognition.py'. The editor contains a Python script for face recognition using scikit-learn and matplotlib. The script imports necessary libraries, loads the LFW dataset, and performs a series of steps: printing data shapes, displaying a grid of face images, splitting the data into training and testing sets, training an SVM model, and finally printing a classification report. The code is written in a dark-themed editor with syntax highlighting. The bottom status bar indicates the current position is 26:1, using CRLF line endings, UTF-8 encoding, 1 space for indentation, and Python 3.11.

```
1 # import required models
2 from sklearn.model_selection import train_test_split
3 from sklearn.svm import SVC
4 from sklearn.metrics import classification_report
5 from sklearn.datasets import fetch_lfw_people
6 import matplotlib.pyplot as plt
7 # load the data
8 face_data = fetch_lfw_people(min_faces_per_person=80)
9 X = face_data.data
10 Y = face_data.target
11 print("Input data size", X.shape)
12 print("Output data size: ", Y.shape)
13 print("Label names: ", face_data.target_names)
14 for i in range(5):
15     print(f"Class {i} has {(Y==i).sum()} samples.")
16     fig, ax = plt.subplots(3, 4)
17     for i_axi in enumerate(ax.flat):
18         axi.imshow(face_data.images[i], cmap="bone")
19         axi.set(xticks=[], yticks=[], xlabel=face_data.target_names[face_data.target[i]])
20     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state=42)
21     clf = SVC(class_weight="balanced", random_state=42)
22     model = clf.fit(X_train, Y_train)
23     pred = model.predict(X_test)
24     print('The best model:\n', pred)
25     print(classification_report(Y_test, pred, target_names=face_data.target_names))
26
```

Output

```
Run: face recognition x
Input data size (1140, 2914)
Output data size: (1140,)
Label names: ['Colin Powell' 'Donald Rumsfeld' 'George W Bush' 'Gerhard Schroeder'
'Tony Blair']
Class 0 has 236 samples.
Class 1 has 121 samples.
Class 2 has 530 samples.
Class 3 has 109 samples.
Class 4 has 144 samples.
The best model:
[2 0 2 3 2 2 0 4 2 2 4 2 1 0 2 0 4 4 2 0 0 2 1 4 2 3 4 2 2 1 4 1 1 0 0 0 2
0 0 4 2 2 0 2 0 3 0 2 2 0 2 0 2 0 2 2 2 2 2 0 2 2 2 0 0 3 3 2 0 2 4 2 1 4
2 2 1 4 2 2 2 2 2 2 3 1 2 2 0 1 2 2 2 4 2 0 0 0 2 3 2 3 2 1 2 3 2 1 2 2 2
2 2 0 0 2 2 1 0 1 2 0 2 0 0 1 3 1 2 4 2 1 0 0 4 2 3 4 0 2 2 0 2 1 1 0 3 2
2 2 2 2 2 2 2 3 4 0 0 4 0 2 2 1 3 1 2 2 2 0 1 2 2 2 0 2 2 2 2 4 4 2 4 2 4
2 2 1 2 2 2 2 3 0 1 0 2 2 0 1 0 4 0 3 0 2 0 0 0 2 3 4 4 3 2 1 2 2 2 2 2 3
0 2 1 2 0 2 2 2 2 0 2 2 4 4 2 0 2 3 1 2 2 3 2 0 0 2 0 2 2 3 0 2 2 0 1 0 0
0 3 2 2 4 4 2 1 0 4 3 4 3 2 0 0 1 4 1 0 3 2 0 2 1 2]
precision    recall  f1-score   support

   Colin Powell      0.82      0.86      0.84         64
  Donald Rumsfeld      0.84      0.84      0.84         32
   George W Bush      0.88      0.89      0.88        127
  Gerhard Schroeder      0.77      0.69      0.73         29
     Tony Blair      0.94      0.88      0.91         33

 accuracy              0.86              285
  macro avg           0.85           0.83      0.84        285
  weighted avg        0.86           0.86      0.86        285

Process finished with exit code 0
```