**EXPT NO : 6 A python program for face recognition using SVM**

**DATE: 27.9.24 classifier**

**AIM:**

To write a python program to implement face recognition using SVM classifiers

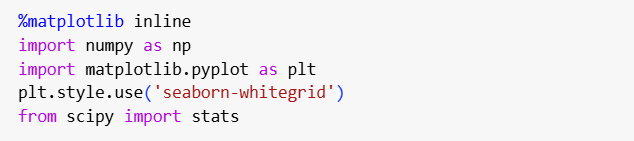
**PROCEDURE:**

Implementing SVM classifiers involve the following steps:

**Step 1: Import Necessary Libraries**

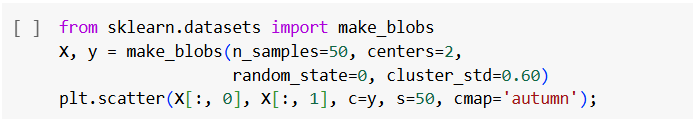
First, import the libraries that are essential for data manipulation, visualization, and

model building.

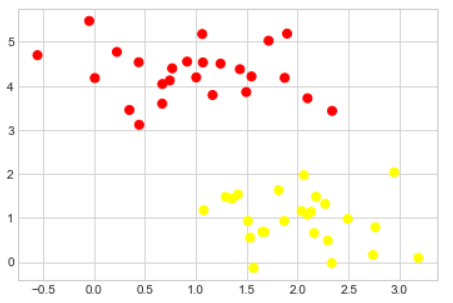


**Step 2: Making blobs**

Consider the simple case of a classification task in which the two classes of points are well separated

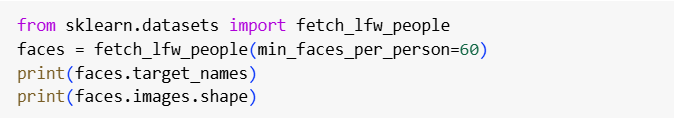


Output:

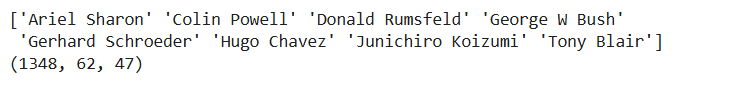


**Step 3: Face recognition**

As an example of support vector machines in action, let's take a look at the facial recognition problem. We will use the Labeled Faces in the Wild dataset, which consists of several thousand collated photos of various public figures. A fetcher for the dataset is built into Scikit-Learn:

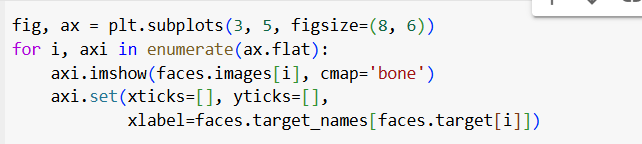


Output:



**Step 4: Plotting**

Let's plot a few of these faces to see what we're working with (see the following figure):

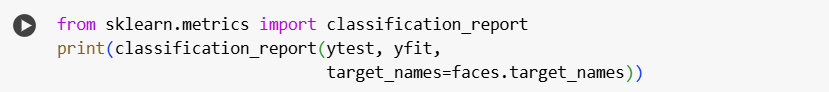


Output:

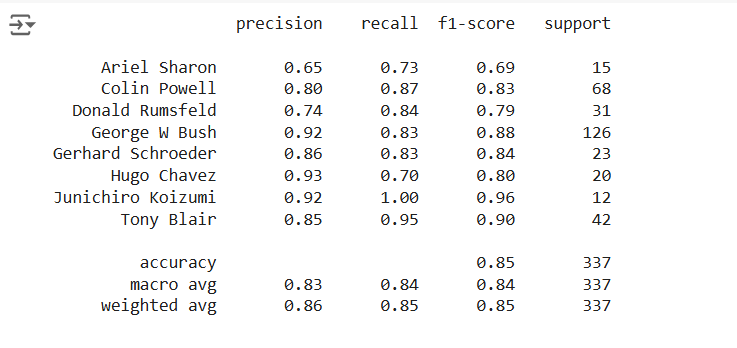


**Step 5: Classification report**

Out of this small sample, our optimal estimator mislabeled only a single face (Bush’s face in the bottom row was mislabeled as Blair). We can get a better sense of our estimator's performance using the classification report, which lists recovery statistics label by label:

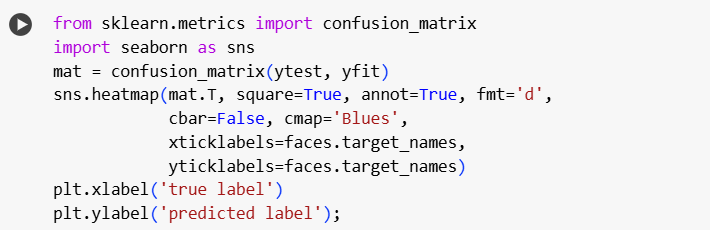


Output:

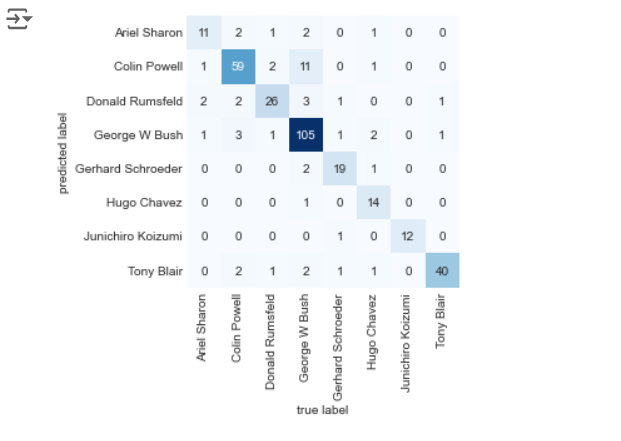


**Step 6: Visualization**

We might also display the confusion matrix between these classes (see the following figure):



Output:



**RESULT:**

This step-by-step process will help us to implement SVM classifiers and analyze performance