**Project7:**

#import fetch\_lfw\_people from sklearn.datasets

from sklearn.datasets import fetch\_lfw\_people

# import the package pandas as pd

import pandas as pd

#import the package numpy as np

import numpy as np

# import PCA from sklearn.decomposition

from sklearn.decomposition import PCA

# import SVC from sklearn.svm

from sklearn.svm import SVC

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

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

#import classification\_report from sklearn.metrics

from sklearn.metrics import classification\_report

#load the face dataset

faces = fetch\_lfw\_people(min\_faces\_per\_person=60)

n\_samples, h, w = faces.images.shape

#printing the face names

print(faces.target\_names)

#printing the face image

print(faces.images.shape)

#splitting the whole data set to the training dataset and testing data set.

X = faces.data

n\_features = faces.data.shape[1]

y = faces.target

target\_names = faces.target\_names

n\_classes = target\_names.shape[0]

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,random\_state=1)

#applying PCA to the original dataset for dimesion reduction

pca = PCA(n\_components=200)

pca.fit(X\_train)

X\_train\_pca = pca.transform(X\_train)

X\_test\_pca = pca.transform(X\_test)

# using the training data set to build kernel support vector classifier

clf = SVC(kernel='linear')

clf = clf.fit(X\_train\_pca,y\_train)

y\_pred = clf.predict(X\_test\_pca)

#printing the evaluate model

print(classification\_report(y\_test,y\_pred,target\_names=faces.target\_names))

**Output:**



