#importing libraries

import warnings

warnings.simplefilter(action="ignore", category=FutureWarning)

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from matplotlib import cm

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

from sklearn.decomposition import PCA

from sklearn.svm import SVC

train\_loc="E:\Datasets\MNIST\Train.csv"

test\_loc="E:\Datasets\MNIST\Test.csv"

train\_label="E:\Datasets\MNIST\TrainLab.csv"

train=pd.read\_csv(train\_loc)

X=train.as\_matrix()

train\_size= X.shape[0]

label=pd.read\_csv(train\_label)

y=label.values

test=pd.read\_csv(test\_loc).as\_matrix()

test\_size=test.shape[0]

#visualising random numbers

"""img\_idx=np.random.randint(test\_size, size=6)

fig=plt.figure()

for i in range(6):

instance=test[img\_idx[i]]

assert(len(instance)==784)

img=instance.reshape(28, 28)

plt.subplot(230+i+1)

plt.title(img\_idx[i]+1)

plt.axis("off")

plt.imshow(img, cmap=cm.gray)"""

#train\_test split

X\_train, X\_test, y\_train, y\_test= train\_test\_split(X, y, train\_size=0.7, random\_state=50)

#using pca

pca=PCA(n\_components=33, whiten=True)

X\_train=pca.fit\_transform(X\_train)

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

test= pca.transform(test)

svc=SVC(C=10, kernel="poly", degree="2", gamma="scale", class\_weight="balanced", random\_state=50)

svc.fit(X\_train, y\_train)

score=svc.score(X\_test, y\_test)

print("Score {0:.4f}".format(score))

labels= svc.predict(test)