

```

import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import fetch_mldata
mnist = fetch_mldata('MNIST original')

X = mnist.data
y = mnist.target

#plt.title("The 1st image is a {label}'.format(label = int(y[1])))
#plt.imshow(X[1].reshape((28,28)), cmap = 'gray')
#plt.show()

X4 = X[y==4, :]
X9 = X[y==9, :]
y4 = y[y==4]
y9 = y[y==9]

X4_design = X4[0:4000, :]
y4_design = y4[0:4000]
X9_design = X9[0:4000, :]
y9_design = y9[0:4000]

X_fit = np.concatenate((X4_design[0:2000, :], X9_design[0:2000, :]), axis=0)
y_fit = np.concatenate((y4_design[0:2000], y9_design[0:2000]), axis=0)
X_hold = np.concatenate((X4_design[2000:-1, :], X9_design[2000:-1, :]), axis=0)
y_hold = np.concatenate((y4_design[2000:-1], y9_design[2000:-1]), axis=0)
X_test = np.concatenate((X4[4000:-1, :], X9[4000:-1, :]), axis=0)
y_test = np.concatenate((y4[4000:-1], y9[4000:-1]), axis=0)

from sklearn import svm
clf = svm.SVC(C=1000000, kernel='rbf', gamma = 0.000001)
clf.fit(X_fit, y_fit)
Pe = 1 - clf.score(X_hold, y_hold)
print Pe

clf.fit(np.concatenate((X_fit, X_hold), axis=0), np.concatenate((y_fit, y_hold), axis=0))
Pe = 1 - clf.score(X_test, y_test)
print Pe, clf.support_vectors_.shape

# I used the same code, and only changed the parameters

```