

```

import numpy
import numpy.linalg
import scipy
import scipy.linalg
import cvxopt
import cvxopt.solvers

def svdd_fit_predict(self,X):
    N = len(X)
    XX = X.dot(X.T)
    X2 = (X**2).sum(axis=1)

    P = 2*cvxopt.matrix(XX)
    q = - cvxopt.matrix(X2)
    G = cvxopt.matrix(numpy.concatenate([-numpy.identity(N),numpy.identity(N)]))
    h = cvxopt.matrix(numpy.concatenate([numpy.zeros([N]),self.C*numpy.ones([N])]))
    A = cvxopt.matrix(numpy.ones([1,N]))
    b = cvxopt.matrix(numpy.ones([1]))

    self.X = X
    self.alpha = numpy.array(cvxopt.solvers.qp(P,q,G,h,A,b)['x'][:,0])

    sv = numpy.argmin(numpy.abs(self.alpha - self.C/2))

    self.c = self.X.T.dot(self.alpha)
    self.S = ((X[sv]-self.c)**2).sum()

    return ((X-self.c)**2).sum(axis=1) > self.S

def phi(X,omega,tau):
    return numpy.cos(X.dot(omega)+tau)

```