1.

(a) Xmean = np.mean(X,0) = [1.5 2.5 3]

(b) Xs = X – Xmean

Q = \* np.dot(Xs.T, Xs) =

(c)w, v = np.linalg.eig(Q)

w = [3.56166464, 1.1733803 , 0.01495506]

v = [[-0.45056922, -0.66677184, -0.59363515],

       [ 0.19247228, -0.72187235,  0.66472154],

       [ 0.87174641, -0.18524476, -0.45358856]]

(d) coefficient = np.dot(Xs, v)

[[-2.95145599, -0.17610969, -0.0888421 ],

       [ 1.37104342, -1.69406159,  0.0198819 ],

       [-0.30682473,  0.78694448,  0.19125108],

       [ 1.8872373 ,  1.0832268 , -0.12229089]])

(e)  Xhat = X\_mean + np.dot(coefficient, v.T)

[3.0000000e+00, 2.0000000e+00, 1.0000000e+00],

       [2.0000000e+00, 4.0000000e+00, 5.0000000e+00],

       [1.0000000e+00, 2.0000000e+00, 3.0000000e+00],

       [8.8817842e-16, 2.0000000e+00, 5.0000000e+00]]

(f) Xhat = X\_mean + np.dot(coefficient[:,:2], v[:,:2].T)

[[ 2.94726021,  2.05905526,  0.95970224],

       [ 2.0118026 ,  3.98678407,  5.0090182 ],

       [ 1.11353336,  1.87287129,  3.0867493 ],

       [-0.07259617,  2.08128939,  4.94453025]]

(g) np.sum(coefficient[:,2]\*\*2) == np.sum( np.sum((X-Xhat)\*\*2, 1))

2.

(a)

(b) = (2,3,0)

(c) approximation error = 4

3.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

mu, V = PCA(X\_train)

ncomp = V.shape[0]

acc = []

for n in range(n\_components):

clf = Classifier()

Z = V[:n]

clf.fit(Z,y)

yhat = clf.predit(X\_test)

acc.append(np.mean((y\_test - yhat)\*\*2))

index = np.argmax(acc)

opt\_num = index + 1

4.

Y = reshape(X, (1000, 784))

pca = PCA(n\_components = 5)

pca.fit(Y[:500,:])

Z = pca.transform(Y[500:,:])

yhat = pca.inverse\_transform(Z)

5.

Xmean = np.mean(X,0)

Xs = X - Xmean[None,:]

U,S, Vtr = svd(Xs, full\_matrices = False)

lam = S\*\*2

pov = np.cumsum(lam) / np.sum(lam)

numbers = np.where(pov > 0.9)[0][0] + 1

Xhat = (U[:,:numbers] \* S[None, :numbers]).dot(Vtr[:numbers,:]) + Xmean