Q3.

Entropy criterion:

Suppose class1=A, class0=B

The training examples have labels: 4A & 2B

X1:

Yes: 3A

No: 1A & 2B

X2:

Yes: 2A & 2B

No: 2A

X3:

Yes: 3A & 1B

No: 1A & 1B

X4:

Yes: 2A & 2B

No: 2A

Because we want to minimize Hafter to find the best split, X1 will be chosen for the root.

Gini criterion:

X1:

Yes: 3A

No: 1A & 2B

X2:

Yes: 2A & 2B

No: 2A

X3:

Yes: 3A & 1B

No: 1A & 1B

X4:

Yes: 2A & 2B

No: 2A

Because Gini index is to calculate how frequently a randomly chosen element will be wrongly identified and so that we want to minimize Gini indexto find the best split, X1 will be chosen for the root.

Misclassification criterion:

X1:

Yes: 3A

No: 1A & 2B

X2:

Yes: 2A & 2B

No: 2A

X3:

Yes: 3A & 1B

No: 1A & 1B

X4:

Yes: 2A & 2B

No: 2A

Because we want to minimize number of points that are incorrectly classified, X1 will be chosen so that we minimize J(S).

4.

=5

=-3

If g1 is greater than g2, then we assign it to C1.

5 > -3

8 > 0

We can replace these two functions by a single discriminant function:

8

7.

a.

S1=

S2=

S3=

B=

b.

S1,3 =

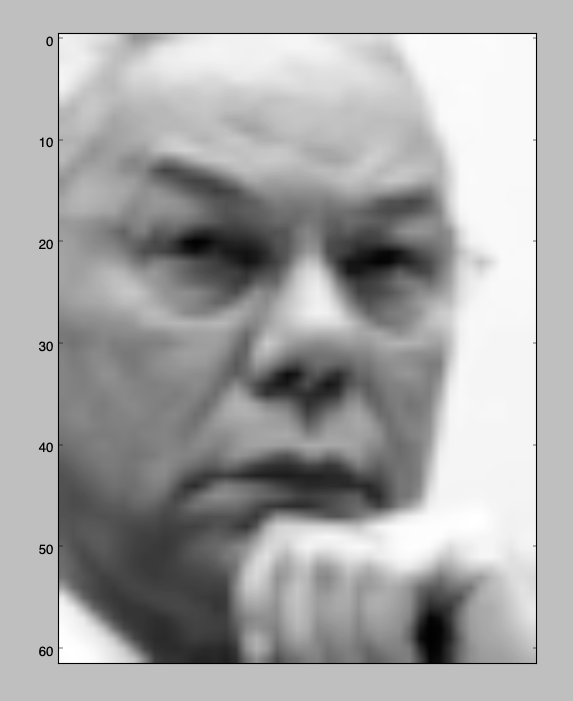
c.

12.9788

d.

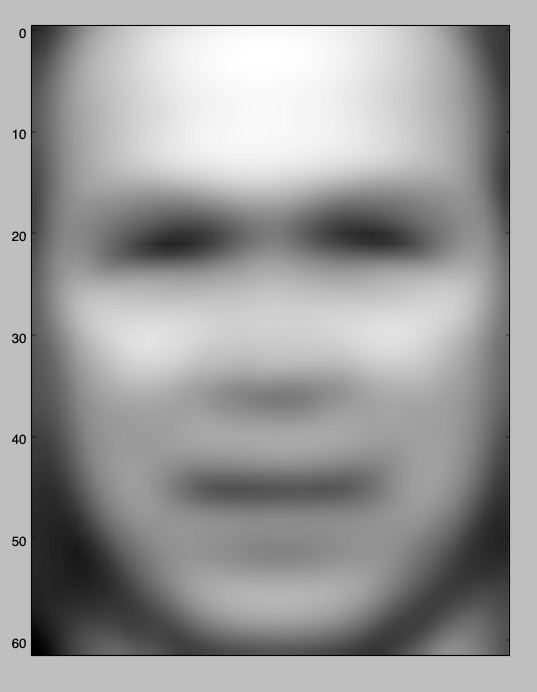
5.

a.



plt\_face(fea[3])

b.



plt\_face(np.mean(fea,axis=0))

c.

pca = skd.PCA(n\_components = 5)

skd.PCA.fit(pca,fea)

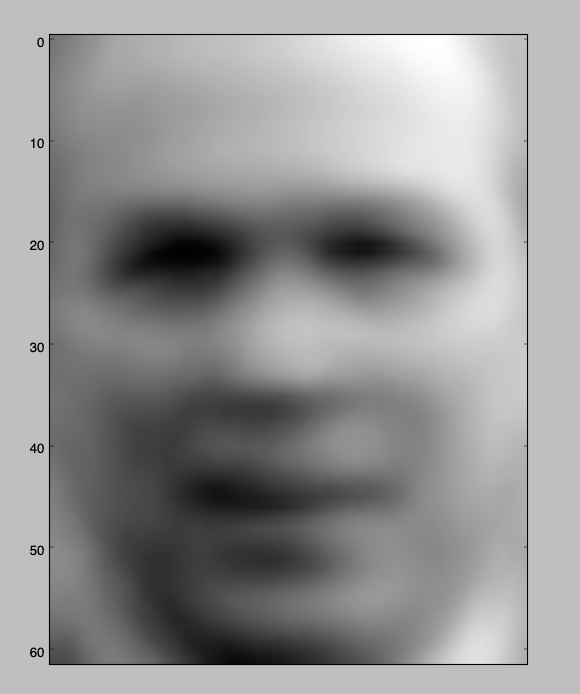
W1 = pca.components\_

W = W1.transpose()

Z = pca.transform(fea)

print Z[3]

d.



pca = skd.PCA(n\_components = 5)

skd.PCA.fit(pca,fea)

W1 = pca.components\_

W = W1.transpose()

Z = pca.transform(fea)

A=np.dot(W, Z[3])+np.mean(fea,axis=0)

plt\_face(A)

plt.show()



pca = skd.PCA(n\_components = 50)

skd.PCA.fit(pca,fea)

W1 = pca.components\_

W = W1.transpose()

Z = pca.transform(fea)

B=np.dot(W, Z[3])+np.mean(fea,axis=0)

plt\_face(B)

plt.show()