ECE GY-6143 ML Homework 4 Solution

Yogya Sharma
(ys5250@nyu.edu)

1. The idea behind Principal Component Analysis is that the data is represented as mean + steps in one direction instead of representing it in all its dimensions. More generally a subset is chosen from all the dimensions.

The optimal directions towards which these steps will be taken in will be along eigenvectors of covariance and the eigen vectors corresponding to the highest eigenvalues will be chosen.

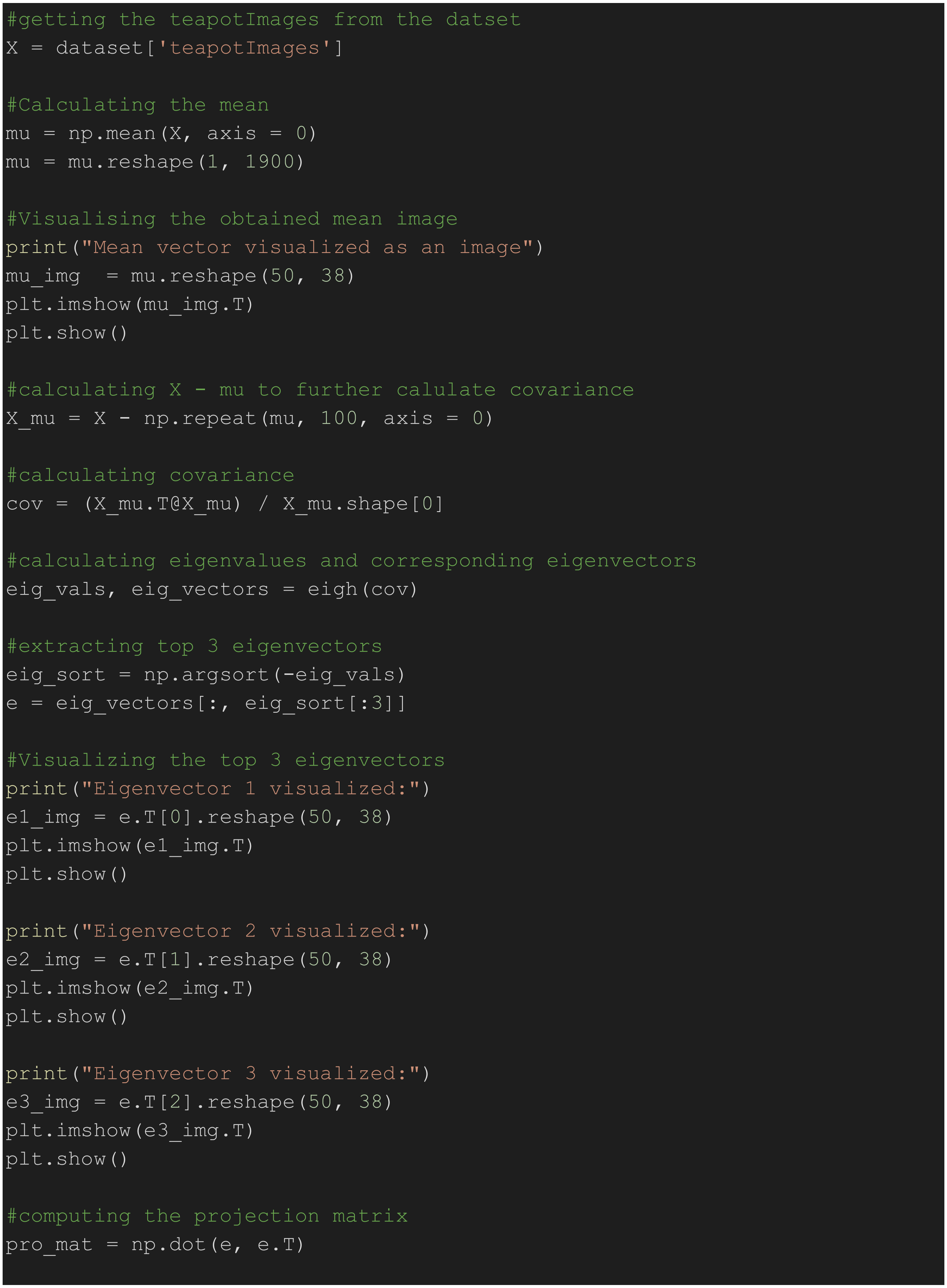
In this question we were supposed to choose the top 3 eigenvectors along with the mean to reconstruct images with least square errors.

In the code below these are the steps:

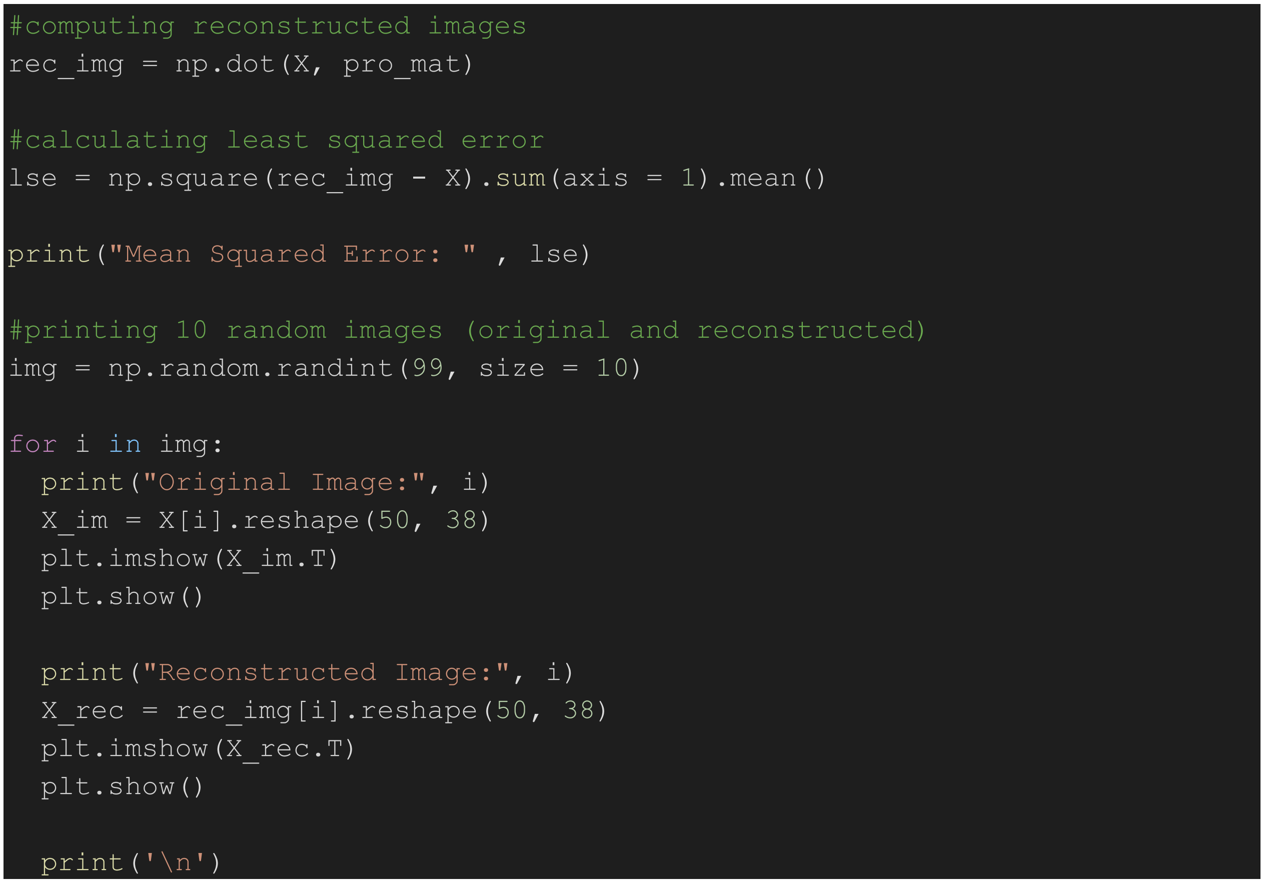
1. Calculated the mean for each feature (1900) along all 100 images given in the dataset. This mean is a [1 X 1900] vector.
2. Computed the covariance matrix using the data and the mean calculated in the previous step.
3. Using the covariance matrix, computed the eigenvalues and the corresponding eigenvectors.
4. The function eigh in numpy.linalg returns a sorted list of eigenvalues, of which I chose the top 3 principal components. Principal components are the new feature values that are constructed as linear combinations of initial feature values. This is to ensure least correlation, to reduce the redundancy in the data. Majority of the information is present in the 1st principal component and remaining in the next one, and so on. As, higher the eigenvalue, higher the variance.
5. Then compute the projection matrix using these principal components, which maps the all the data points to a low-dimensional eigen space.
6. Then reconstruct the images with reduced dimensionality by computing the matrix product of the data matrix X and the projection matrix
7. Calculate the least square error.

This way we have found the optimal directions that explain the data in the best way, and we reorient the data to the axes described by the principal components. This results in the data with reduced dimensions.

```
import scipy.io
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import eigh
from sklearn.model_selection import train_test_split

#importing dataset
dataset = scipy.io.loadmat('teapots.mat')
```

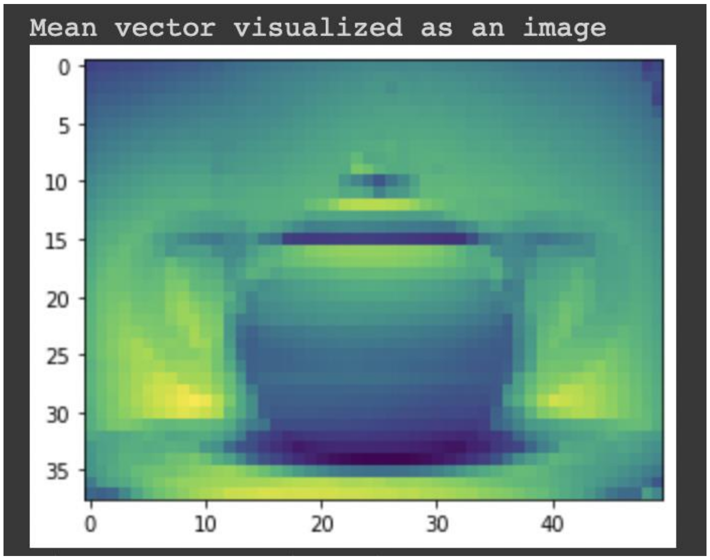


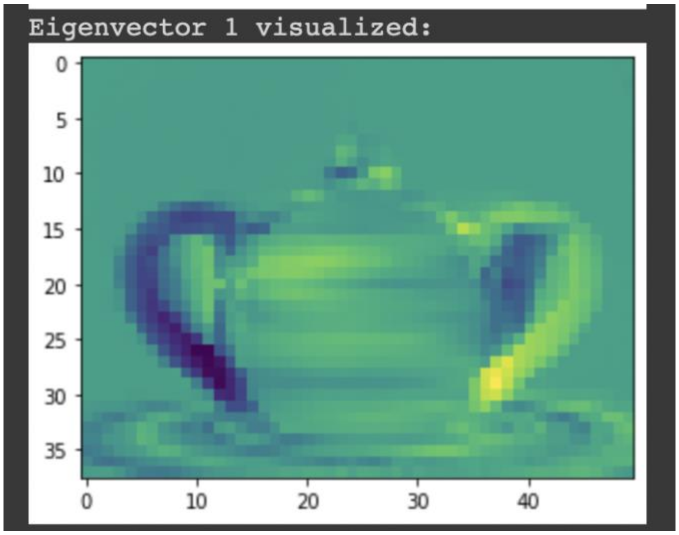
```
#getting the teapotImages from the datset
X = dataset['teapotImages']

#Calculating the mean
mu = np.mean(X, axis = 0)
mu = mu.reshape(1, 1900)

#Visualising the obtained mean image
print("Mean vector visualized as an image")
mu_img  = mu.reshape(50, 38)
plt.imshow(mu_img.T)
plt.show()

#calculating X - mu to further calulate covariance
X_mu = X - np.repeat(mu, 100, axis = 0)

#calculating covariance
cov = (X_mu.T@X_mu) / X_mu.shape[0]

#calculating eigenvalues and corresponding eigenvectors
eig_vals, eig_vectors = eigh(cov)

#extracting top 3 eigenvectors
eig_sort = np.argsort(-eig_vals)
e = eig_vectors[:, eig_sort[:3]]

#Visualizing the top 3 eigenvectors
print("Eigenvector 1 visualized:")
e1_img = e.T[0].reshape(50, 38)
plt.imshow(e1_img.T)
plt.show()

print("Eigenvector 2 visualized:")
e2_img = e.T[1].reshape(50, 38)
plt.imshow(e2_img.T)
plt.show()

print("Eigenvector 3 visualized:")
e3_img = e.T[2].reshape(50, 38)
plt.imshow(e3_img.T)
plt.show()

#computing the projection matrix
pro_mat = np.dot(e, e.T)
```



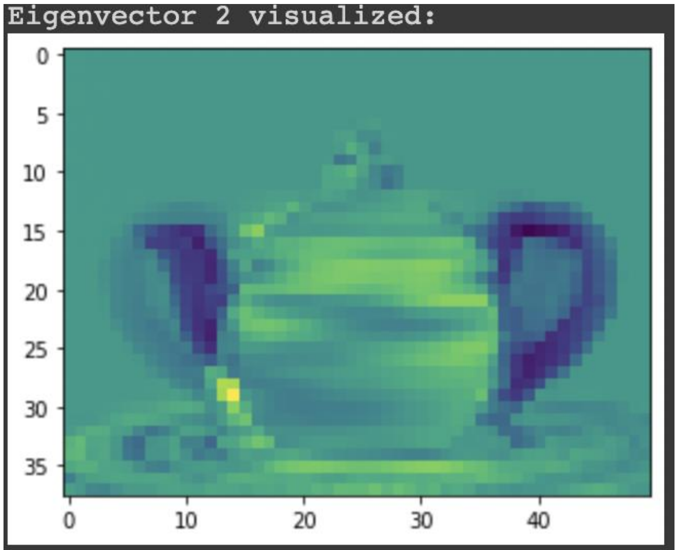
```
#computing reconstructed images
rec_img = np.dot(X, pro_mat)

#calculating least squared error
lse = np.square(rec_img - X).sum(axis = 1).mean()

print("Mean Squared Error: " , lse)

#printing 10 random images (original and reconstructed)
img = np.random.randint(99, size = 10)

for i in img:
  print("Original Image:", i)
  X_im = X[i].reshape(50, 38)
  plt.imshow(X_im.T)
  plt.show()

  print("Reconstructed Image:", i)
  X_rec = rec_img[i].reshape(50, 38)
  plt.imshow(X_rec.T)
  plt.show()

  print('\n')
```

Outputs:

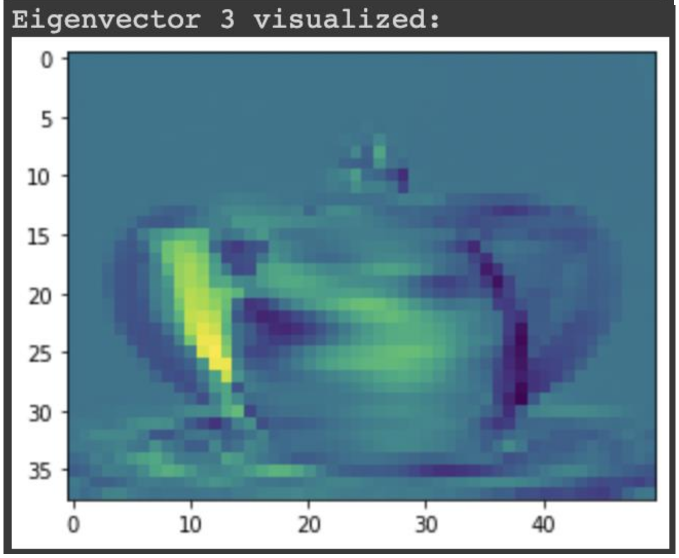




```
Eigenvector 1 visualized:
```

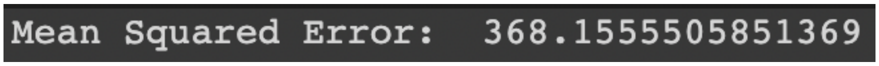


```
Eigenvector 2 visualized:
```



```
Eigenvector 3 visualized:
```

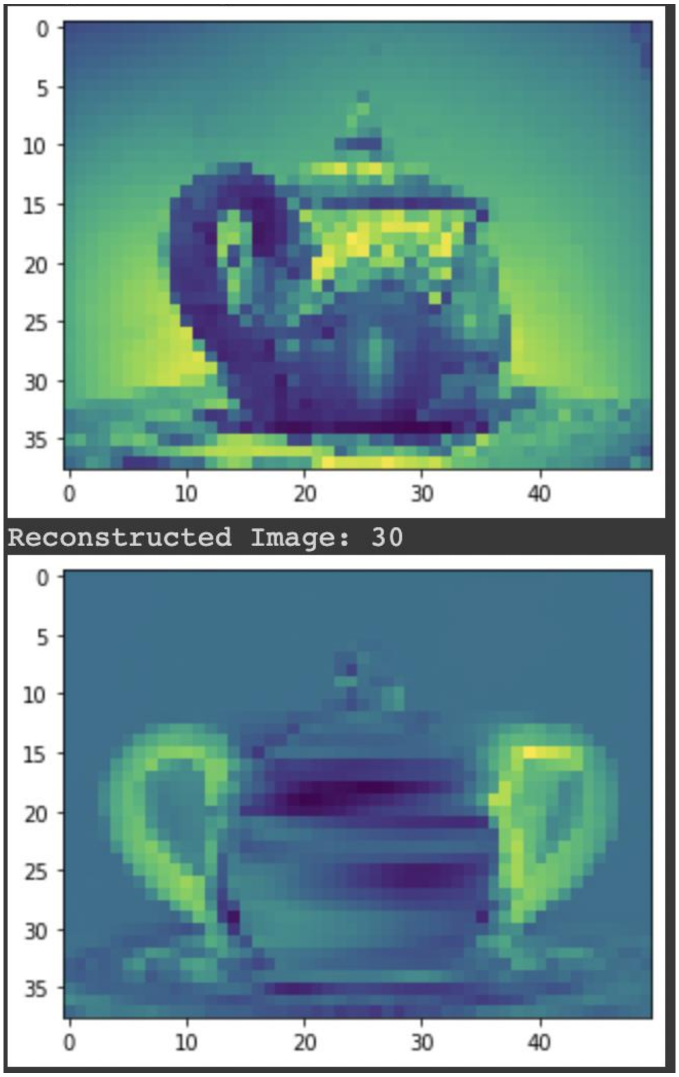
Least Squared Error:



```
Mean Squared Error:  368.1555505851369
```

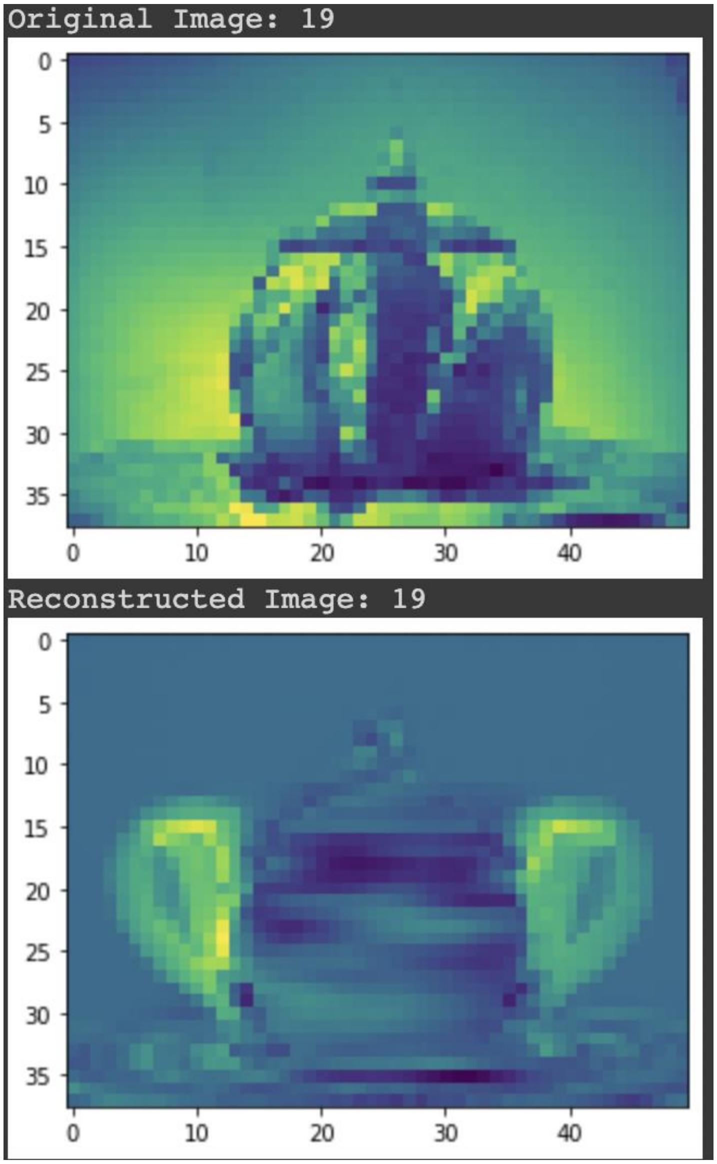
10 images original and reconstructed:

1.



```
Reconstructed Image: 30
```

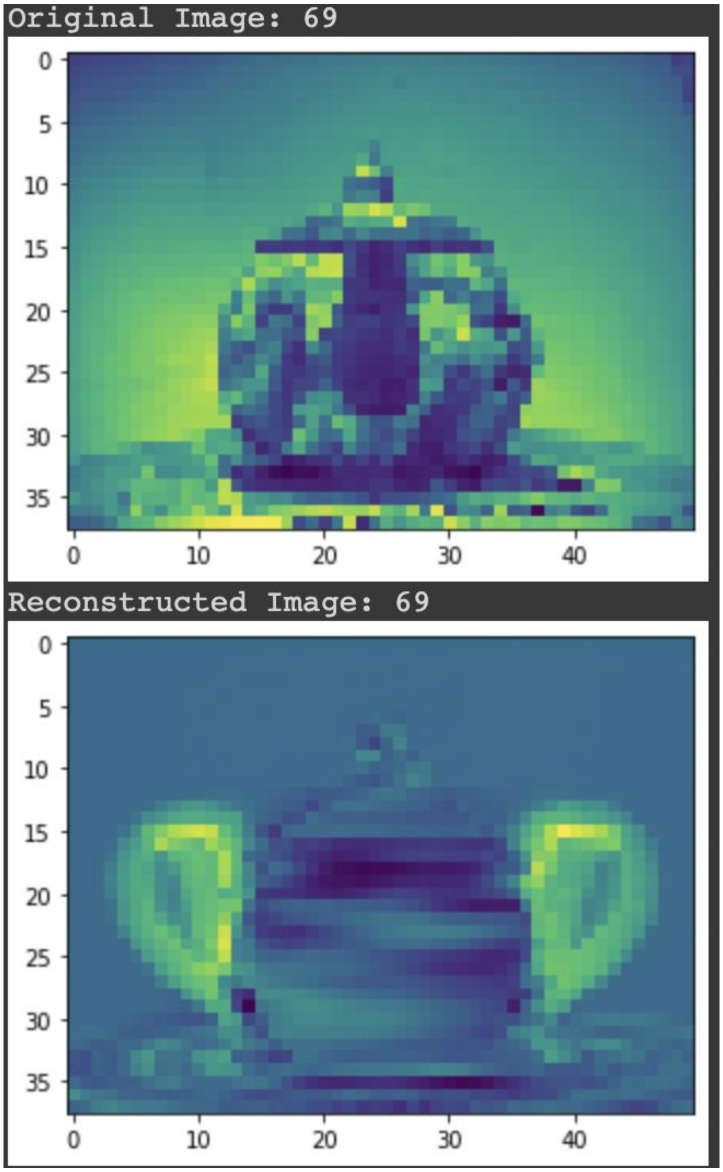
2.



```
Original Image: 19
```

```
Reconstructed Image: 19
```

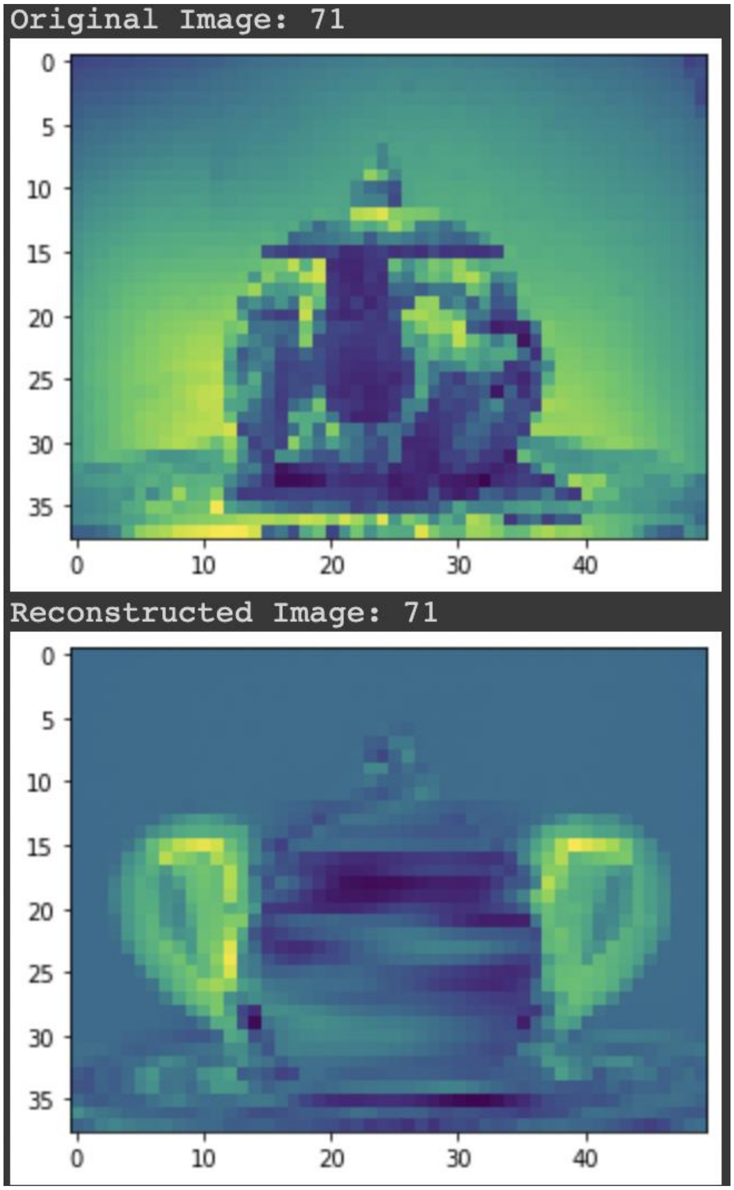
3.



```
Original Image: 69
```

```
Reconstructed Image: 69
```

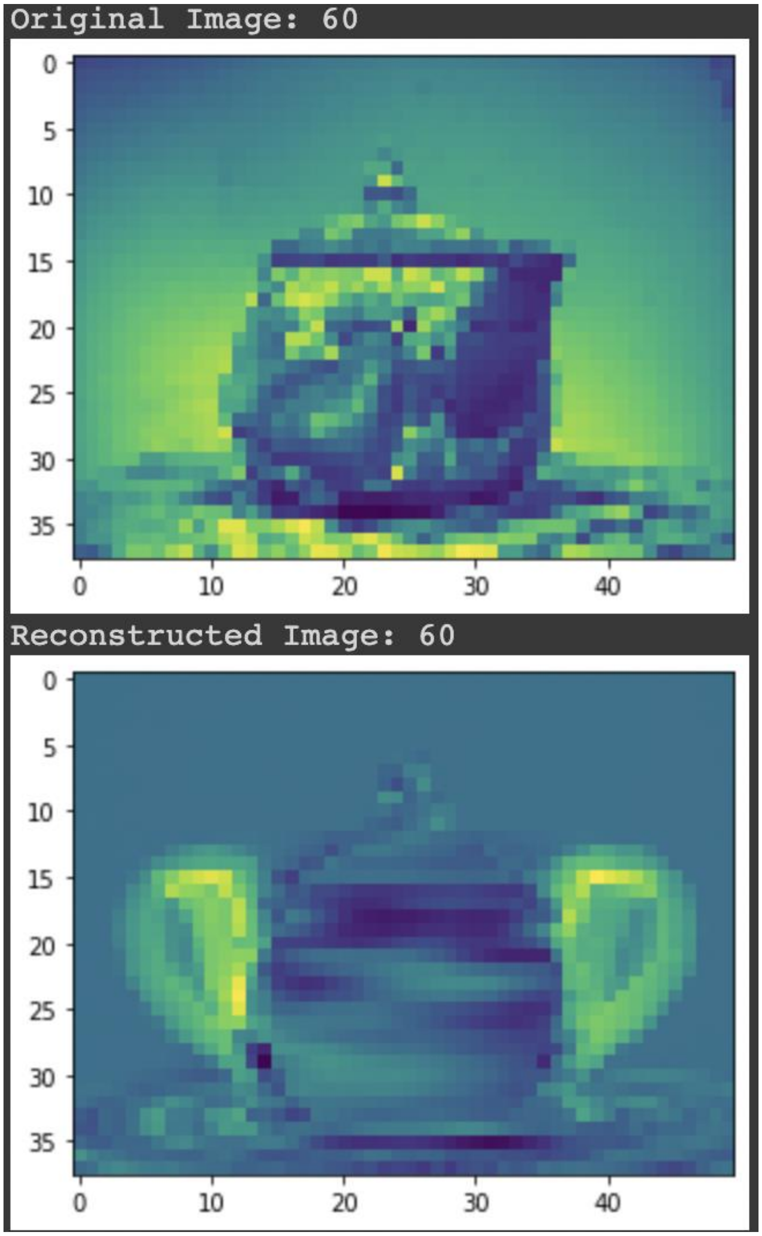
4.



```
Original Image: 71
```

```
Reconstructed Image: 71
```

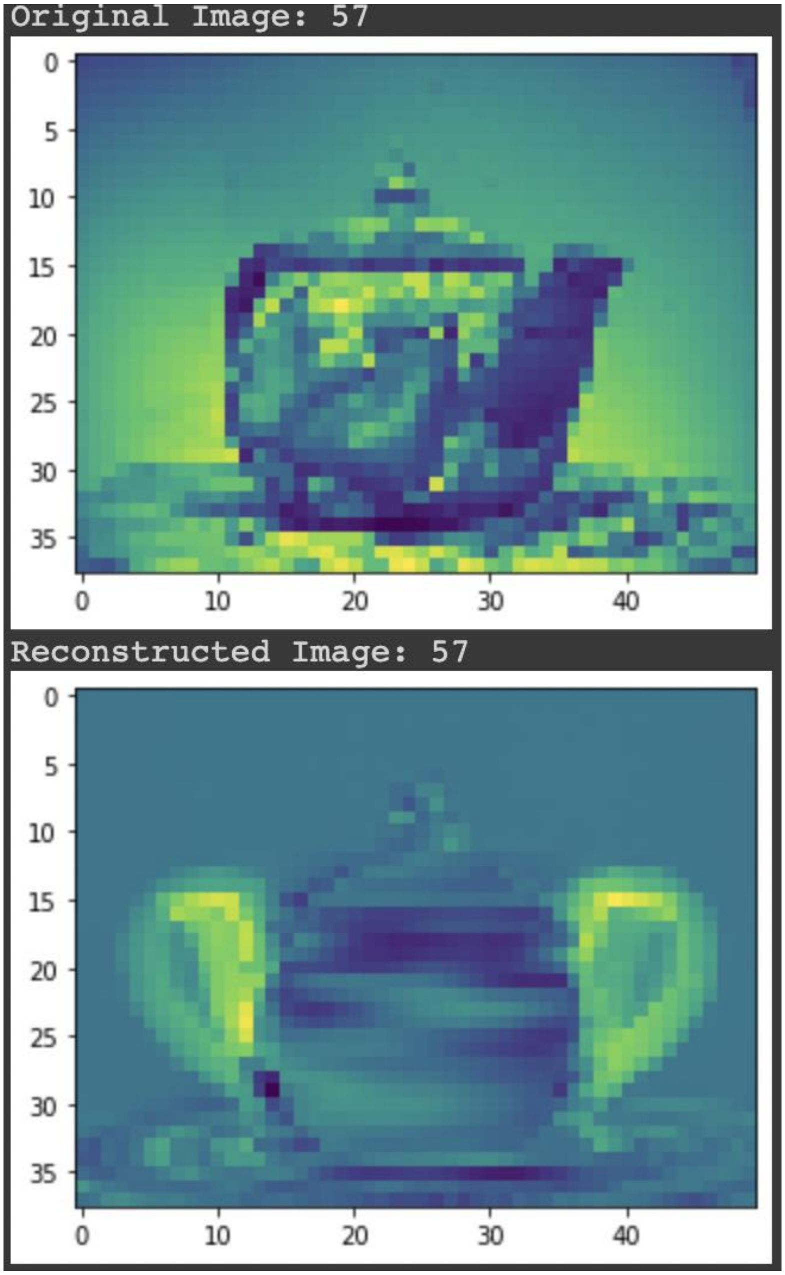
5.



```
Original Image: 60
```

```
Reconstructed Image: 60
```

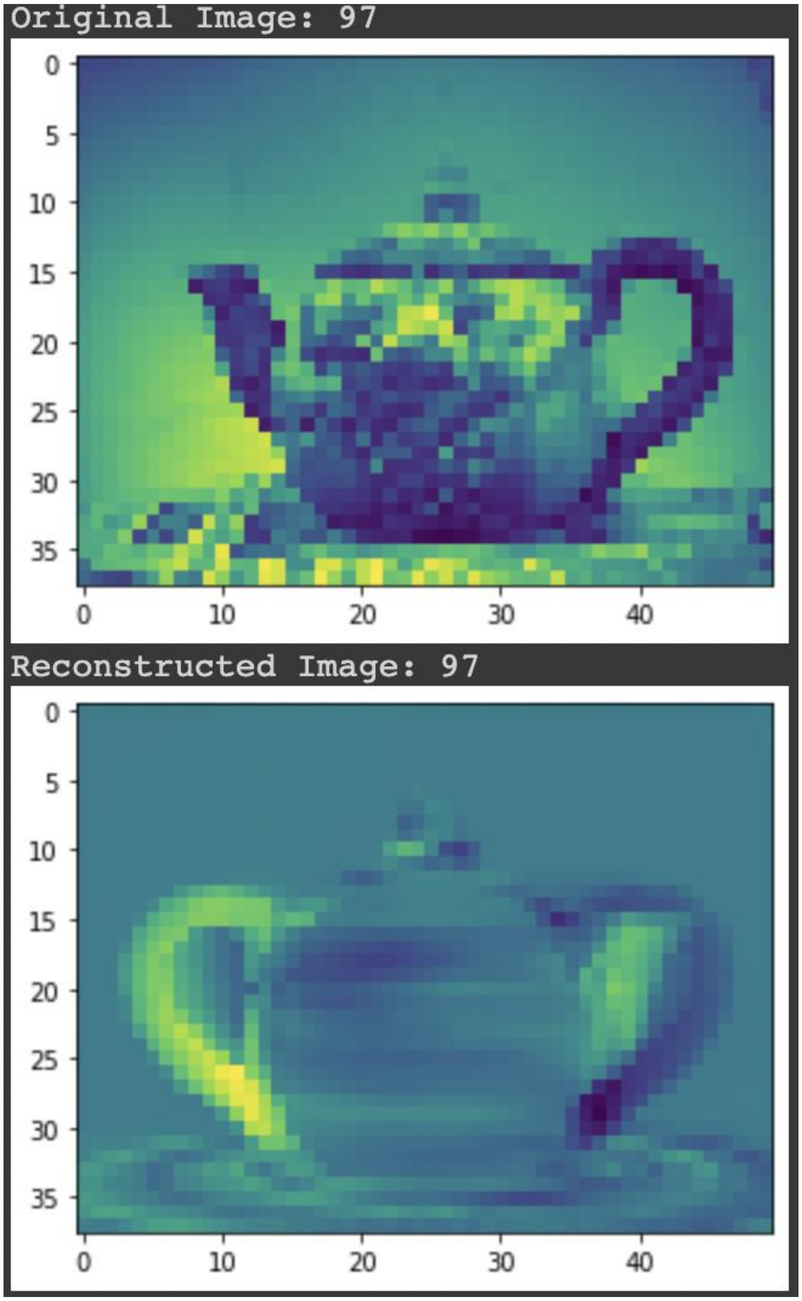
6.



```
Original Image: 57
```

```
Reconstructed Image: 57
```

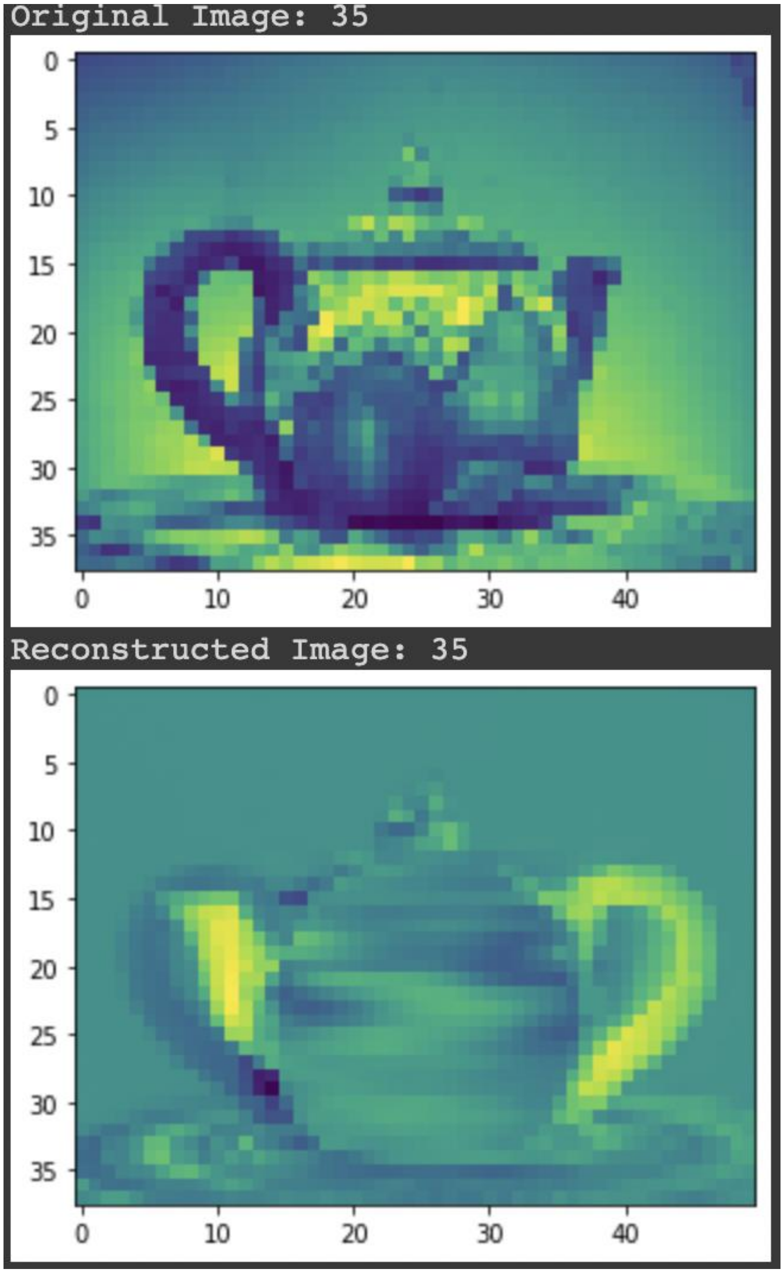
7.



Original Image: 97

Reconstructed Image: 97

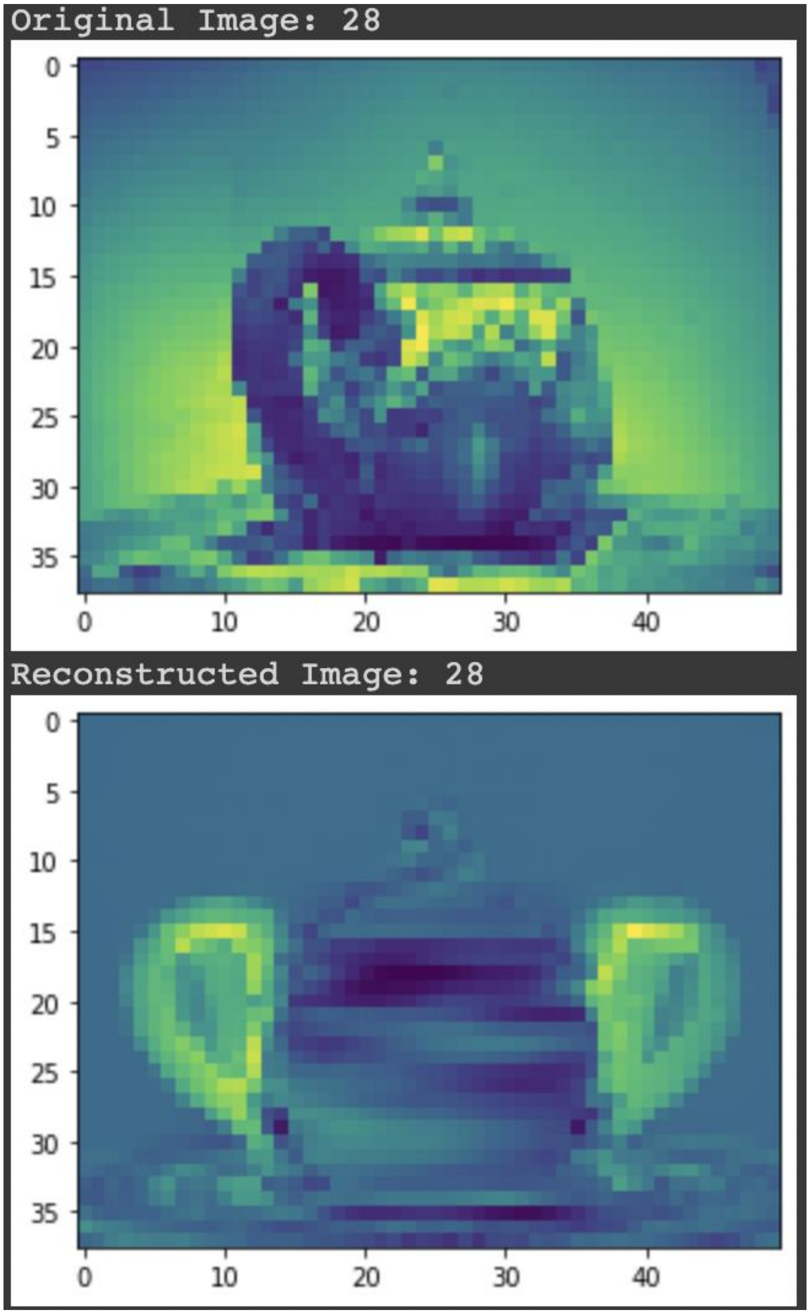
8.



Original Image: 35

Reconstructed Image: 35

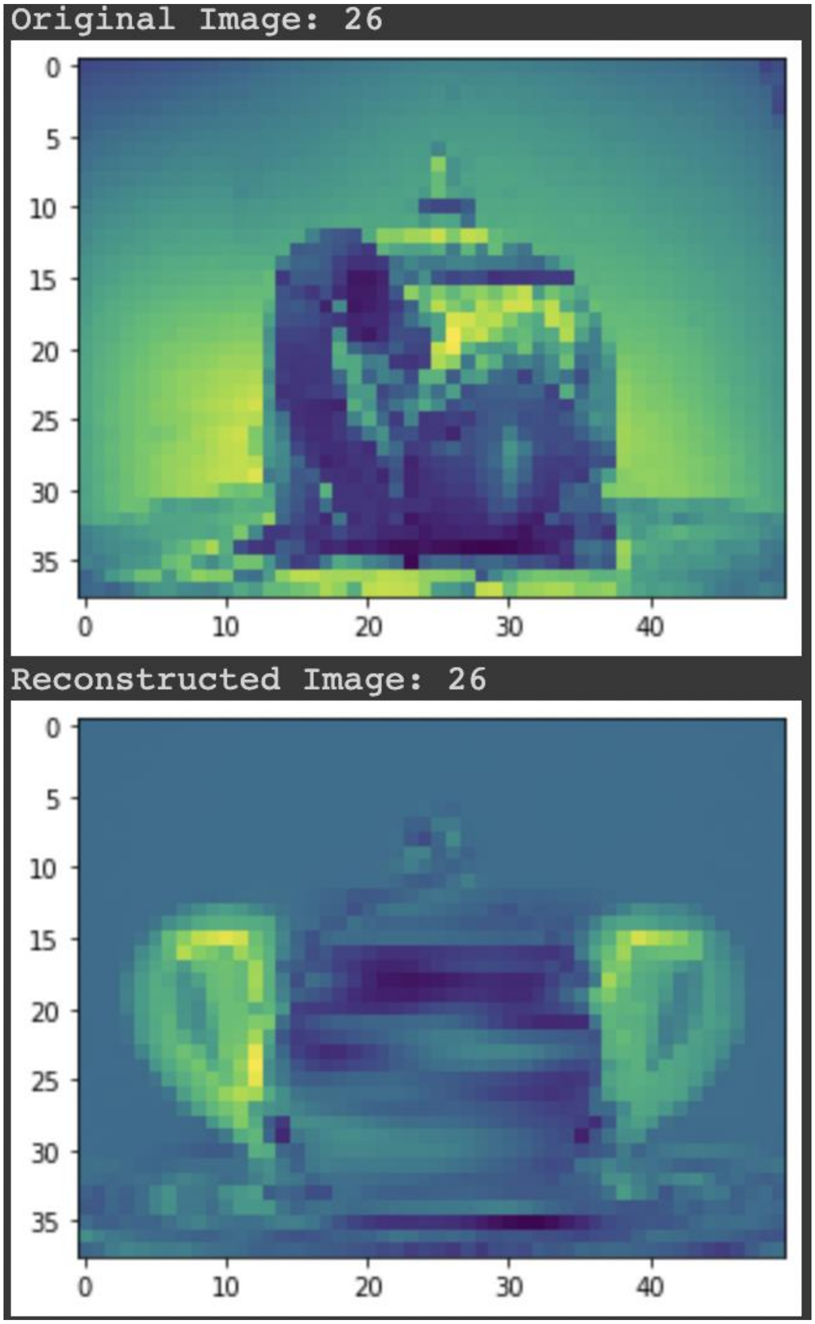
9.



Original Image: 28

Reconstructed Image: 28

10.



```
Original Image: 26
```

```
Reconstructed Image: 26
```

PCA retains the patterns in the data even after simplifying the complexities in the high dimensional data. It reduces the dimensionality of the data, later acting as summaries of the data.

Here we see an error of 368.15, corresponding to the usage of only 3 highest principal components. This could be reduced by using more of these components, but that in turn will increase the dimensionality of the data and will make the image tougher to compute.

As we reduce the dimensionality of the data, we reduce the correlations and in turn decrease the redundancy of the features. Essentially, this reduces the noise in the data and helps in reducing the computing time for these images.

Answer 2)

let $P(O) \rightarrow$ Probability of picking an orange.
$P(A) \rightarrow$ Probability of picking an apple

let first box be be b_1 and second box be b_2 so $P(b_1)$ and $P(b_2)$ are their respective probabilities.

To find : Probability that given an apple was chosen, , it was picked from b_1.

using Baye's Rule,

$$P(b_1|A) = \frac{P(A/b_1).\ P(b_1)}{P(A)}$$

$P(A/b_1)$: Probability of picking an apple from box b_1.

$$P(A|b_1) = \frac{\text{Total apples in } b_1}{\text{Total fruits in } b_1}$$

$$= \frac{8}{8+4} = \frac{2}{3}$$

$P(b_1) = 1/2$

$$P(A) = P(A/b_1)\,P(b_1) + P(A|b_2) \cdot P(b_2)$$

$$= \frac{2}{3} \times \frac{1}{2} + \frac{5}{6} \times \frac{1}{2} = \frac{1}{3} + \frac{5}{12} = \frac{9}{12} = \frac{3}{4}$$

So, $P(b_1/A) = \dfrac{2/3 \times 1/2}{3/4} = \dfrac{4}{9}$

Answer 3)

$p(y/\theta) = \alpha^{y}(1-\alpha)^{1-y}$

$p(x/y,\theta) = N(x/\mu_y, \Sigma_y)$

$\theta = \{\alpha, \mu_1, \Sigma_1, \mu_2, \Sigma_2\}$

As class probability $p(y/\theta)$ is modelled via Bernoulli distribution, and we are sampling an x from y^{th} Gaussian in $p(x/y,\theta)$

$$p(x,y/\theta) = p(y/\theta)\,p(x/y,\theta) \quad \text{—}\ \textcircled{1}$$

using maximum likelihood to recover the parameters.

$$p(x,y/\theta) = \prod_{i=1}^{N} p(y_i/\theta)\,p(x_i/y_i,\theta)$$

$$\log(p(x,y/\theta) = l(\theta) = \sum_{i=1}^{N} \log p(y_i/\theta) + \sum_{i=1}^{N} \log(p(x_i/y_i;\theta))$$

as y is being modelled using α,

$$\sum_{i=1}^{N} \log p(y_i/\theta) = \sum_{i=1}^{N} \log p(y_i/\alpha)$$

moreover, given that this is a 2 class problem let there be 2 classes, $y \in \{0, 1\}$

$$\sum_{i=1}^{N} \log(p(x_i/y_i, \theta) = \sum_{y_i \in 0} \log(p(x_i|\mu_1, \Sigma_2)) + \sum_{y_i \in 1} \log(p(x_i|\mu_2, \Sigma_2))$$

So, $l(\theta)$ can be written as,

$$l(\theta) = \underset{\overset{\uparrow}{\textcircled{1}}}{\sum_{i=1}^{N} \log p(y_i|\alpha)} + \sum_{y_i \in 0} \log(p(x_i|\mu_1, \Sigma_2)) \leftarrow \textcircled{2}$$

$$+ \sum_{y_i \in 1} \log(p(\underset{\overset{\uparrow}{\textcircled{3}}}{x_i}|\mu_2, \Sigma_2))$$

$$l(\theta) = l_1(\theta) + l_2(\theta) + l_3(\theta)$$

Doing max likelihood for the 3 terms separately.

$\textcircled{1} \rightarrow l_1(\theta) = \sum_{i=1}^{N} \log p(y_i|\alpha) = \sum_{i=1}^{N} \log(\alpha^{y_i}(1-\alpha)^{1-y_i})$

$$= \sum_{i=1}^{N} (y_i \log \alpha + (1-y_i) \log(1-\alpha))$$

$$= \sum_{i \in \text{class } 1} \log \alpha + \sum_{i \in \text{class } 0} \log(1-\alpha)$$

differentiating wrt α and equating to 0 to find α, where this equation has the max. value.

$$\frac{\partial}{\partial \alpha} \sum_{i \in \text{class } 1} \log \alpha + \sum_{i \in \text{class } 0} \log(1-\alpha) = 0$$

$$N_1 \frac{1}{\alpha} - N_0 \frac{1}{1-\alpha} = 0$$

$$N_1 (1-\alpha) - N_0 \alpha = 0$$

$$\boxed{\alpha = \frac{N_1}{N_1 + N_0}}$$

where N_0 and N_1 are the number of samples that belong to class $y \in 0$ and $y \in 1$ - respectively.

(2) → $\sum_{y_i \in 0} \log(p(x_i | \mu_1, \Sigma_1))$

$$\ell_2(\theta) = \sum_{y_i \in 0} \log\left(\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_1|}} \exp\left(-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1}(x_i - \vec{\mu}_1)\right)\right)$$

maximizing over μ,

$$\frac{\partial}{\partial \mu_1}\left(\sum_{y_i \in 0} \log\left(\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_1|}} \exp\left(-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1}(x_i - \vec{\mu}_1)\right)\right)\right) = 0$$

$$\frac{\partial}{\partial \mu_1}\left(\sum_{y_i \in 0}\left(-\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_1| - \frac{1}{2}(x_i - \mu_1)^T \Sigma^{-1}(x_i - \mu_1)\right)\right) = 0$$

$$= \frac{\partial}{\partial \mu_1}\left(-\frac{1}{2}\sum_{y_i \in 0}(x_i - \mu_1)^T \Sigma_1^{-1}(x_i - \mu_1)\right) = 0$$

$$\frac{\partial x^T A x}{\partial x} = x^T (A + A^T)$$

here $A = \Sigma^{-1}$

So, $\sum_{y_i \in 0} (x_i - \vec{\mu}_1)^T (\Sigma_1^{-1} + (\Sigma_1^{-1})^T) = 0$

As Σ is symmetric, $\sum_{y_i \in 0} (x_i - \mu_1)^T 2 \Sigma_1^{-1}$

Σ^{-1} can't be 0, as y depends on x.

so, $\sum_{y_i \in 0} (x_i - \vec{\mu}_1)^T = 0$

$$\sum_{y_i \in 0} \mu_1 = \sum_{y_i \in 0} x_i$$

$$\boxed{\mu_1 = \frac{1}{N_0} \sum_{y_i \in 0} x_i}$$

maximizing over Σ_1,

$$l_2(\theta) = \sum_{y_i \in 0} \log \left(\frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma_1|}} \exp \left(-\frac{1}{2} (x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \vec{\mu}_1) \right) \right)$$

$$= -\frac{N_0 D}{2} \log 2\pi + \frac{N_0}{2} \log (\Sigma_1^{-1}) - \frac{1}{2} \sum_{y_i \in 0} tr \left[(x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \vec{\mu}_1) \right]$$

$$= -\frac{N_0 D}{2} \log 2\pi + \frac{N_0}{2} \log (\Sigma_1^{-1}) - \frac{1}{2} \sum_{y_i \in 0} tr \left[(x_i - \mu_1) (x_i - \vec{\mu}_1)^T \Sigma_1^{-1} \right]$$

let $\Sigma_1^{-1} = A$

$$= -\frac{N_0 D}{2} \log 2\pi + \frac{N_0}{2} \log (A) - \frac{1}{2} \sum_{y_i \in 0} tr \left[(x_i - \mu_1) (x_i - \vec{\mu}_1)^T A \right]$$

We know, $\frac{\partial \log|A|}{\partial A} = (A^{-1})^T$ $\qquad$ $\frac{\partial tr[BA]}{\partial A} = B^T$

$$\frac{\partial l_2}{\partial A} = -0 + \frac{N_0}{2}(A^{-1})^T - \frac{1}{2}\sum_{y_i \in 0}\left[(x_i-\mu_1)(x_i-\mu_1)^T\right]^T$$

Getting sample covariance,

$$\frac{\partial l_2}{\delta A} = 0$$

So, $\frac{N_0}{2}\Sigma_1 - \frac{1}{2}\sum_{y_i \in 0}(x_i-\mu_1)(x_i-\mu_1)^T$

$\boxed{\Sigma_1 = \frac{1}{N_0}\sum_{y_i \in 0}(x_i-\mu_1)(x_i-\mu_1)^T}$ when $\mu_1 = \frac{1}{N_0}\sum_{i \in 0} x_i$

$\textcircled{3} \longrightarrow \sum_{y_i \in 1}\log(p(x_i|\mu_2, \Sigma_2))$

$$l_3(\theta) = \sum_{y_i \in 1}\log\left(\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_2|}}\exp\left(-\frac{1}{2}(x_i-\mu_2)^T\Sigma_2^{-1}(x_i-\vec{\mu}_2)\right)\right)$$

maximizing over μ,

$$\frac{\partial}{\partial \mu_2}\left(\sum_{y_i \in 1}\log\left(\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_2|}}\exp\left(-\frac{1}{2}(x_i-\mu_2)^T\Sigma_2^{-1}(x_i-\vec{\mu}_2)\right)\right)\right) = 0$$

$$\frac{\partial}{\partial \mu_2}\left(\sum_{y_i \in 1}\left(-\frac{D}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_2| - \frac{1}{2}(x_i-\mu_2)^T\Sigma_2^{-1}(x_i-\mu_2)\right)\right) -$$

$$= \frac{\partial}{\partial \mu_2}\left(\frac{-1}{2}\sum_{y_i \in 1}(x_i-\mu_2)^T\Sigma_2^{-1}(x_i-\mu_2)\right) = 0$$

we know, $\frac{\partial x^T A x}{\partial x} = x^T (A + A^T)$

So, $\sum_{y_i \in 1} (x_i - \vec{\mu}_2)^T (\Sigma_2^{-1} + (\Sigma_2^{-1})^T) = 0$

As covariance matrix is symmetric $\Sigma_2^{-1} + (\Sigma_2^{-1})^T = 2\Sigma_2^{-1}$ which cannot be 0 as there is no correlation between x and y.

So, $\sum_{y_i \in 1} (x_i - \vec{\mu}_2)^T = 0$

$$\sum_{y_i \in 1} \mu_2 = \sum_{y_i \in 1} x_i$$

$$\boxed{\mu_2 = \frac{1}{N_1} \sum_{y_i \in 1} x_i}$$

maximizing over Σ_2,

$$l_3(\theta) = \sum_{y_i \in 1} \log \left(\frac{1}{(2\pi)^{D/2} \sqrt{|\Sigma_2|}} \exp\left(-\frac{1}{2} (x_i - \mu_2)^T \Sigma_2^{-1} (x_i - \vec{\mu}_2) \right) \right)$$

$$= -\frac{N_1 D}{2} \log 2\pi + \frac{N_1}{2} \log(\Sigma_2^{-1}) - \frac{1}{2} \sum_{y_i \in 1} tr\left[(x_i - \mu_2)^T \Sigma_2^{-1} (x_i - \vec{\mu}_2) \right]$$

$$= -\frac{N_1 D}{2} \log 2\pi + \frac{N_1}{2} \log(\Sigma_2^{-1}) - \frac{1}{2} \sum_{y_i \in 1} tr\left[(x_i - \mu_2) (x_i - \vec{\mu}_2)^T \Sigma_2^{-1} \right]$$

let $\Sigma_2^{-1} = A$

$$= -\frac{N_1 D}{2} \log 2\pi + \frac{N_1}{2} \log(A) - \frac{1}{2} \sum_{y_i \in 1} tr\left[(x_i - \mu_2) (x_i - \vec{\mu}_2)^T A \right]$$

We know, $\frac{\partial \log|A|}{\partial A} = (A^{-1})^T$ $\qquad \frac{\partial tr[BA]}{\partial A} = B^T$

$$\frac{\partial l}{\partial A} = -0 + \frac{N_1}{2}(A^{-1})^T - \frac{1}{2}\sum_{y^i \in 1}[(x^i-\mu_2)(x^i-\mu_2)^T]^T$$

Getting sample covariance,

$$\frac{\partial l_3}{\delta A} = 0$$

So, $\frac{N_1}{2}\Sigma - \frac{1}{2}\sum_{y^i \in 1}(x^i-\mu_2)(x^i-\mu_2)^T$

$\boxed{\Sigma_2 = \frac{1}{N_1}\sum_{y^i \in 2}(x^i-\mu_2)(x^i-\mu_2)^T}$ where $\mu_2 = \frac{1}{N_1}\sum_{i \in 1} x^i$

Plotting decision boundary,

Bayes optimal label $y = \arg\max_{\hat{y}=\{0,1\}} P(\hat{y}/x)$

$$p(\hat{y}/x) = \frac{p(x,y)}{p(x)} = \frac{p(x,y)}{\sum_y p(x,y)} = \frac{p(x,y)}{p(x,y=0)+p(x,y=1)}$$

$$P(y=1/x) = \frac{P(x,y=1)}{P(x,y=0)+P(x,y=1)} = \frac{1}{\frac{P(x,y=0)}{P(x,y=1)}+1}$$

$$= \frac{1}{\exp\left(-\log\left(\frac{P(x,y=0)}{P(x,y=1)}\right)+1} = \text{sigmoid}\left(\log\left(\frac{P(x,y=0)}{P(x,y=1)}\right)\right)$$

Decision boundary depends on posterior.

$\Rightarrow \quad P(y=1/x) = P(y=0/x)$

$\Rightarrow \quad \alpha N(x/\mu_2, \Sigma_2) = (1-\alpha) N(x/\mu_1, \Sigma_1)$

As on the decision boundary $\alpha = 0.5$,

$\alpha = 1-\alpha$

So, $N(x/\mu_2, \Sigma_2) = N(x/\mu_1, \Sigma_1)$

$$\Rightarrow \frac{}{(2\pi)^{D/2}\sqrt{|\Sigma_2|}} \exp\left(-\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2)\right)$$

$$= \frac{}{(2\pi)^{D/2}\sqrt{|\Sigma_1|}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma_2^{-1}(x-\mu_1)\right)$$

$$\Rightarrow \frac{}{\cancel{(2\pi)^{D/2}}\sqrt{|\Sigma_2|}} \exp\left(-\frac{1}{2}(x-\mu_2)^T \Sigma_2^{-1}(x-\mu_2)\right)$$

$$= \frac{}{\cancel{(2\pi)^{D/2}}\sqrt{|\Sigma_1|}} \exp\left(-\frac{1}{2}(x-\mu_1)^T \Sigma_2^{-1}(x-\mu_1)\right)$$

$$\Rightarrow \frac{}{\sqrt{|\Sigma_2|}} (\exp(x^T \Sigma_2^{-1} x - x^T \Sigma^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} x + \mu_2^T \Sigma_2^{-1} \mu_2)$$

$$= \frac{}{\sqrt{|\Sigma_1|}} (\exp(x^T \Sigma_1^{-1} x - x^T \Sigma_1^{-1} \mu_1 - \mu_1^T \Sigma_1^{-1} x + \mu_1^T \Sigma_1^{-1} \mu_1)$$

$$\Rightarrow \sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}} = \frac{(\exp(x^T \Sigma_1^{-1} x - x^T \Sigma_1^{-1} \mu_1 - \mu_1^T \Sigma_2^{-1} x + \mu_1^T \Sigma_1^{-1} \mu \ )}{\exp(x^T \Sigma_2^{-1} x - x^T \Sigma_2^{-1} \mu_2 - \mu_2^T \Sigma_2^{-1} x + \mu_2^T \Sigma_2^{-1} \mu_2)}$$

$$\Rightarrow \sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}} = \frac{(\exp(x^T\Sigma_1^{-1}x - x^T\Sigma_1^{-1}\mu_1 - \mu_1^T\Sigma_2^{-1}x + \mu_1^T\Sigma_1^{-1}\mu\ )}{\exp(x^T\Sigma_2^{-1}x - x^T\Sigma_2^{-1}\mu_2 - \mu_2^T\Sigma_2^{-1}x + \mu_2^T\Sigma_2^{-1}\mu_2)}$$

$$\Rightarrow \sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}} = \exp(((x^T\Sigma_1^{-1}x - x^T\Sigma_1^{-1}\mu_1 - \mu_1^T\Sigma_1^{-1}x + \mu_1^T\Sigma_1^{-1}\mu_1) - ((x^T\Sigma_2^{-1}x - x^T\Sigma_2^{-1}\mu_2 - \mu_2^T\Sigma_2^{-1}x + \mu_2^T\Sigma_2^{-1}\mu_2))$$

taking log on both sides,

$$\log\left(\sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}}\right) = x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x - x^T(\Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2) - (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})x + \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2 \quad - (A)$$

when $\Sigma_1 = \Sigma_2 = \Sigma$

(A) becomes,

$$\log(1) = x^T(\Sigma^{-1} - \Sigma^{-1})x - x^T(\Sigma^{-1}\mu_1 - \Sigma^{-1}\mu_2) - (\mu_1^T\Sigma^{-1} - \mu_2^T\Sigma^{-1})x + \mu_1^T\Sigma^{-1}\mu_1 - \mu_2^T\Sigma^{-1}\mu_2$$

$$\Rightarrow -x^T\Sigma^{-1}(\mu_1 - \mu_2) - (\mu_1^T - \mu_2^T)\Sigma^{-1}x + \mu_1^T\Sigma^{-1}\mu_1 - \mu_2^T\Sigma^{-1}\mu_2 = 0$$

$$2(\mu_1^T - \mu_2^T)\Sigma^{-1}x + \mu_2^T\Sigma^{-1}\mu_2 - \mu_1^T\Sigma^{-1}\mu_1 = 0$$

This only has x in linear powers, and $\mu_i\Sigma^{-1}\mu_i$ is a constant, hence we have a linear decision boundary.

if $\Sigma_1 \neq \Sigma_2$

$$\Rightarrow \log\left(\sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}}\right) = x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x - x^T(\Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2) - (\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})x + \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2 \quad — \textcircled{A}$$

$$\Rightarrow x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x - 2(\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1})x + \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2 - \log\left(\sqrt{\frac{|\Sigma_1|}{|\Sigma_2|}}\right) = 0$$

Here we can see the first term has a quadratic term followed by a linear term in x and a constant.

Thus when the covariances aren't equal we get a quadratic decision boundary.

On the next page there is the derivation of decision boundary when $\Sigma_1 \neq \Sigma_2$, but using scalar terms/notations of the variables to have a better understanding of the quadratic decision boundary.

if $\Sigma_1 \neq \Sigma_2$

$$\exp\left(-\frac{1}{2\Sigma_2}(2-\mu_2)^2\right) = \exp\left(-\frac{1}{2\Sigma_1}(x-\mu_1)^2\right)$$

$$1 = \frac{\exp\left(-\frac{1}{2\Sigma_1}(x-\mu_1)^2\right)}{\exp\left(-\frac{1}{2\Sigma_2}(2-\mu_2)^2\right)}$$

$$1 = \exp\left(-\frac{1}{2}\left[\frac{x^2}{\Sigma_1} - \frac{2x\mu_1}{\Sigma_1} + \frac{\mu_1^2}{\Sigma_1} - \frac{x^2}{\Sigma_2} + \frac{2x\mu_2}{\Sigma_2} - \frac{\mu_2^2}{\Sigma_2}\right]\right.$$

taking log,

$$\log(1) = \left[x^2\left(\frac{1}{\Sigma_1} - \frac{1}{\Sigma_2}\right) - 2x\left(\frac{\mu_1}{\Sigma_1} - \frac{\mu_2}{\Sigma_2}\right) + \frac{\mu_1^2}{\Sigma_1} - \frac{\mu_2^2}{\Sigma_2}\right]$$

$$x^2\left(\frac{1}{\Sigma_1} - \frac{1}{\Sigma_2}\right) - 2x\left(\frac{\mu_1}{\Sigma_1} - \frac{\mu_2}{\Sigma_2}\right) + \frac{\mu_1^2}{\Sigma_1} - \frac{\mu_2^2}{\Sigma_2} = 0$$

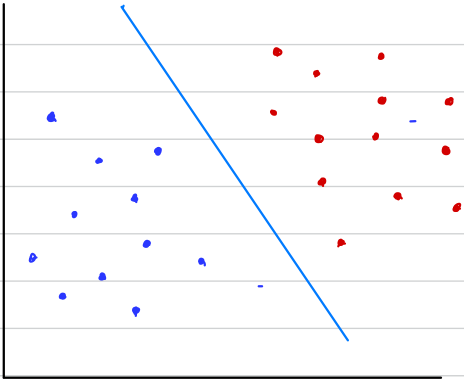
which can be written as, $ax^2 + bx + c = 0$

where,

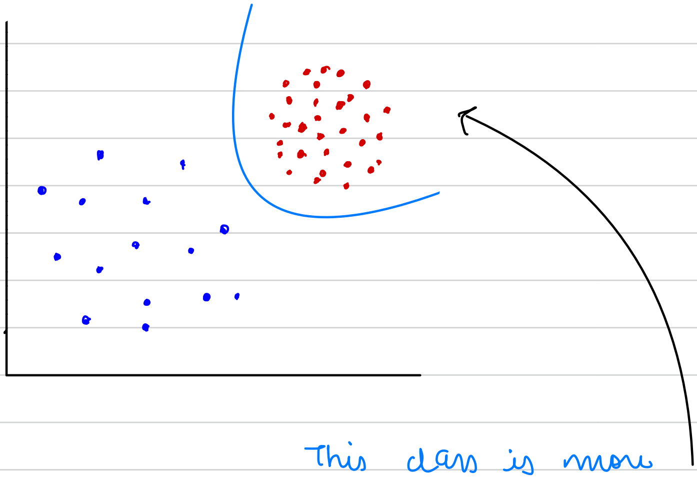
$$a = \left(\frac{1}{\Sigma_1} - \frac{1}{\Sigma_2}\right) \quad b = \left(\frac{\mu_1}{\Sigma_1} - \frac{\mu_2}{\Sigma_2}\right) \quad c = \frac{\mu_1^2}{\Sigma_1} - \frac{\mu_2^2}{\Sigma_2}$$

Thus resulting in a quadratic decision boundary.

When $\Sigma_1 = \Sigma_2 = \Sigma$, linear decision boundary



When $\Sigma_1 \neq \Sigma_2$, quadratic decision boundary.



This class is more concentrated around its mean.