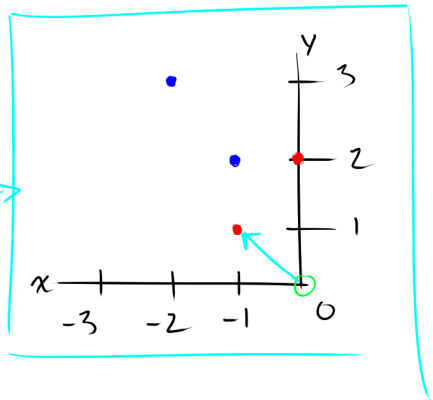
1) a) $m = \frac{1}{4}\left(\begin{bmatrix}0\\2\end{bmatrix}+\begin{bmatrix}-1\\1\end{bmatrix}+\begin{bmatrix}-1\\2\end{bmatrix}+\begin{bmatrix}-2\\3\end{bmatrix}\right) = \frac{1}{4}\begin{bmatrix}-4\\8\end{bmatrix} = \begin{bmatrix}-1\\2\end{bmatrix}$

$\Sigma = \text{cov}(X) = \frac{1}{n-1}XX^T = \frac{1}{3}\begin{bmatrix}-1&0&0&-1\\0&-1&0&1\end{bmatrix}\begin{bmatrix}-1&0\\0&-1\\0&0\\-1&1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}2&-1\\-1&2\end{bmatrix} = \begin{bmatrix}\frac{2}{3}&-\frac{1}{3}\\-\frac{1}{3}&\frac{2}{3}\end{bmatrix}$

$\Sigma w = \lambda w \qquad \det(\Sigma - \lambda I) = 0$

$\det\left(\begin{bmatrix}\frac{2}{3}-\lambda&-\frac{1}{3}\\-\frac{1}{3}&\frac{2}{3}-\lambda\end{bmatrix}\right) = 0 \Rightarrow \begin{matrix}\lambda_1 = 1\\ \lambda_2 \neq 1/3\end{matrix} \Rightarrow \begin{matrix}w_1 = \begin{bmatrix}-1\\1\end{bmatrix}\\ w_2 = \begin{bmatrix}1\\1\end{bmatrix}\end{matrix}$



b) $x^t \in \mathbb{R}^{D\times 1}$
$W \in \mathbb{R}^{D\times d}$, where $d < D$
$z^t = W^T x^t$
$v^t = Wz^t = WW^T x^t = x^t$?

} ID MATRIX?

DIDN'T HAVE ENOUGH TIME UMU

c) S_{win} ← Within class $= S_A + S_B$

$m_A = \begin{bmatrix}-.5\\1.5\end{bmatrix} \Rightarrow S_A = \begin{bmatrix}-.5&-.5\\.5&-.5\end{bmatrix}\begin{bmatrix}-.5&.5\\-.5&-.5\end{bmatrix} = \begin{bmatrix}.5&0\\0&.5\end{bmatrix}$

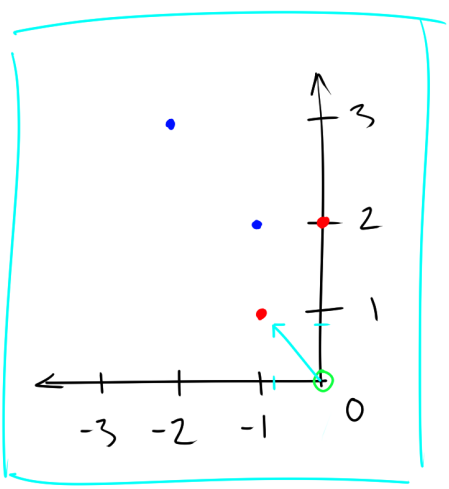
$m_B = \begin{bmatrix}-1.5\\2.5\end{bmatrix} \Rightarrow S_B = \begin{bmatrix}.5&-.5\\-.5&.5\end{bmatrix}\begin{bmatrix}.5&-.5\\-.5&.5\end{bmatrix} = \begin{bmatrix}.5&-.5\\-.5&.5\end{bmatrix}$ +

$= \begin{bmatrix}1&-.5\\-.5&1\end{bmatrix}$

$S_{B/T} = \begin{bmatrix}1\\-1\end{bmatrix}\begin{bmatrix}1&,&-1\end{bmatrix} = \begin{bmatrix}1&-1\\-1&1\end{bmatrix}$

↑ Between-Class

d) $w = c * S_w^{-1}(m_1 - m_2) \Rightarrow c * \begin{bmatrix}\frac{4}{3}&\frac{2}{3}\\\frac{2}{3}&\frac{4}{3}\end{bmatrix}\begin{bmatrix}1\\-1\end{bmatrix} \Rightarrow c * \begin{bmatrix}\frac{2}{3}\\-\frac{2}{3}\end{bmatrix} \Rightarrow w = \begin{bmatrix}\frac{1}{\sqrt{2}}\\-\frac{1}{\sqrt{2}}\end{bmatrix}$



2) $x_1 = 1 \quad x_2 = 4 \quad x_3 = 5 \quad x_4 = 6 \quad x_5 = 7 \quad x_6 = 8 \quad x_7 = 10 \quad x_8 = 12 \quad x_9 = 14$

$k = 3$

a) init: $c_1 = 0 \quad c_2 = 5 \quad c_3 = 10$

assignment: $C_1 = \{x_1\} \quad C_2 = \{x_2, x_3, x_4, x_5\} \quad C_3 = \{x_6, x_7, x_8, x_9\}$ Iter 1

update means: $m_1 = 1 \quad m_2 = 5.5 \quad m_3 = 11$

assignment: $C_1 = \{x_1\} \quad C_2 = \{x_2, x_3, x_4, x_5, x_6\} \quad C_3 = \{x_7, x_8, x_9\}$ Iter 2

update means: $m_1 = 1 \quad m_2 = 6 \quad m_3 = 12$

assignment: $C_1 = \{x_1\} \quad C_2 = \{x_2, x_3, x_4, x_5, x_6\} \quad C_3 = \{x_7, x_8, x_9\}$ Iter 3

update means: $m_1 = 1 \quad m_2 = 6 \quad m_3 = 12$

assignment: $C_1 = \{x_1\} \quad C_2 = \{x_2, x_3, x_4, x_5, x_6\} \quad C_3 = \{x_7, x_8, x_9\}$ Iter 4

$m_1 = 1 \quad m_2 = 6 \quad m_3 = 12$

b) It took 3 iterations for the k-means algorithm to converge.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2 = \cancel{(1-1)^2} + \cancel{(4-6)^2}^{4} + \cancel{(5-6)^2}^{1} + \cancel{(6-6)^2} + \cancel{(7-6)^2}^{1} + \cancel{(8-6)^2}^{4}$$
$$+ \cancel{(10-12)^2}^{4} + \cancel{(12-12)^2} + \cancel{(14-12)^2}^{4} \Rightarrow \underbrace{4+1+1}_{6} + \underbrace{4+4+4}_{12} = \boxed{18}$$

c) init: $c_1 = 2$ $c_2 = 7$ $c_3 = 12$

Iteration			
1	assignment: $c_1 = \{x_1, x_2\}$ (x_1, x_2 = 1, 4)	$c_2 = \{x_3, x_4, x_5, x_6\}$ (5, 6, 7, 8)	$c_3 = \{x_7, x_8, x_9\}$ (10, 12, 14)
	update means: $m_1 = 2.5$	$m_2 = 6.5$	$m_3 = 12$
2	assignment: $c_1 = \{x_1, x_2\}$	$c_2 = \{x_3, x_4, x_5, x_6\}$	$c_3 = \{x_7, x_8, x_9\}$
	update means: $m_1 = 2.5$	$m_2 = 6.5$	$m_3 = 12$
3	assignment: $c_1 = \{x_1, x_2\}$	$c_2 = \{x_3, x_4, x_5, x_6\}$	$c_3 = \{x_7, x_8, x_9\}$
	update means: $m_1 = 2.5$	$m_2 = 6.5$	$m_3 = 12$
4	assignment: $c_1 = \{x_1, x_2\}$	$c_2 = \{x_3, x_4, x_5, x_6\}$	$c_3 = \{x_7, x_8, x_9\}$
	update means: $m_1 = 2.5$	$m_2 = 6.5$	$m_3 = 12$

d) It took 2 iterations for the k-means algorithm to converge in this case.

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2 = \underbrace{(1-2.5)^2}_{2.25} + \underbrace{(4-2.5)^2}_{2.25} + \underbrace{(5-6.5)^2}_{2.25} + \underbrace{(6-6.5)^2}_{.25} + \underbrace{(7-6.5)^2}_{.25} + \underbrace{(8-6.5)^2}_{2.25} + \underbrace{(10-12)^2}_{4} + \underbrace{(12-12)^2}_{0} + \underbrace{(14-12)^2}_{4} = \boxed{17.5}$$

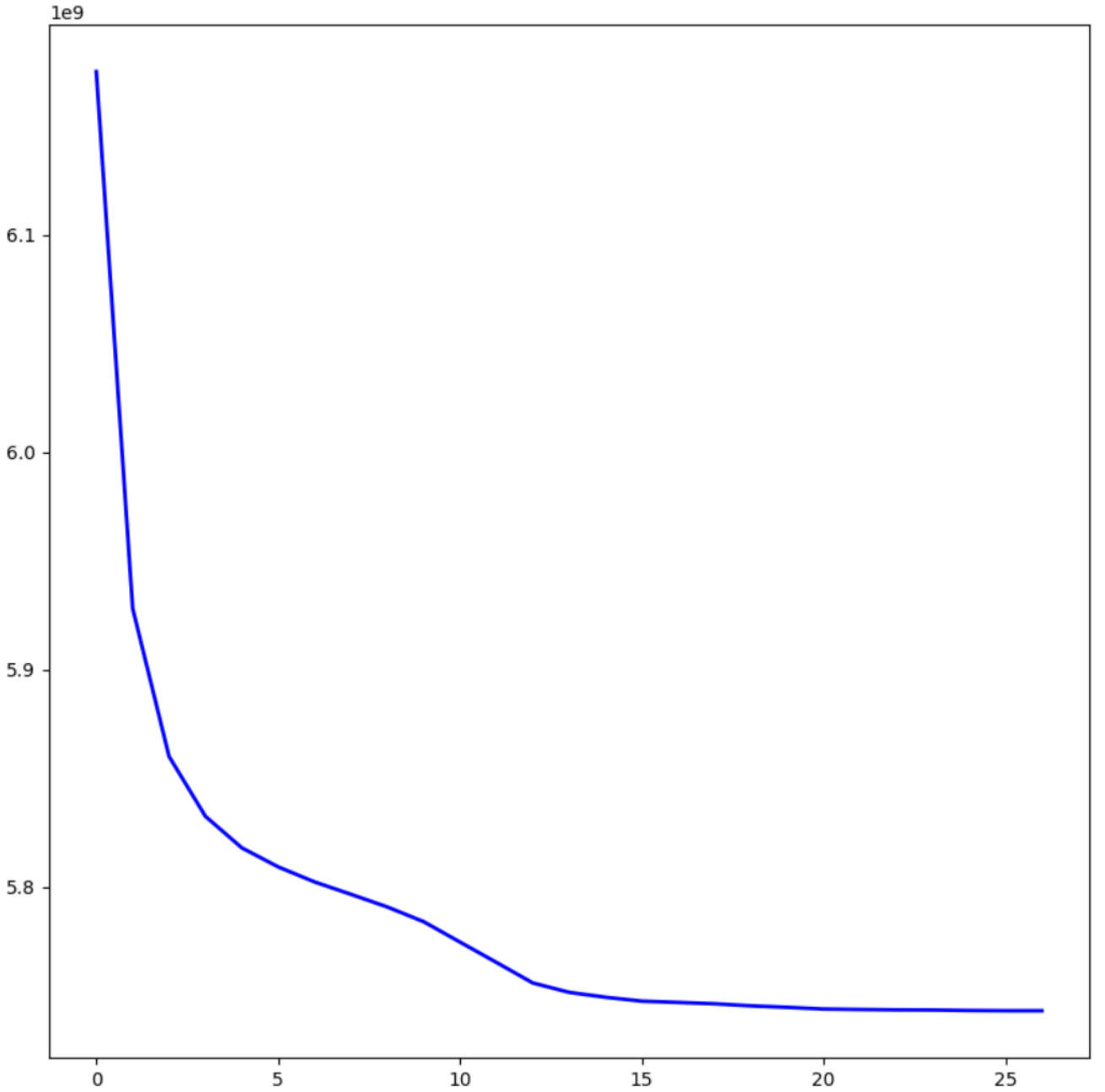
e) When comparing (a) and (c), (b) is the better solution because $J_a > J_b \Rightarrow 18 > 17.5$ (Our goal is to minimize J).

3.

Result:

```
Using raw data converged in 27 iteration (3.74 seconds)
Classification accuracy: 0.94
##################
Project data into 73 dimensions with PCA converged in 26 iteration (2.96 seconds)
Classification accuracy: 0.94
##################
Project data into 1 dimension with PCA converged in 33 iteration (3.75 seconds)
Classification accuracy: 0.60
```

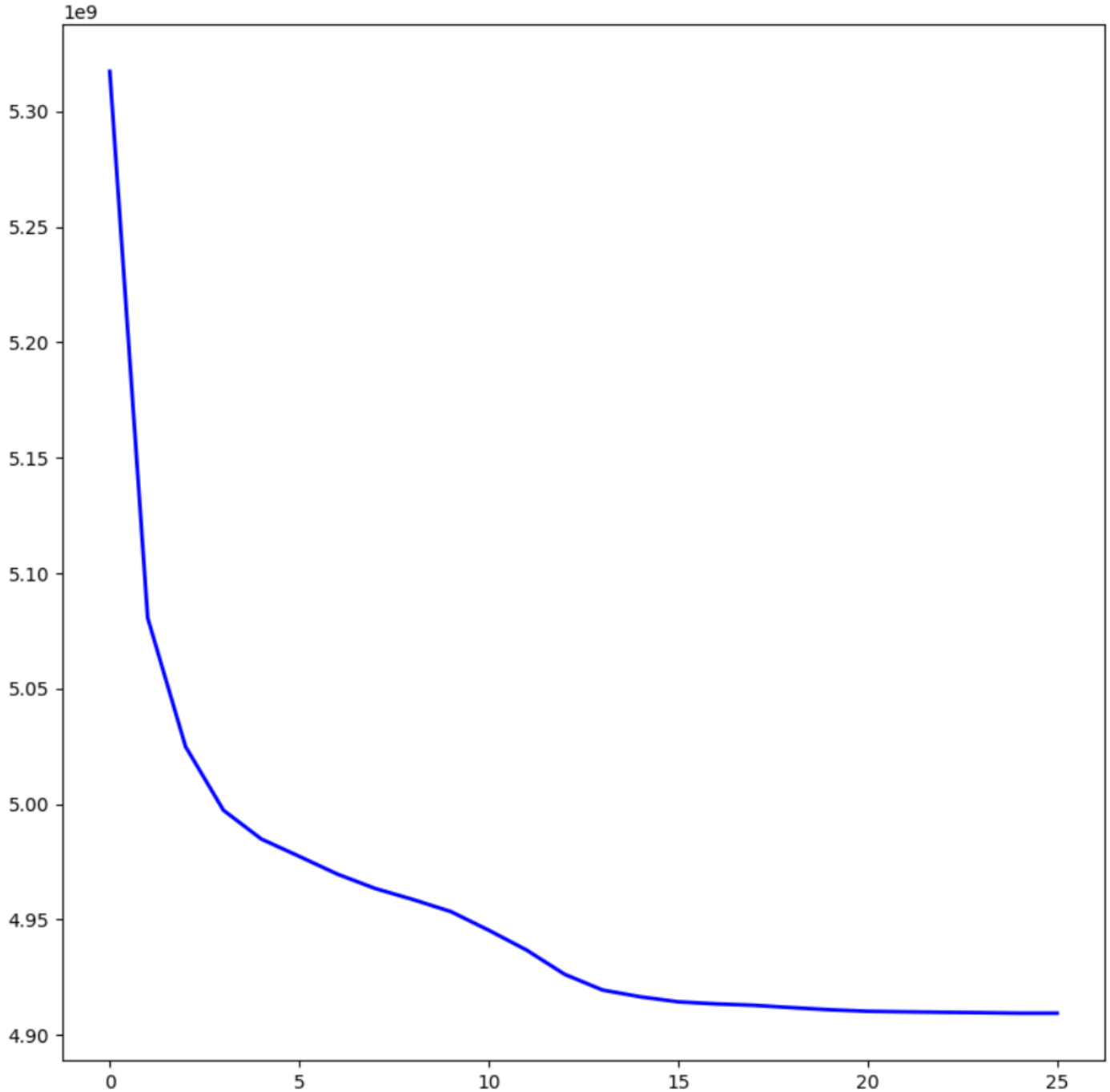
Raw data:



This one took 27 iterations with an accuracy of 94%.

The plot shape is kinda following what I'm expecting -- as there are more iterations, the reconstruction error decreases because the centers' variance from the data gets closer to some maximal value (i.e. J is minimizing). However, there's a small "bump" along the elbow of the curve, and I can probably guess that that's because the difference in the cluster labels changed slightly, but later on changed a reasonable bit.

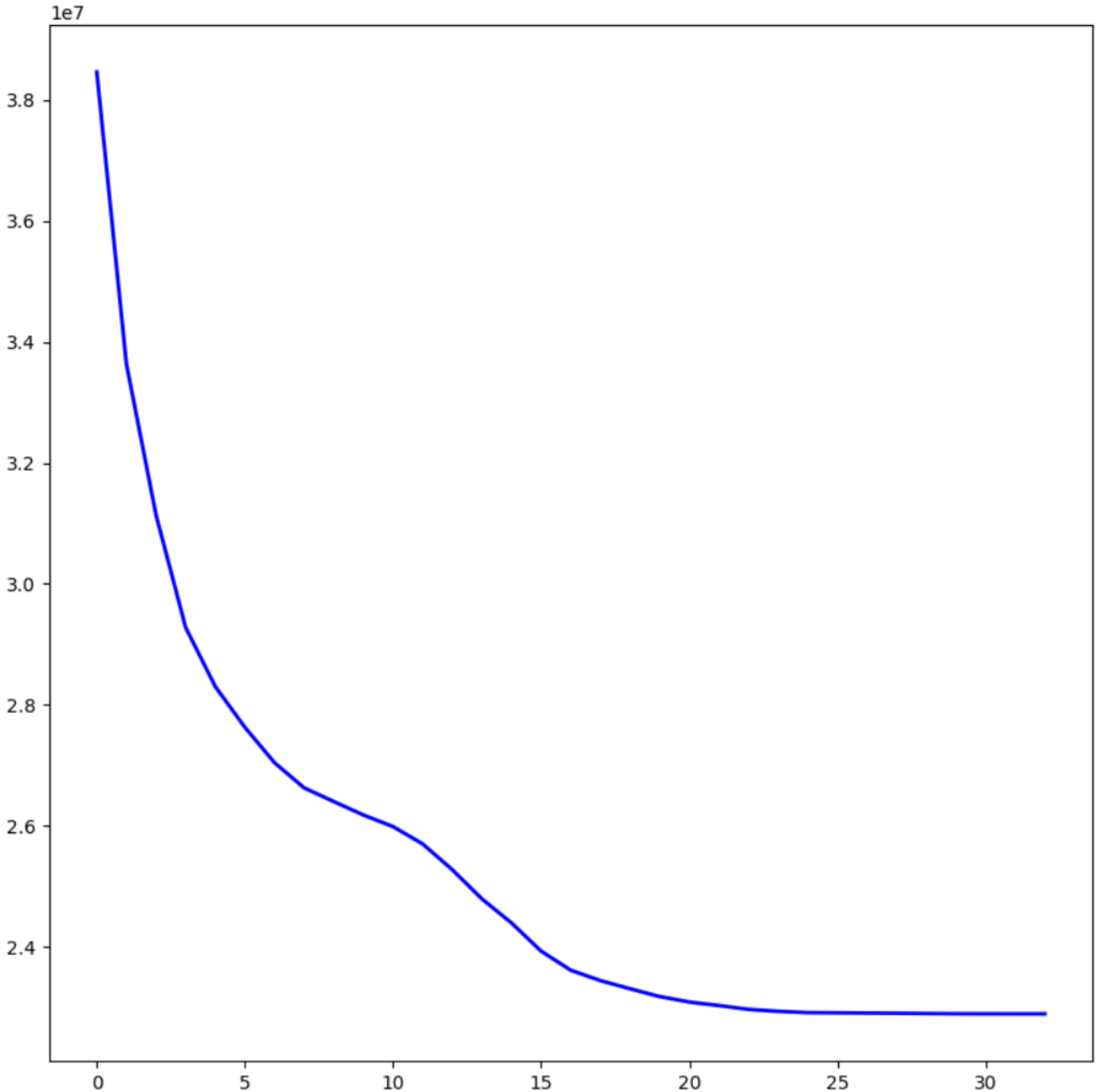
Low-dimensional data (PCA) that captures >90% of variance:



This one took 26 iterations with an accuracy of 94%.

The shape of the curve is noticeably similar to the raw data assumption, however with PCA to help clustering, we see that the value flattens out to about 4.90 rather than about 5.7 with the previous one. So yeah, I can say that PCA helps with clustering.

1 Principal Component



This one took 33 iterations with an accuracy of 60%. More iterations than the last two and terribly inaccurate compared to the first two assumptions.

The results are probably terrible because there wasn't a lot of features to go off of in the training stage. Another reason is because we only took the first principal component, which corresponds to the first eigenvalue and vector. Because of that, we lose much of the information that could be used for learning -- hence the low classification accuracy. However, on the positive side, this terribleness leads to a much lower reconstruction error -- meaning that we could attempt to reconstruct the original with low loss.