Statistics and Machine Learning

K-means and PCA

Week 14: 04/19 — 04/23

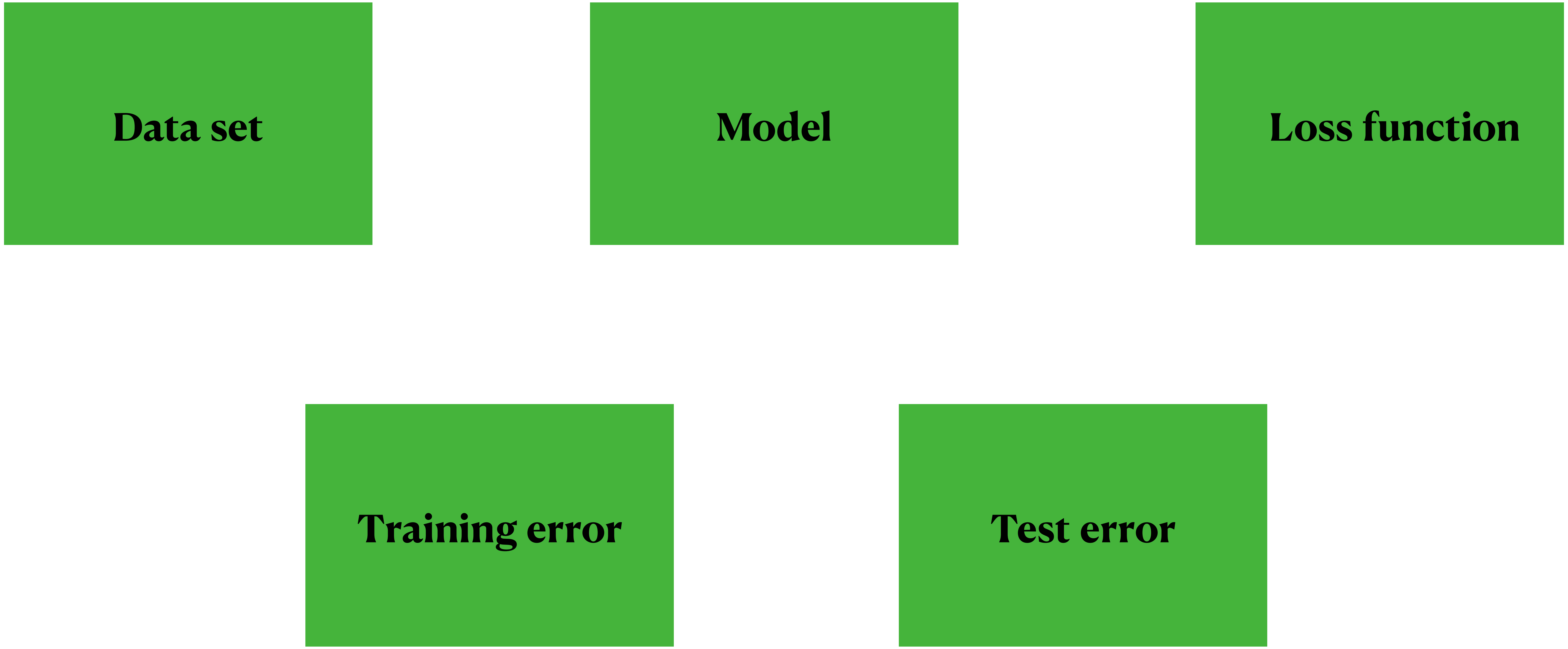
Contents of Week 14

K-means and pca

- Review of supervised learning and 5 building blocks
- What is un-supervised learning? Examples.
- K-means: clustering
- PCA: finding maximum variance from linear combination of features
- Lab: Hand-written digits, PCA plus k-means

5 building blocks for ML tasks

From supervised learning to unsupervised learning



Review: regression

Data set

Model

Loss function

Review: regression

In [26]:
```
advertising.head()
```
Out[26]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

Model

Loss function

Review: regression

In [26]:
```
advertising.head()
```

Out[26]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

$\boxed{\text{Sales = f(TV, Radio, newspaper)}}$

Model

Loss function

Review: regression

```
In [26]: advertising.head()
```

Out[26]:

	Unnamed: 0	TV	radio	newspaper	sales
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

Sales = f(TV, Radio, newspaper)

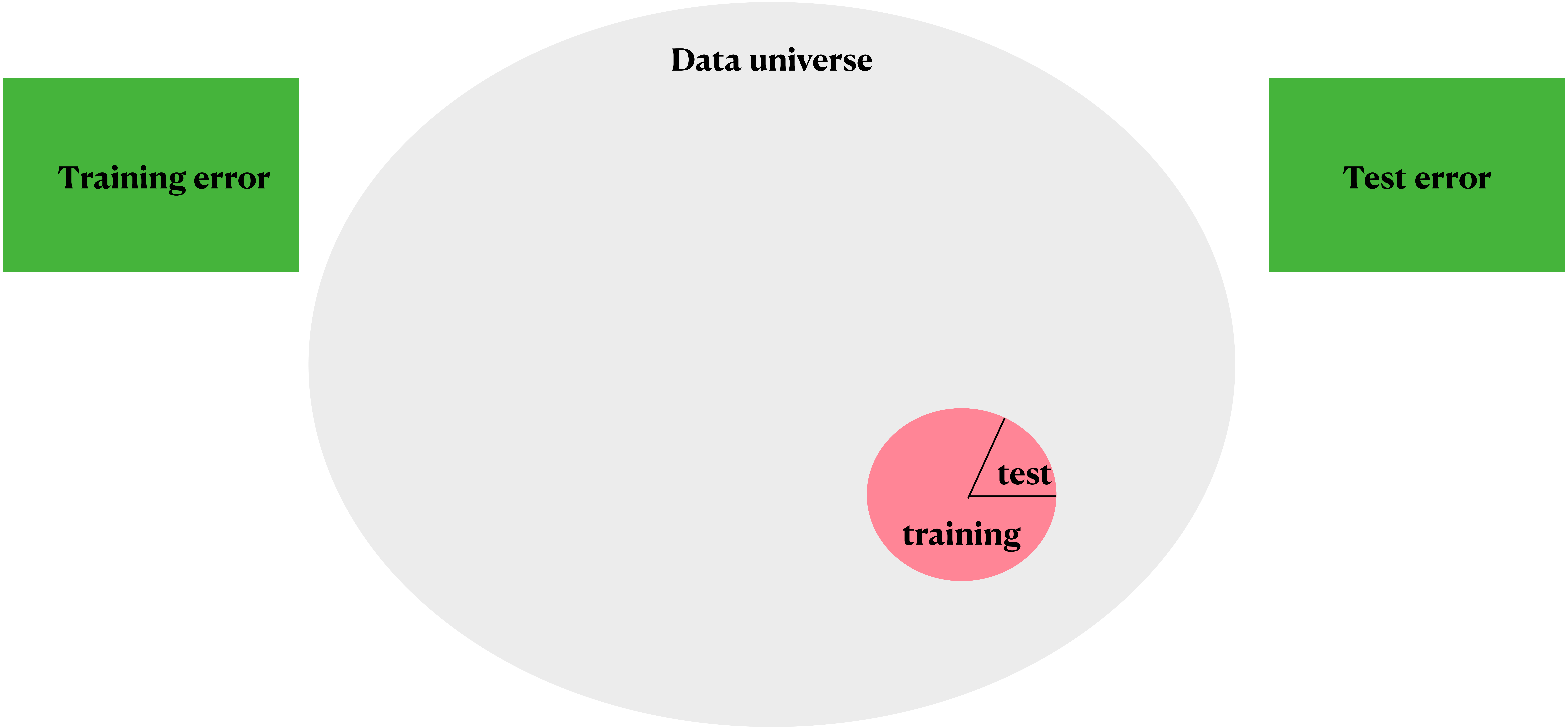
Model

Mean squared deviation between truth and prediction:

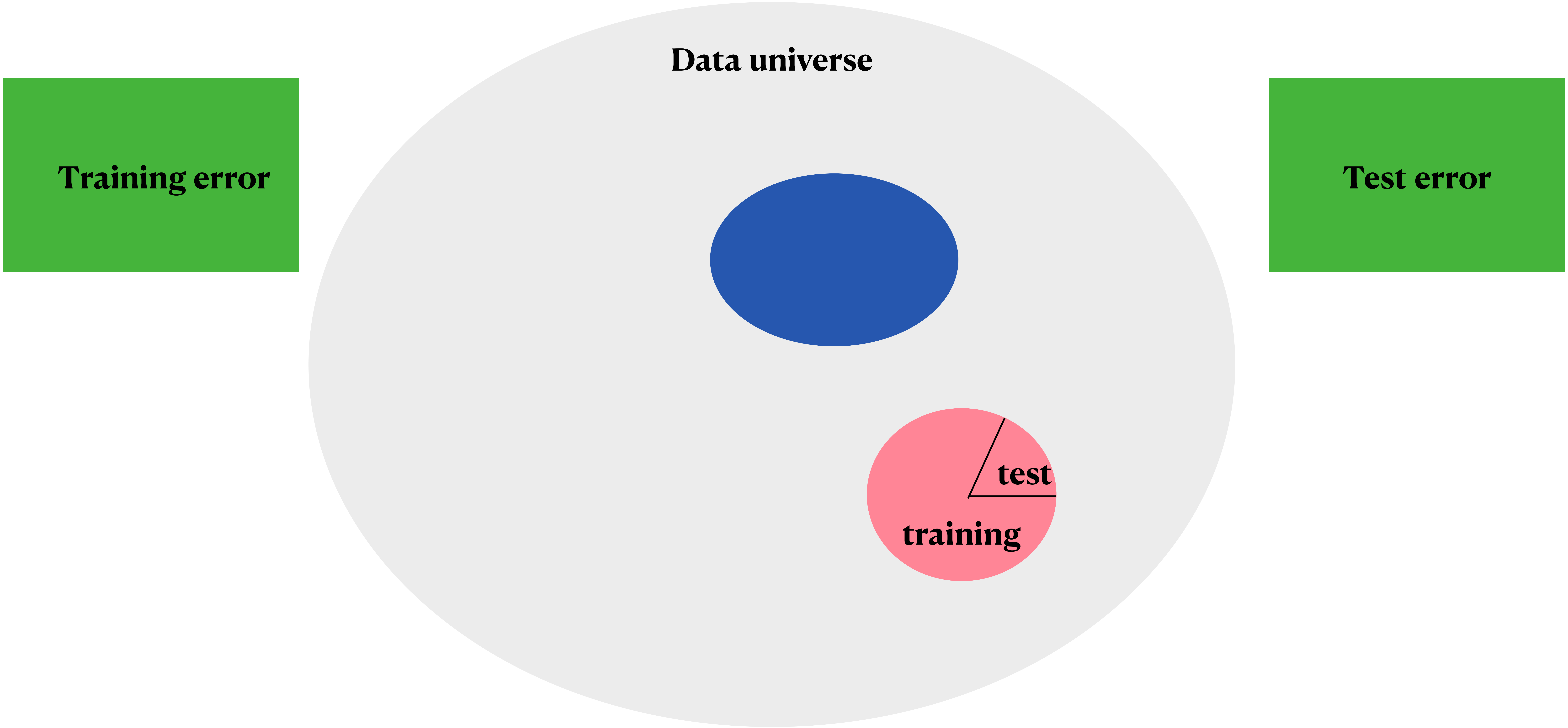
$$\frac{1}{N}\sum_{i=1}^{N}\left[(True - Pred)^2\right]$$

Loss function

Training error and test error

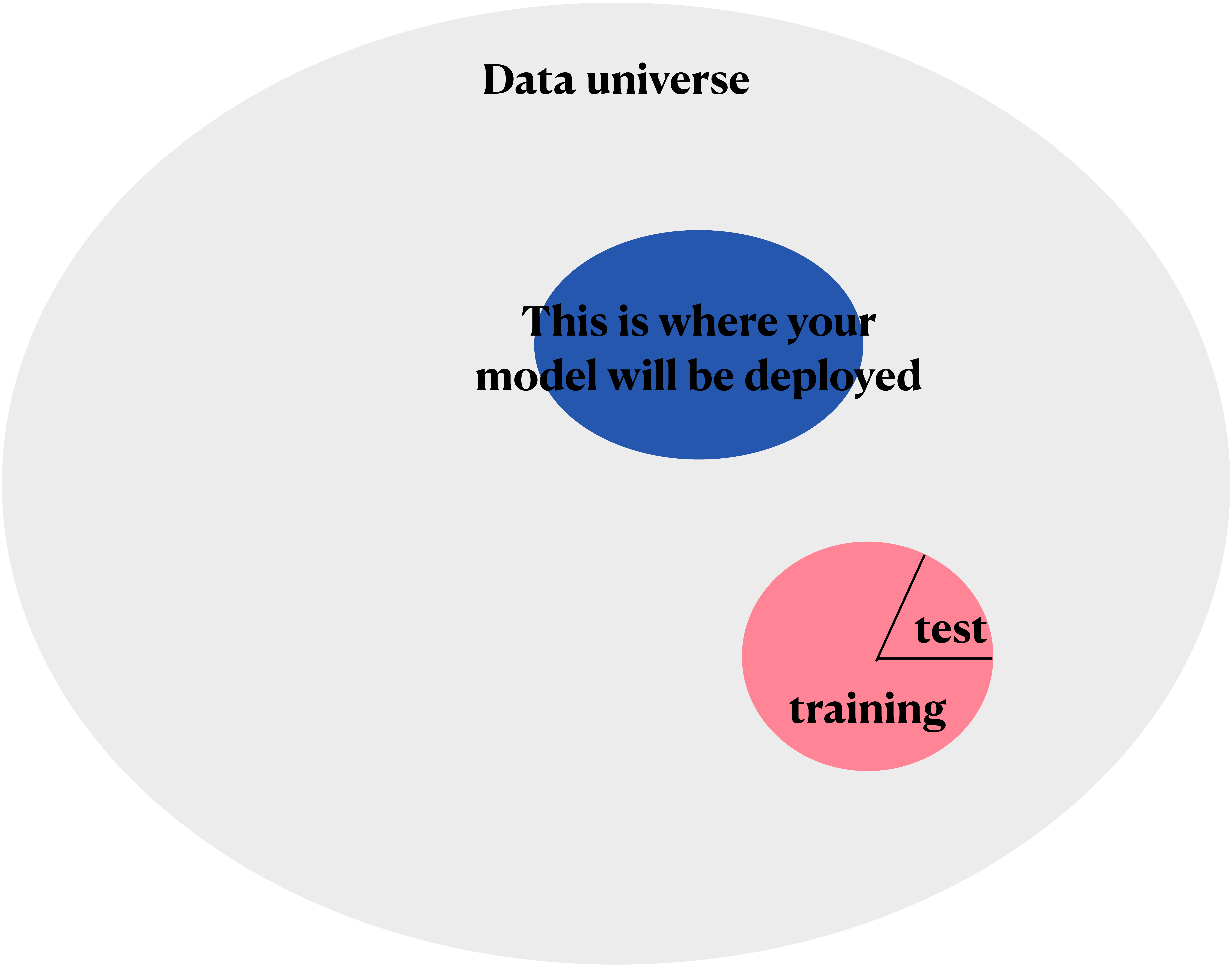


Training error and test error



Training error and test error

Training error



Test error

Unsupervised learning 1: k-means

Data without label

Feature1 (x1)	Feature2 (x2)	Feature3 (x3)	?
X[0, 0]	X[0, 1]	X[0, 2]	Empty
X[1, 0]	X[1, 1]	X[1, 2]	Empty
...	...	...	...
...	...	...	...
...	...	...	...

K-means: K and means

We can still build a model, but what to predict and what is loss function

X1	X2	X3
...	...	...
...	...	...
...	...	...
...	...	...

K-means: K and means

We can still build a model, but what to predict and what is loss function

X1	X2	X3
...	...	...
...	...	...
...	...	...
...	...	...

$$\longrightarrow f(\mathbf{x}) = f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \longrightarrow$$

K-means: K and means

We can still build a model, but what to predict and what is loss function

X1	X2	X3
...	...	...
...	...	...
...	...	...
...	...	...

$$\longrightarrow f(\mathbf{x}) = f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \longrightarrow$$

Cluster index (Values in 0,1,2,...,K-1)
1
0
3
K-1

Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0

Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0

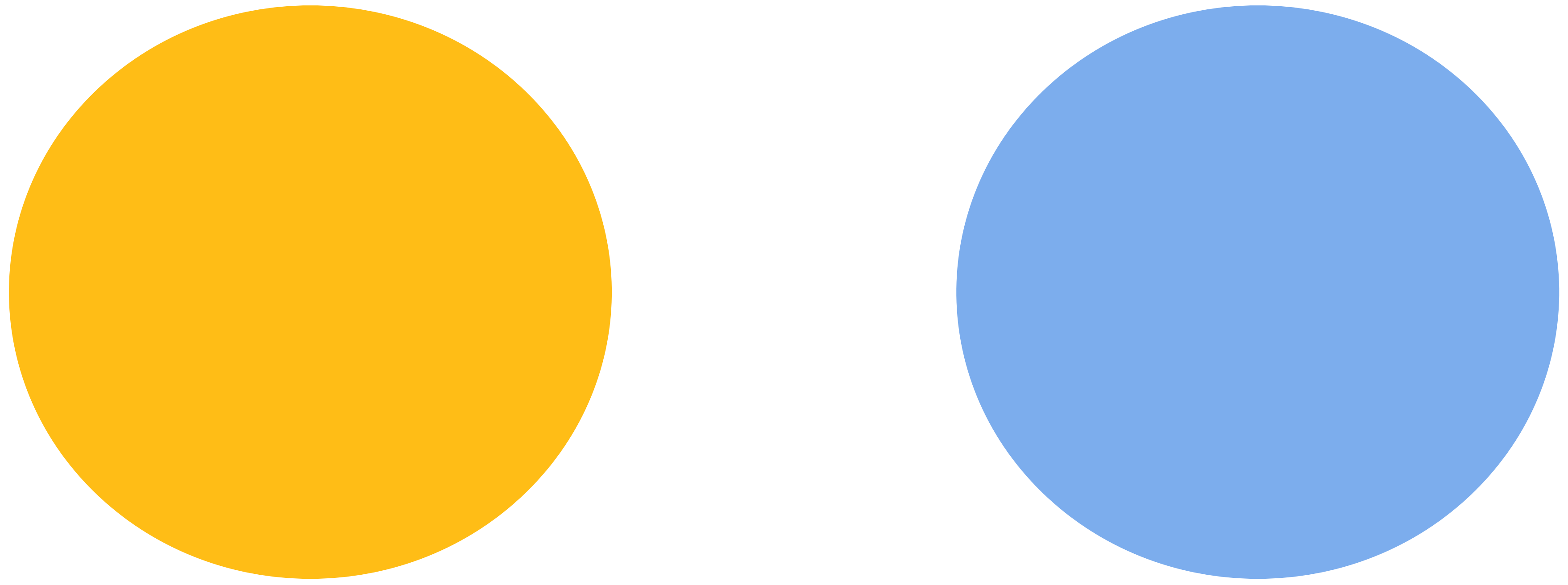
Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



Loss function: 2 steps

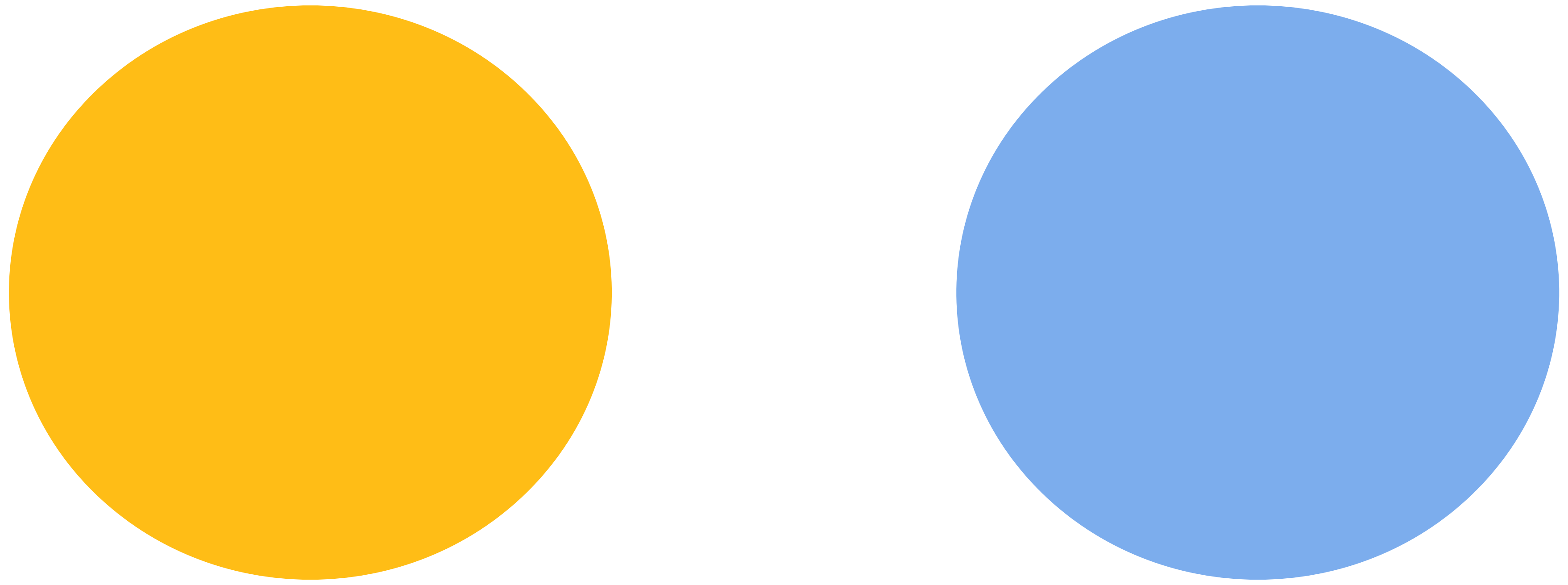
Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0

Cluster 0



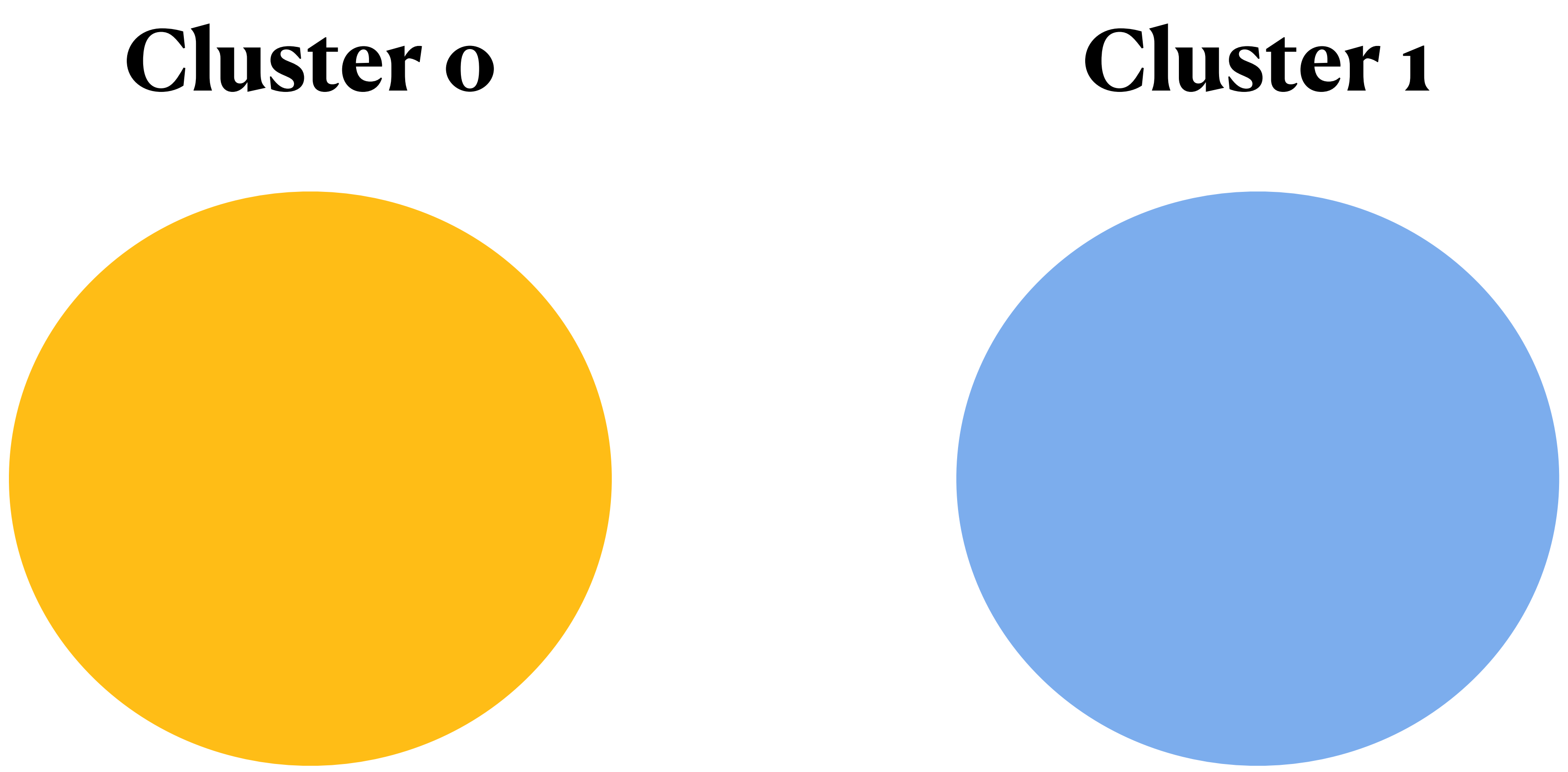
Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



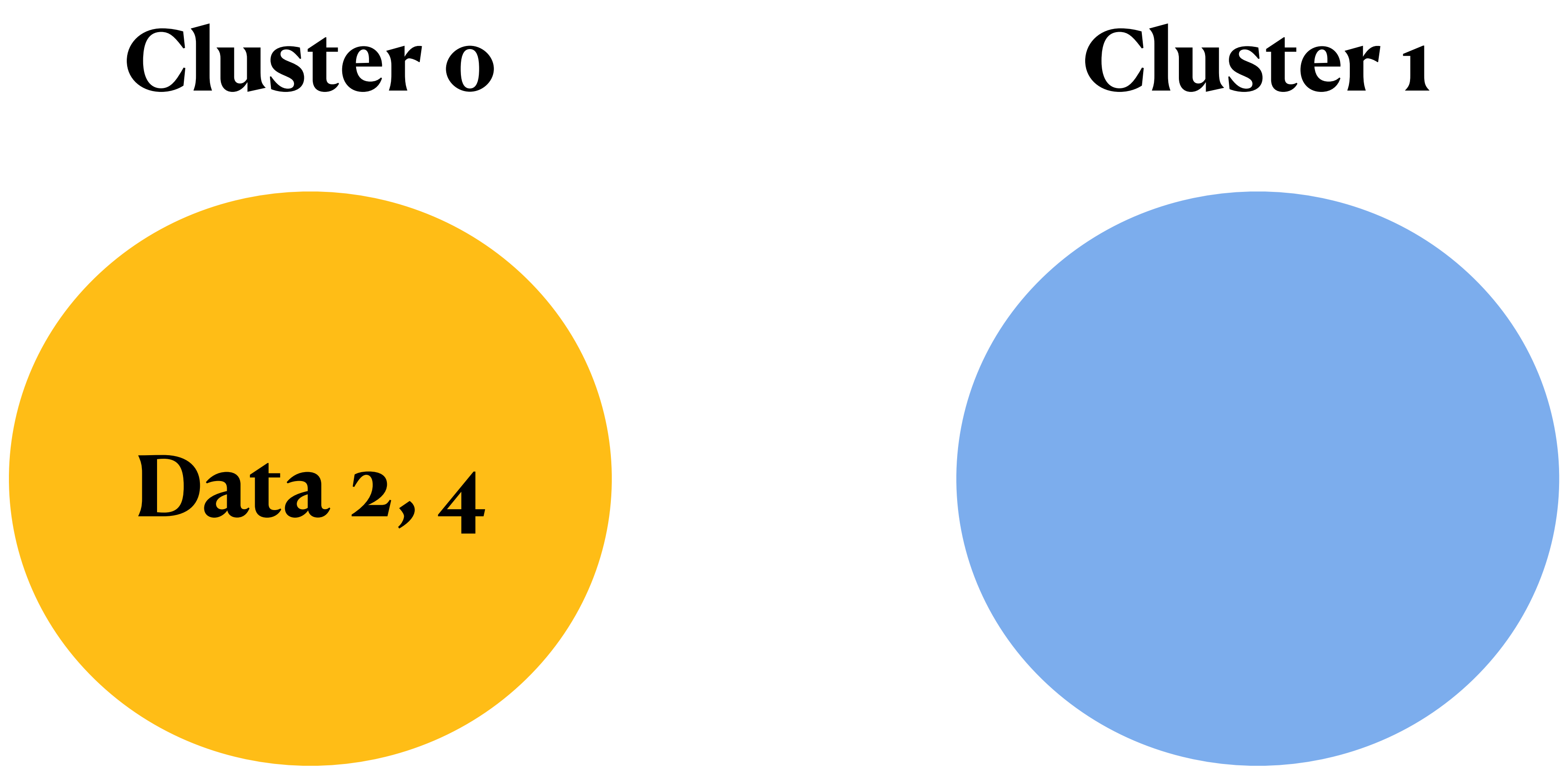
Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



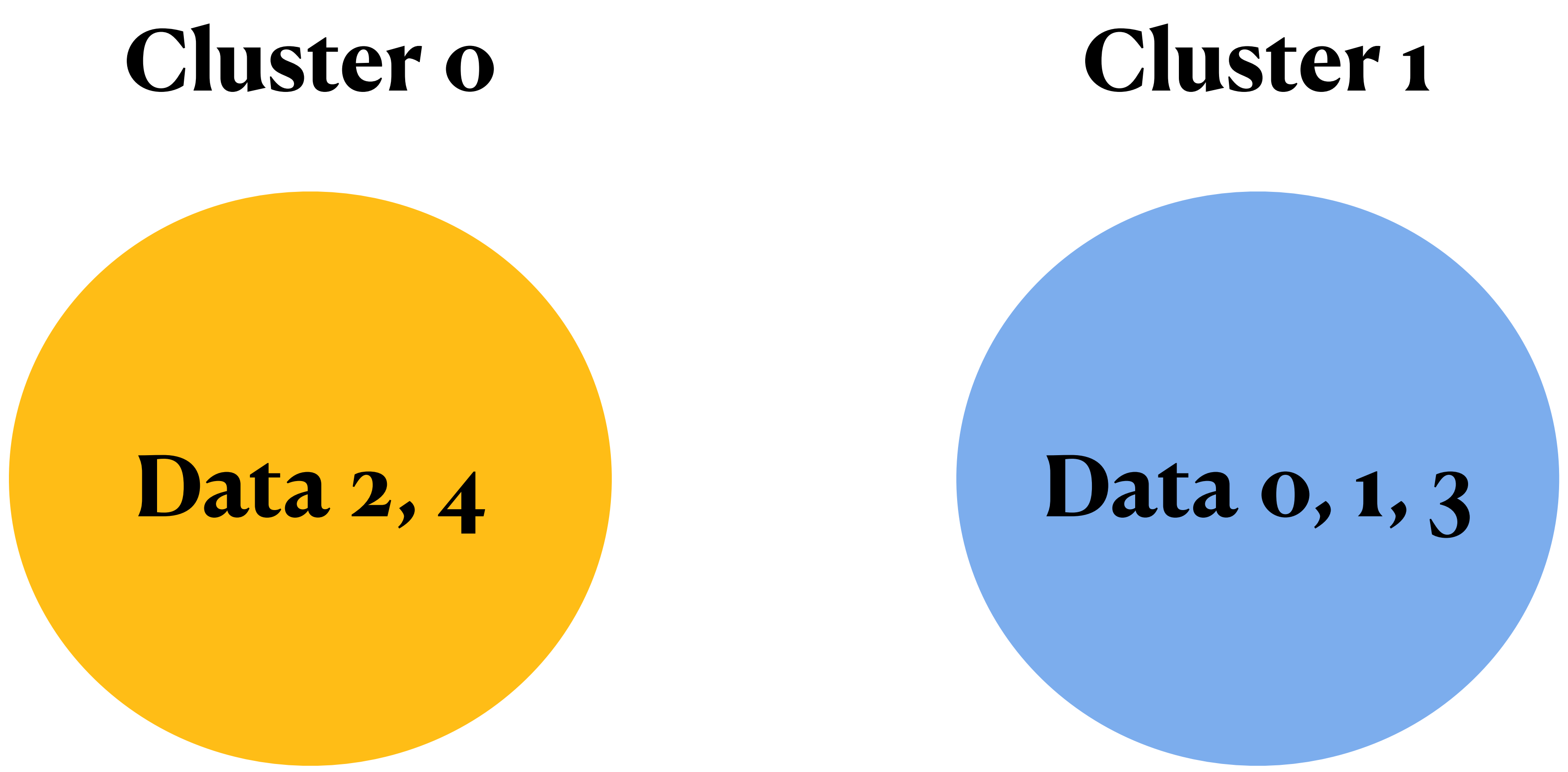
Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



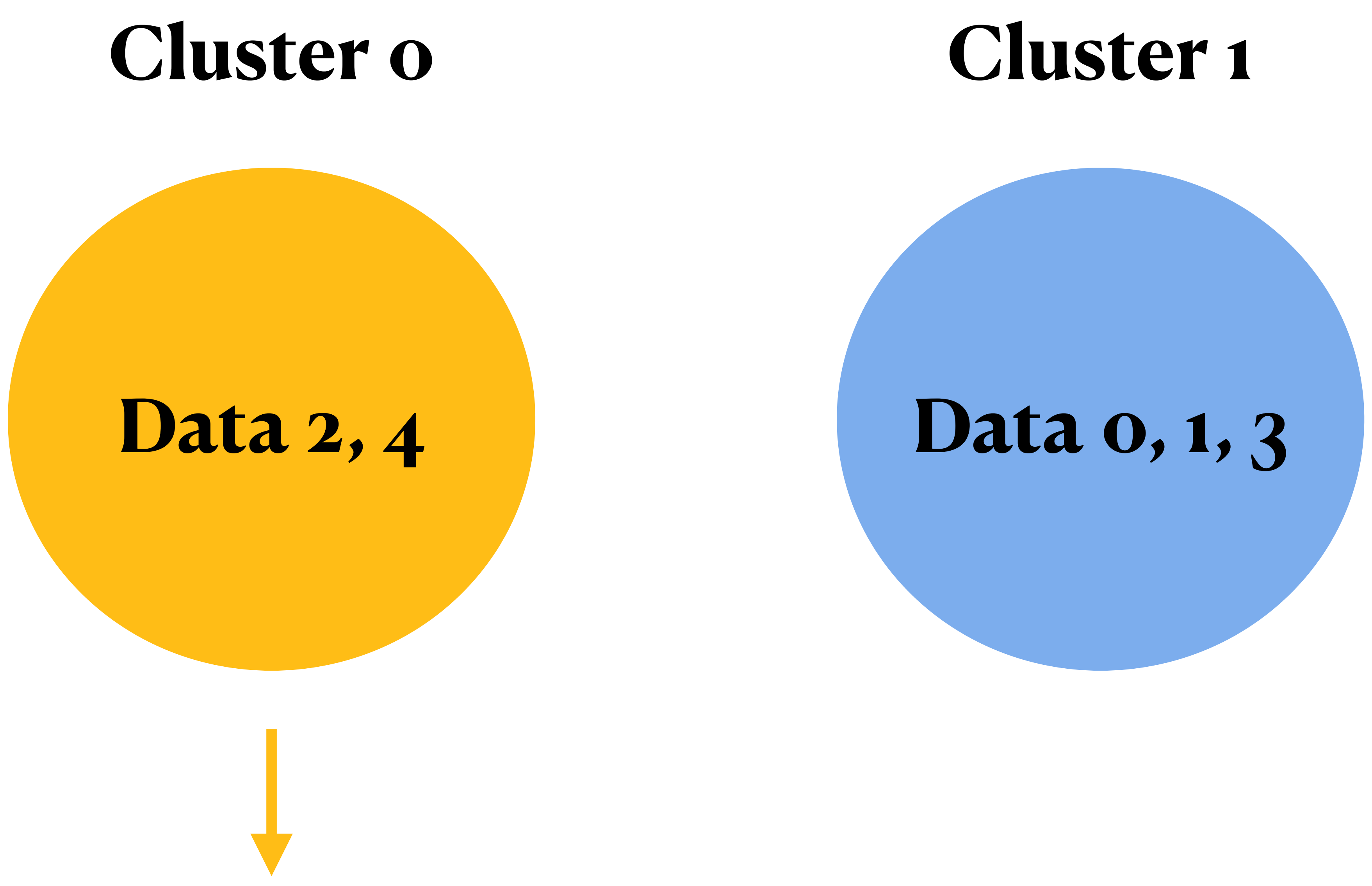
Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



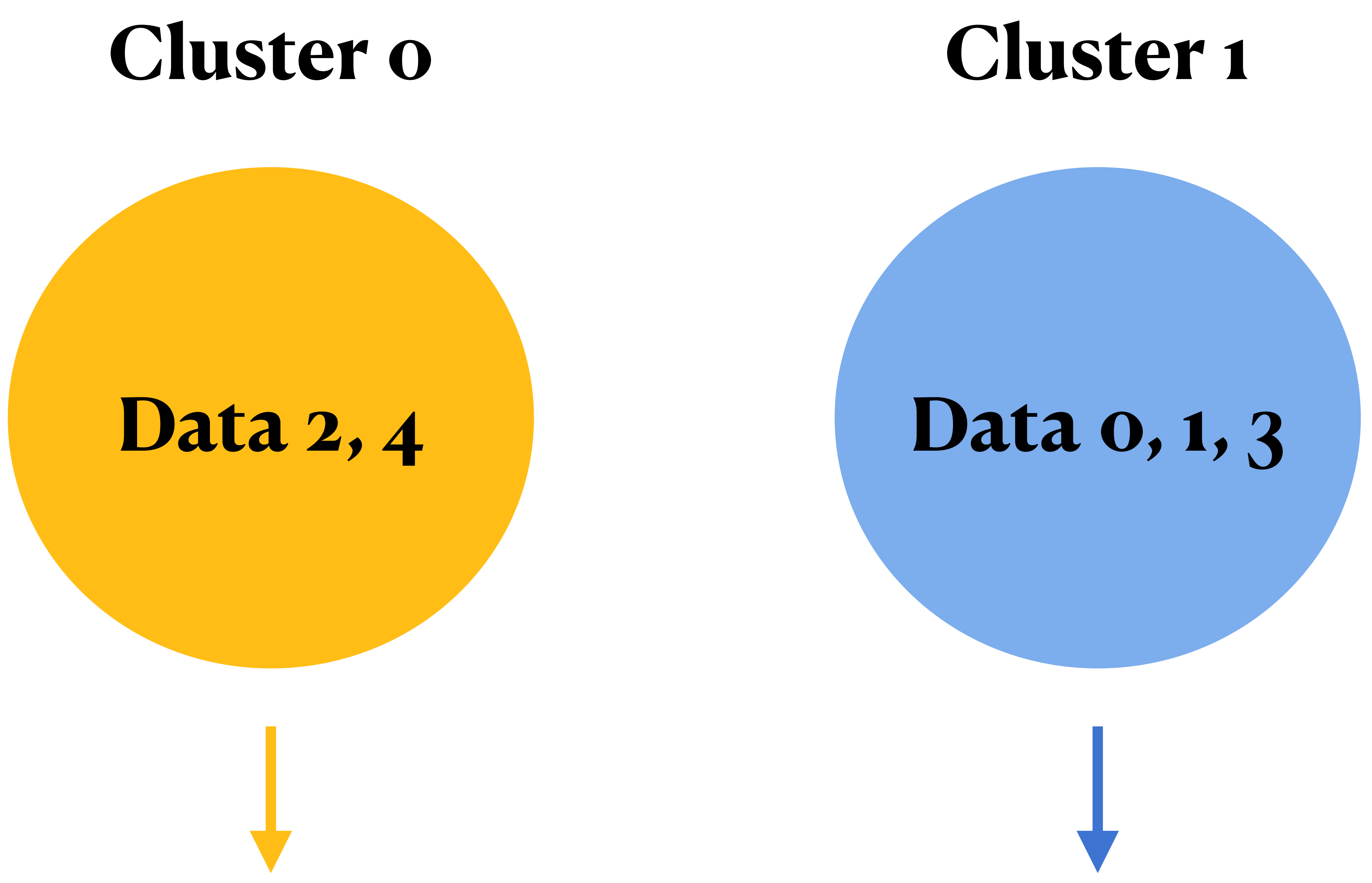
Loss function: 2 steps

Cluster means and cluster variance

**Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!**

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



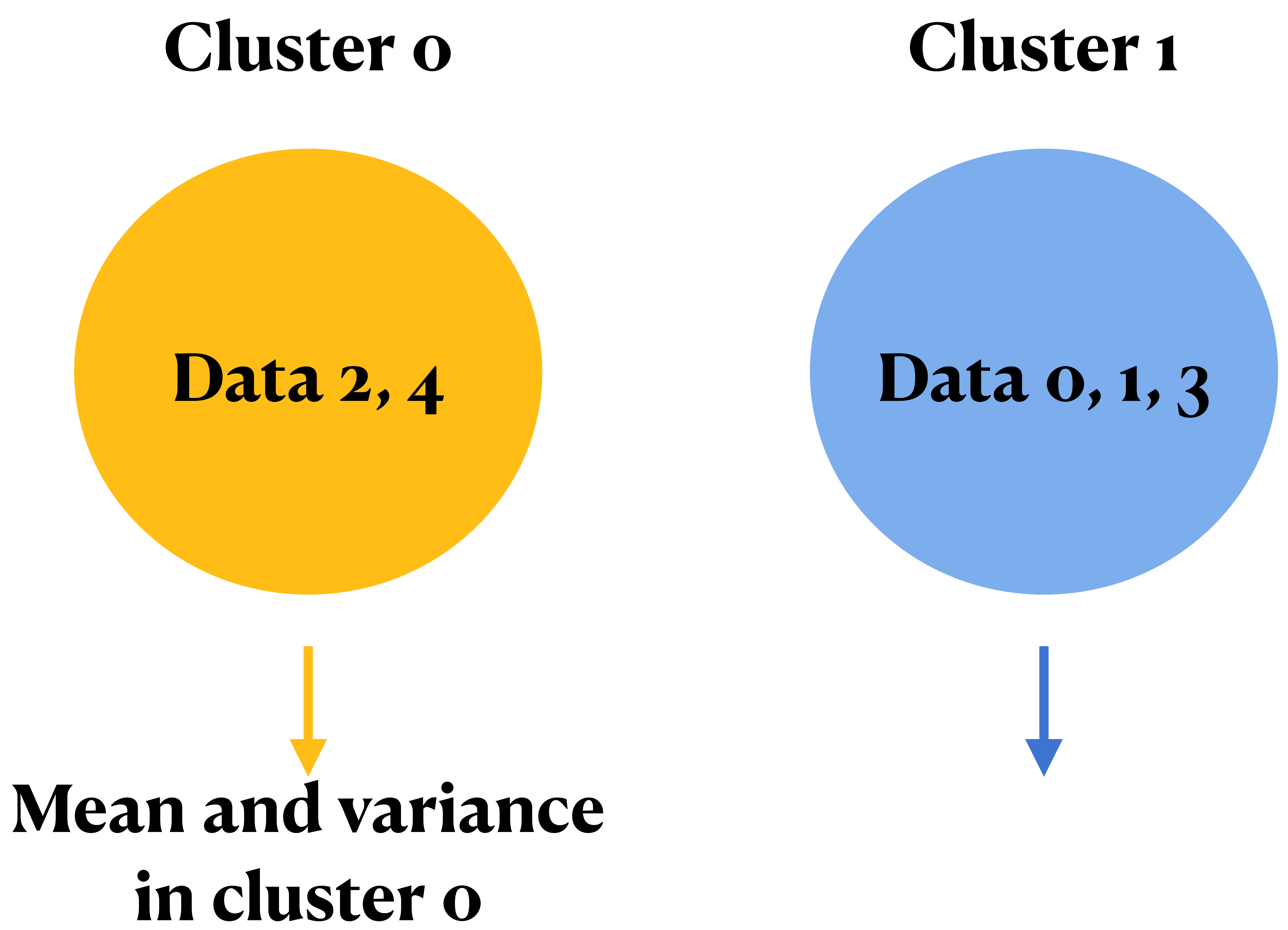
Loss function: 2 steps

Cluster means and cluster variance

Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



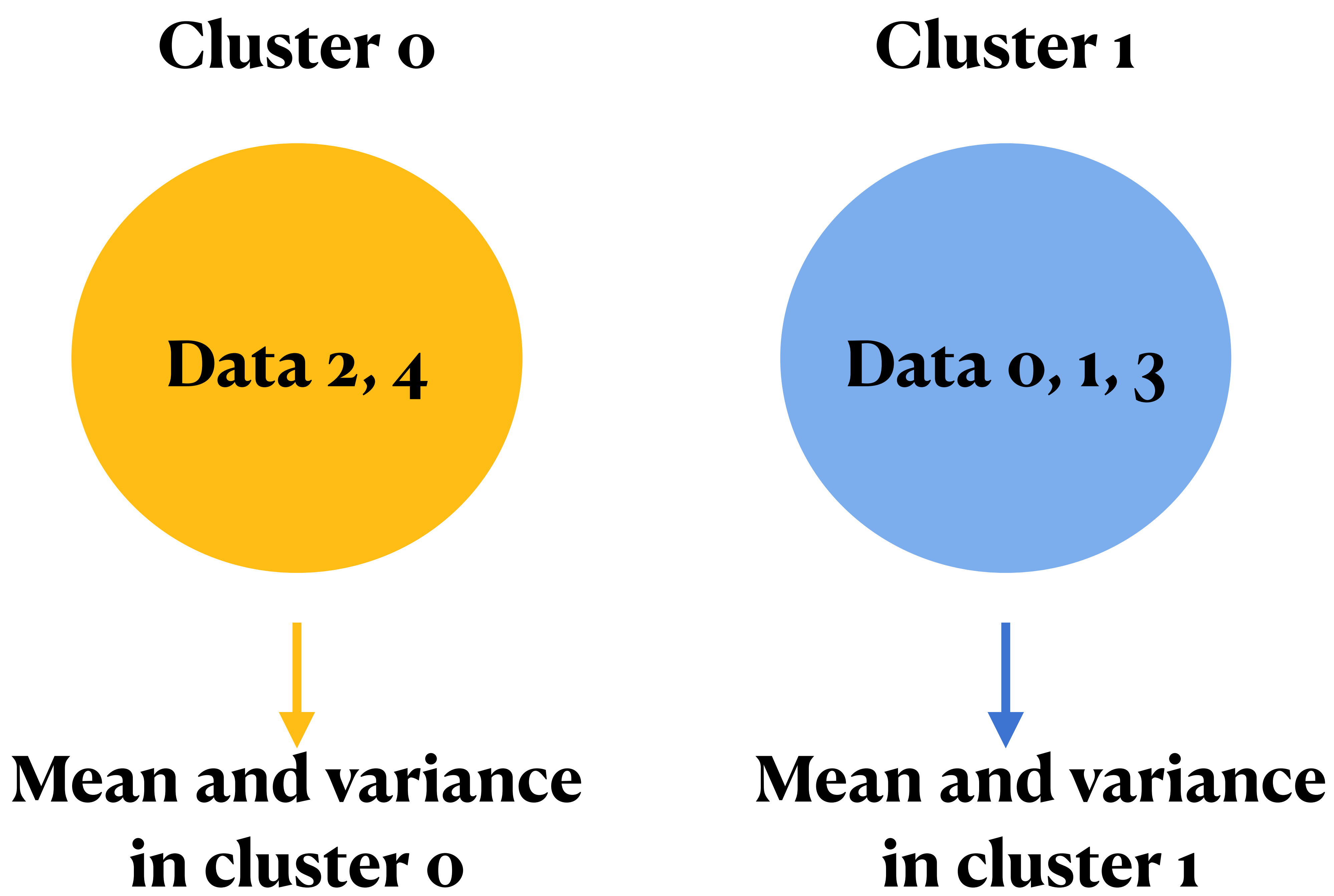
Loss function: 2 steps

Cluster means and cluster variance

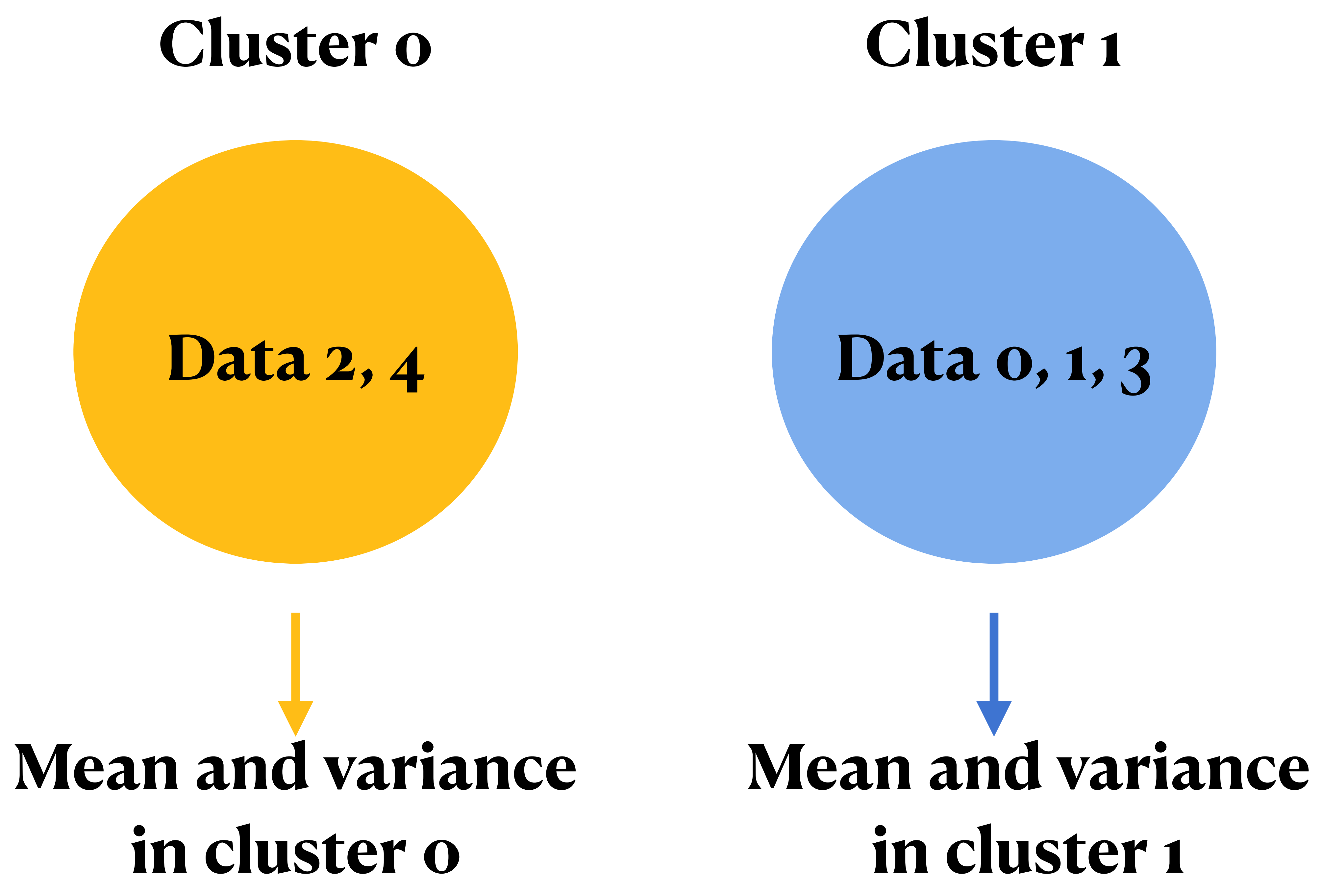
Very important: Every datum is associated with an integer in {0,1,2,..,K-1}
-> there are K clusters and each datum is associated with a cluster!

Example: We have 5 data points, each with two features. Assume there are two clusters (K=2)

Feature 1	Feature 2	Cluster index
		1
		1
		0
		1
		0



$$\textbf{Loss function} = \sum_i \mathrm{Var}[\mathrm{cluster} = i]$$



$$\textbf{Loss function} = \sum_i \mathrm{Var}[\mathrm{cluster} = i]$$

Cluster 0

Data 2, 4

Mean and variance in cluster 0

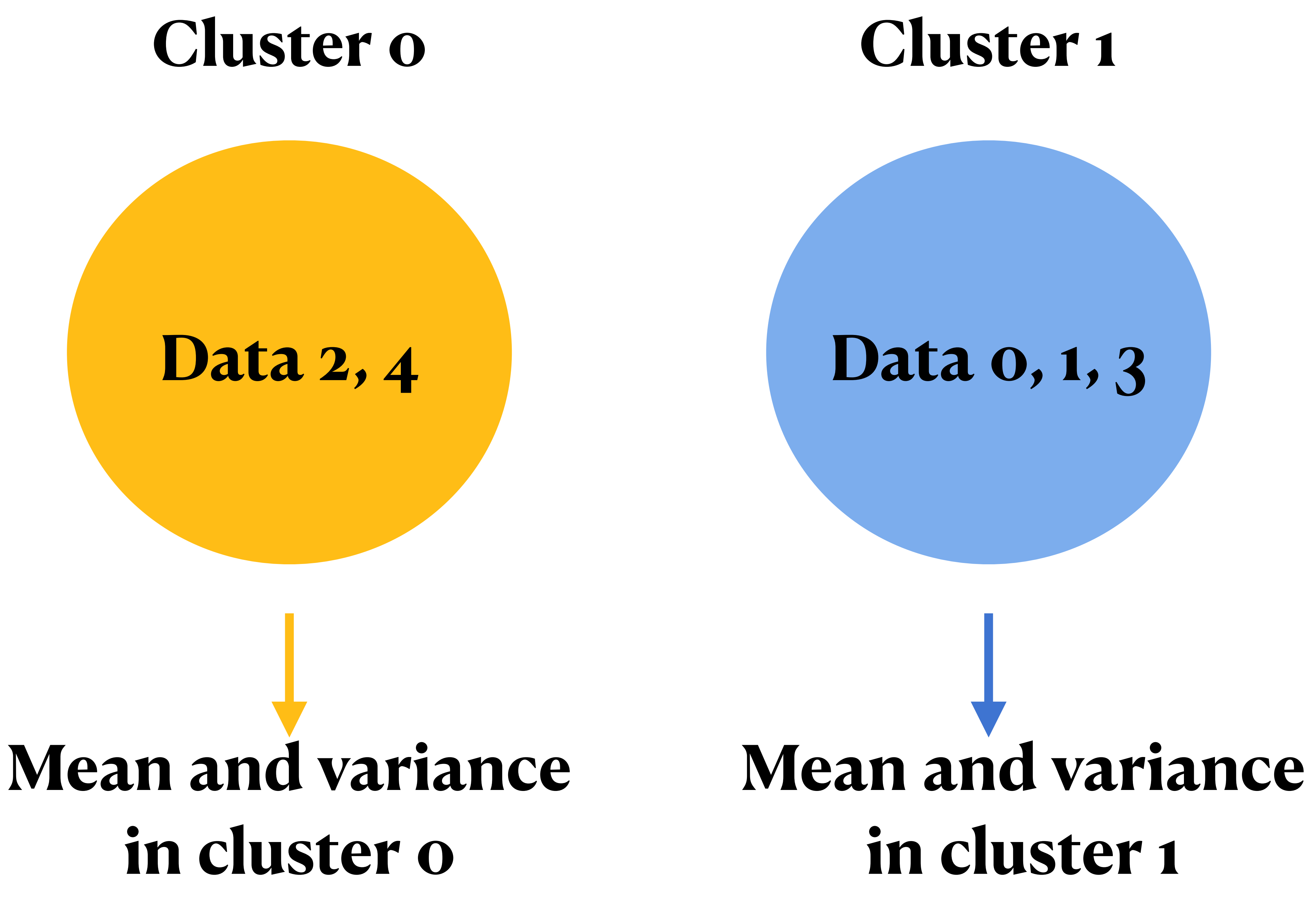
Cluster 1

Data 0, 1, 3

Mean and variance in cluster 1

In the end, we search for the cluster index for each datum that minimizes the total variance from all clusters!

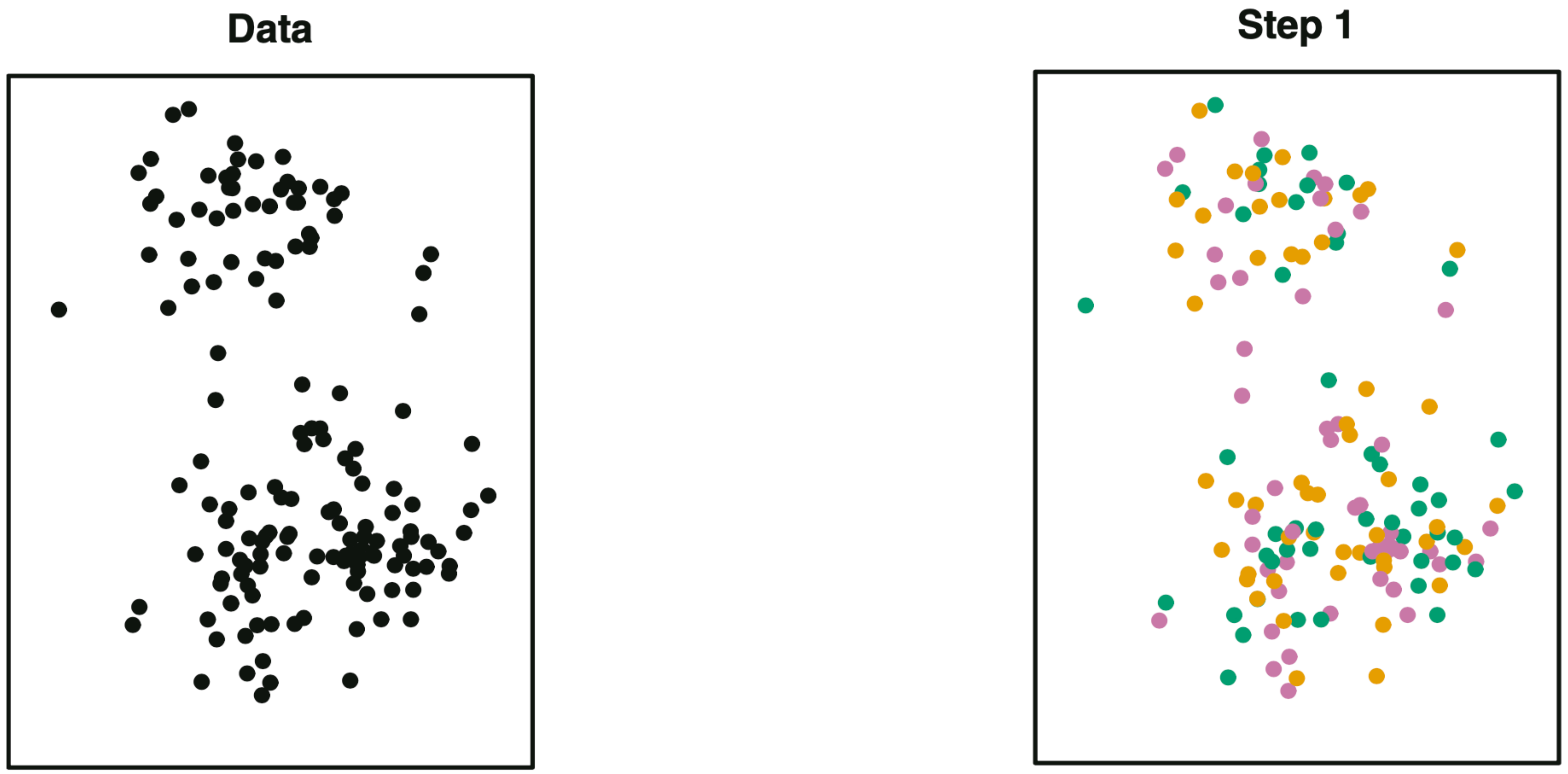
$$\textbf{Loss function} = \sum_i \mathrm{Var}[\mathrm{cluster} = i]$$

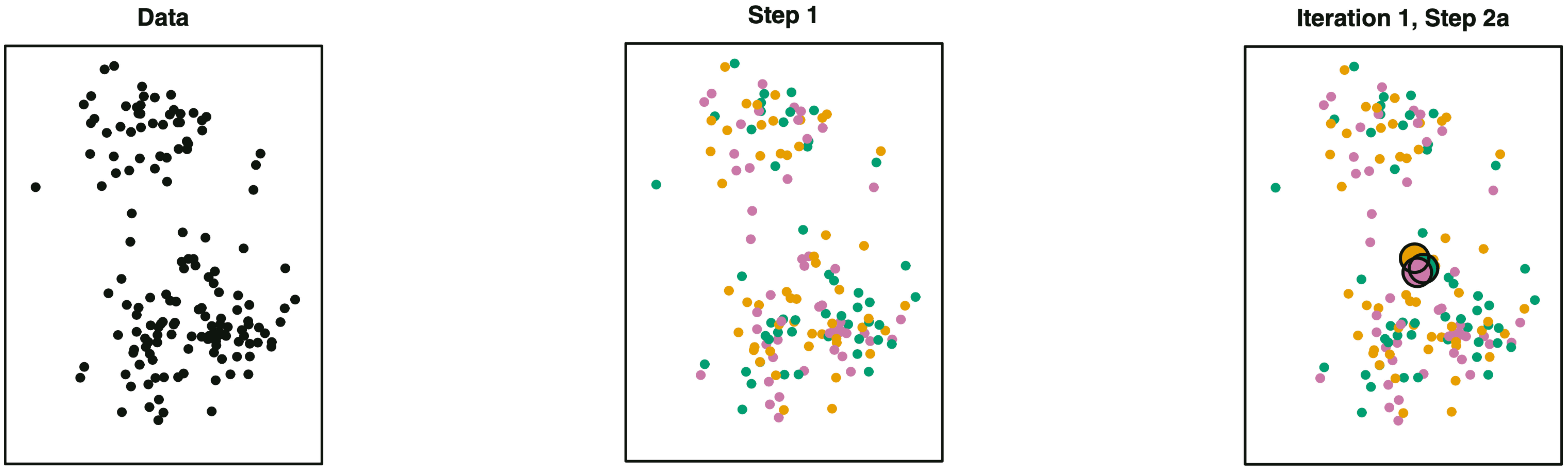


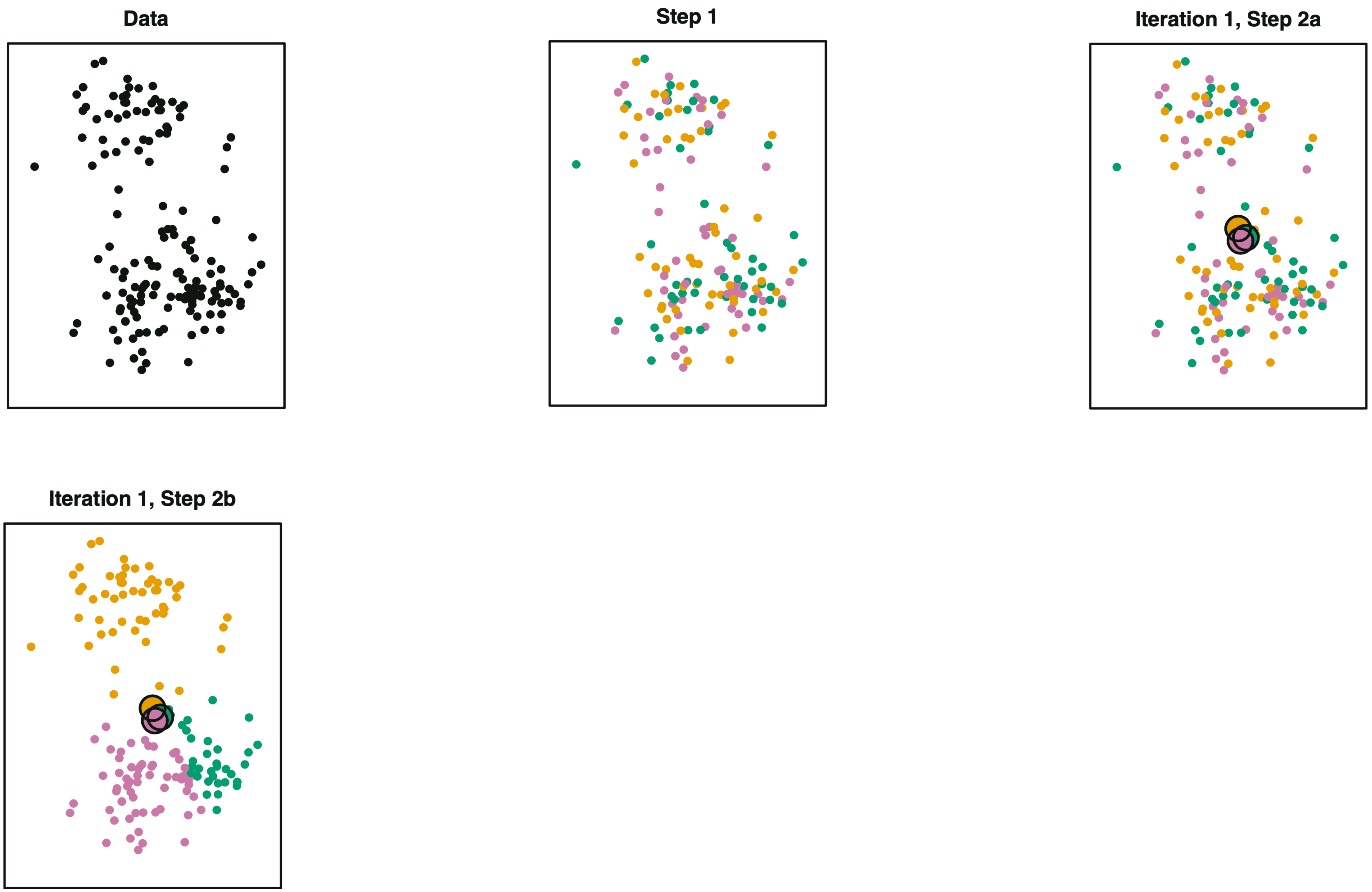
In the end, we search for the cluster index for each datum that minimizes the total variance from all clusters!

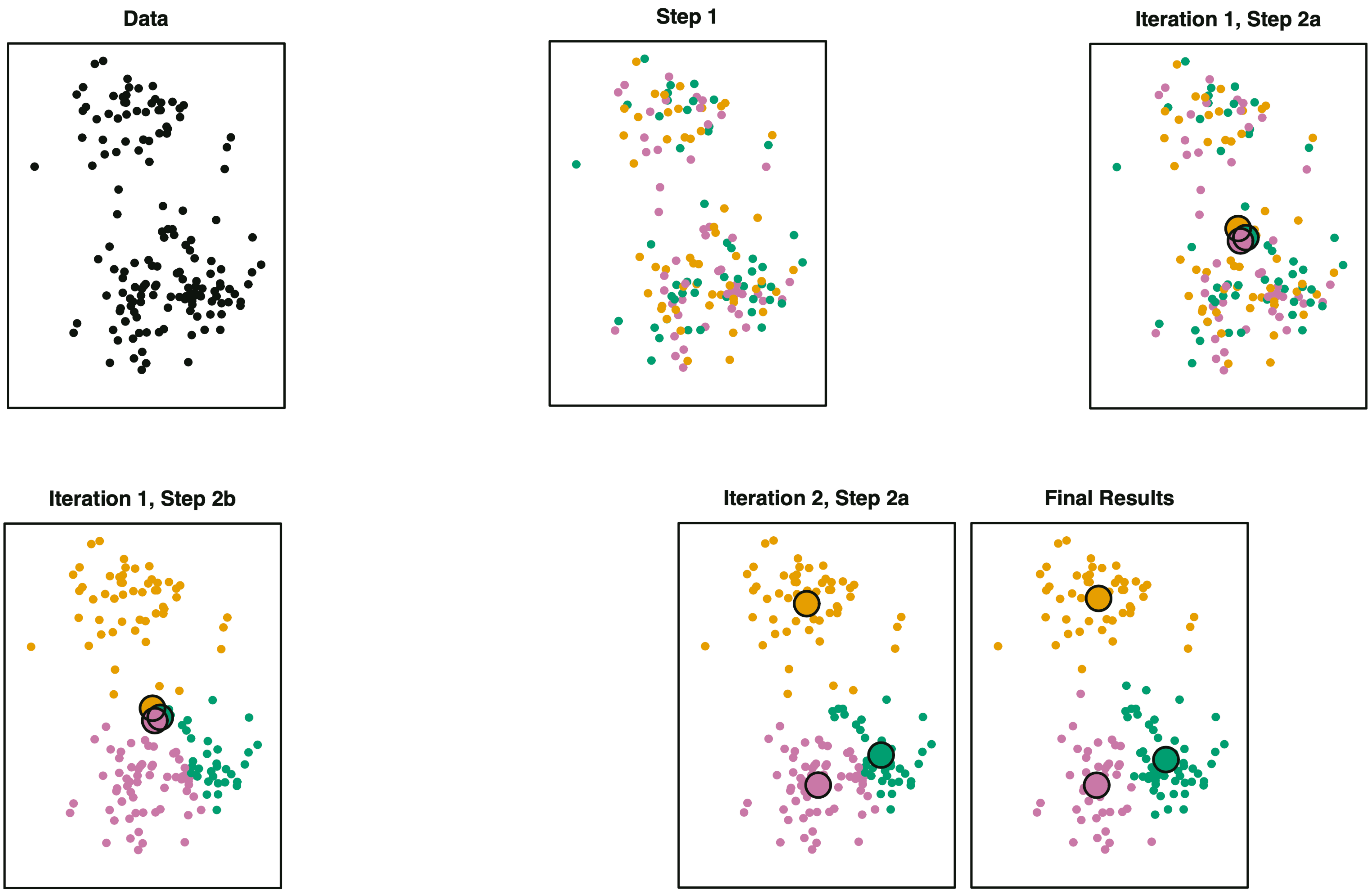
Then how do we do it? First method: enumerate all possibilities.
Second method: google 'EM'

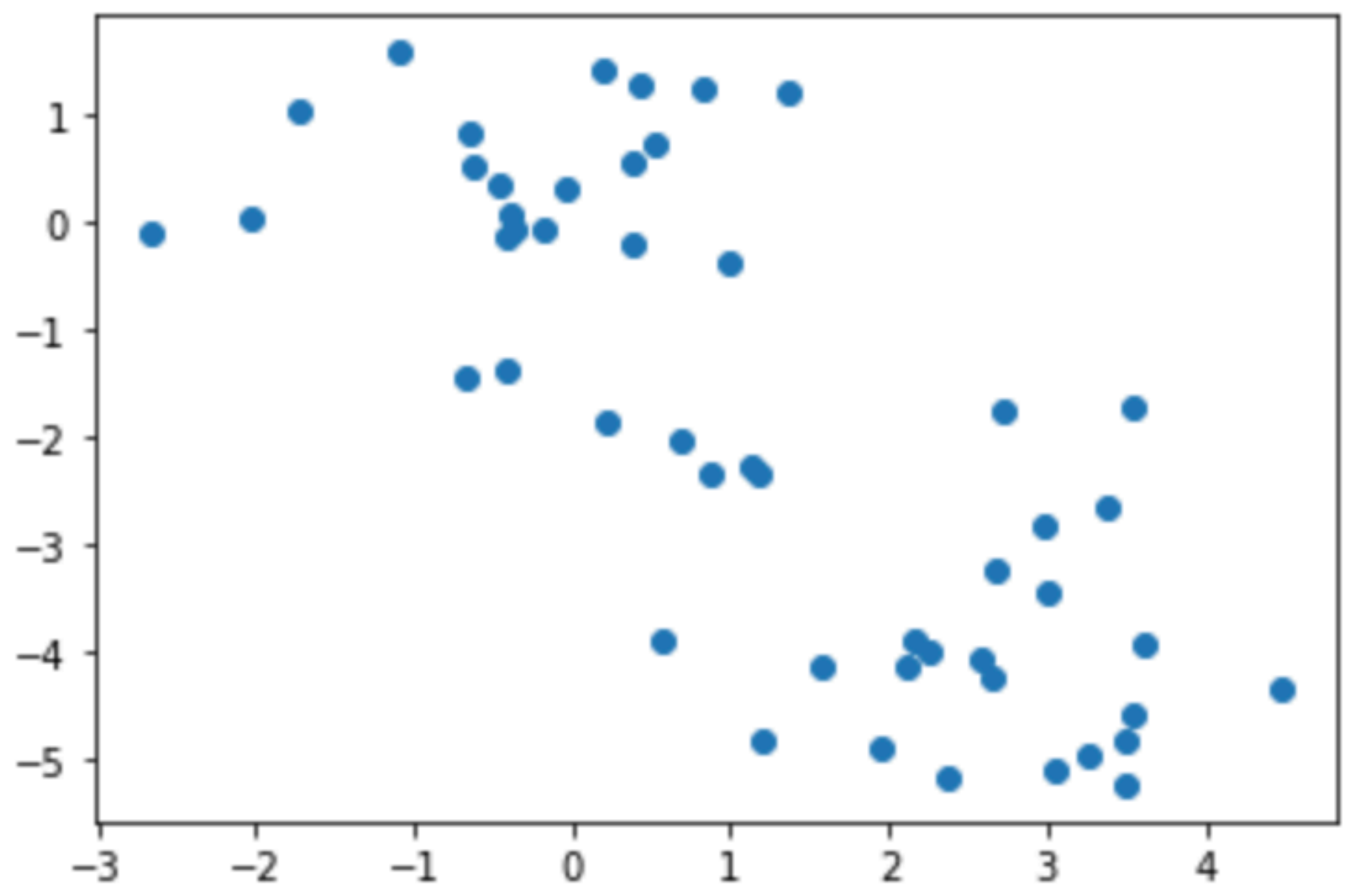
Data

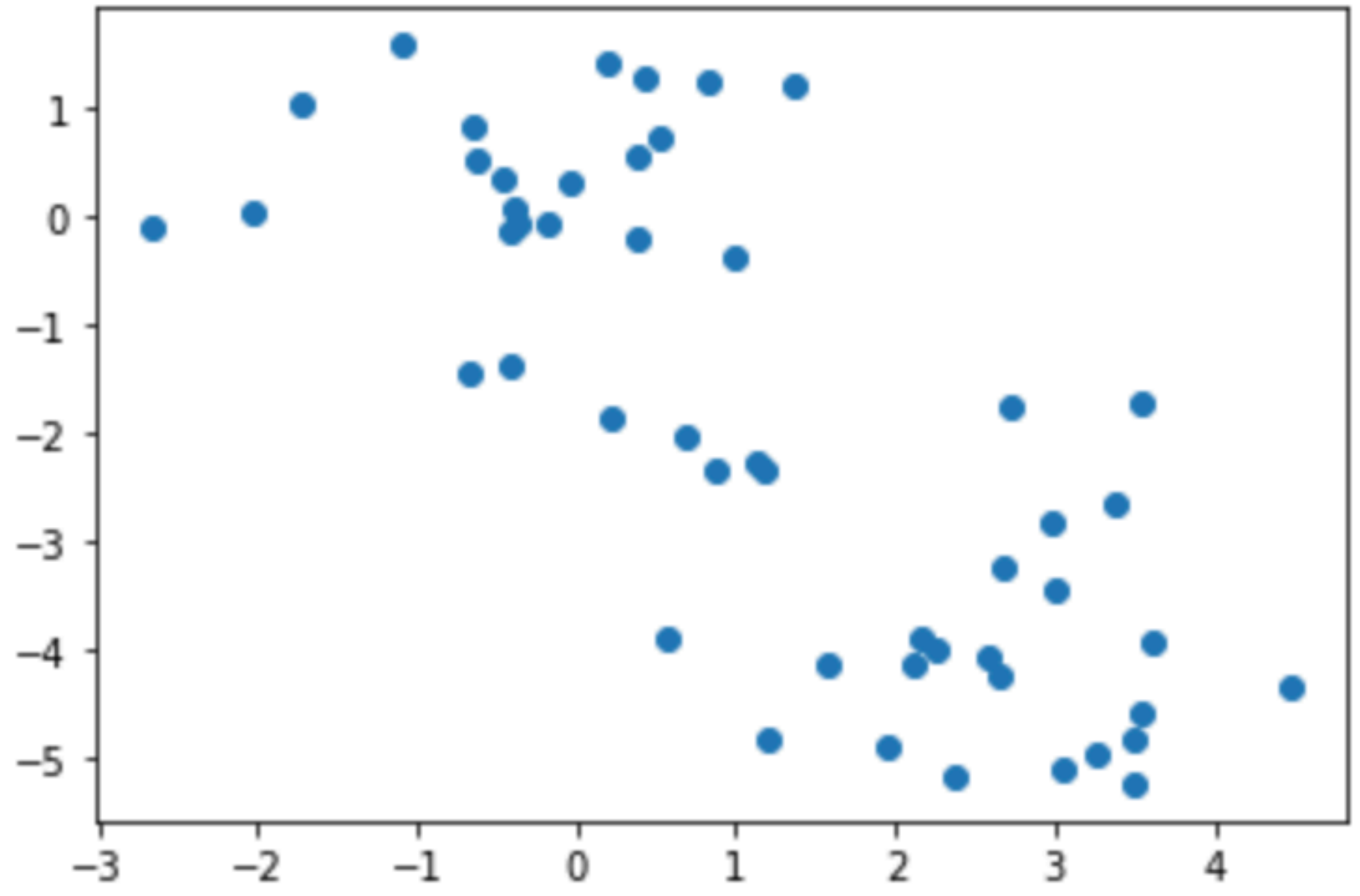

Data
Step 1


Data
Step 1
Iteration 1, Step 2a

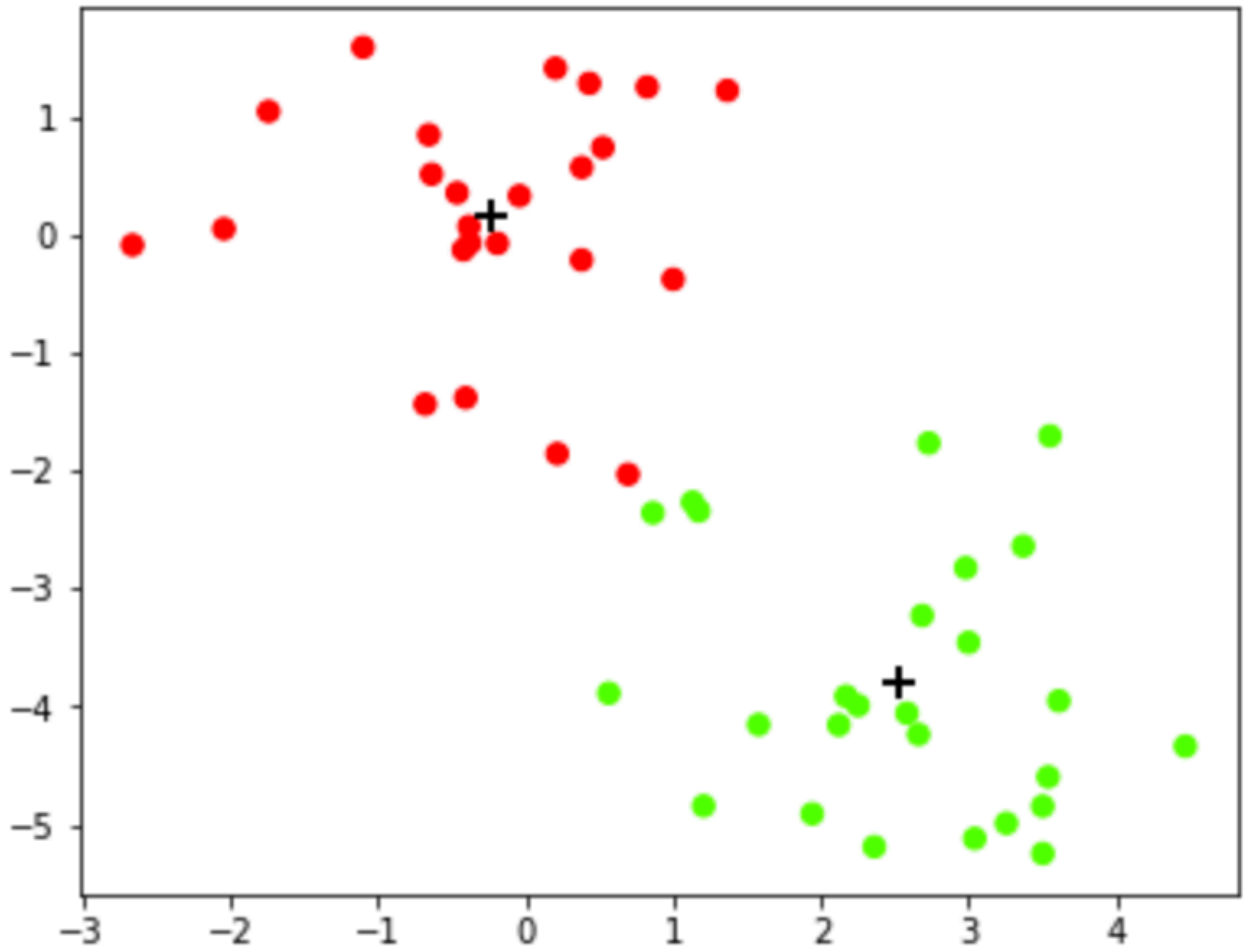

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b

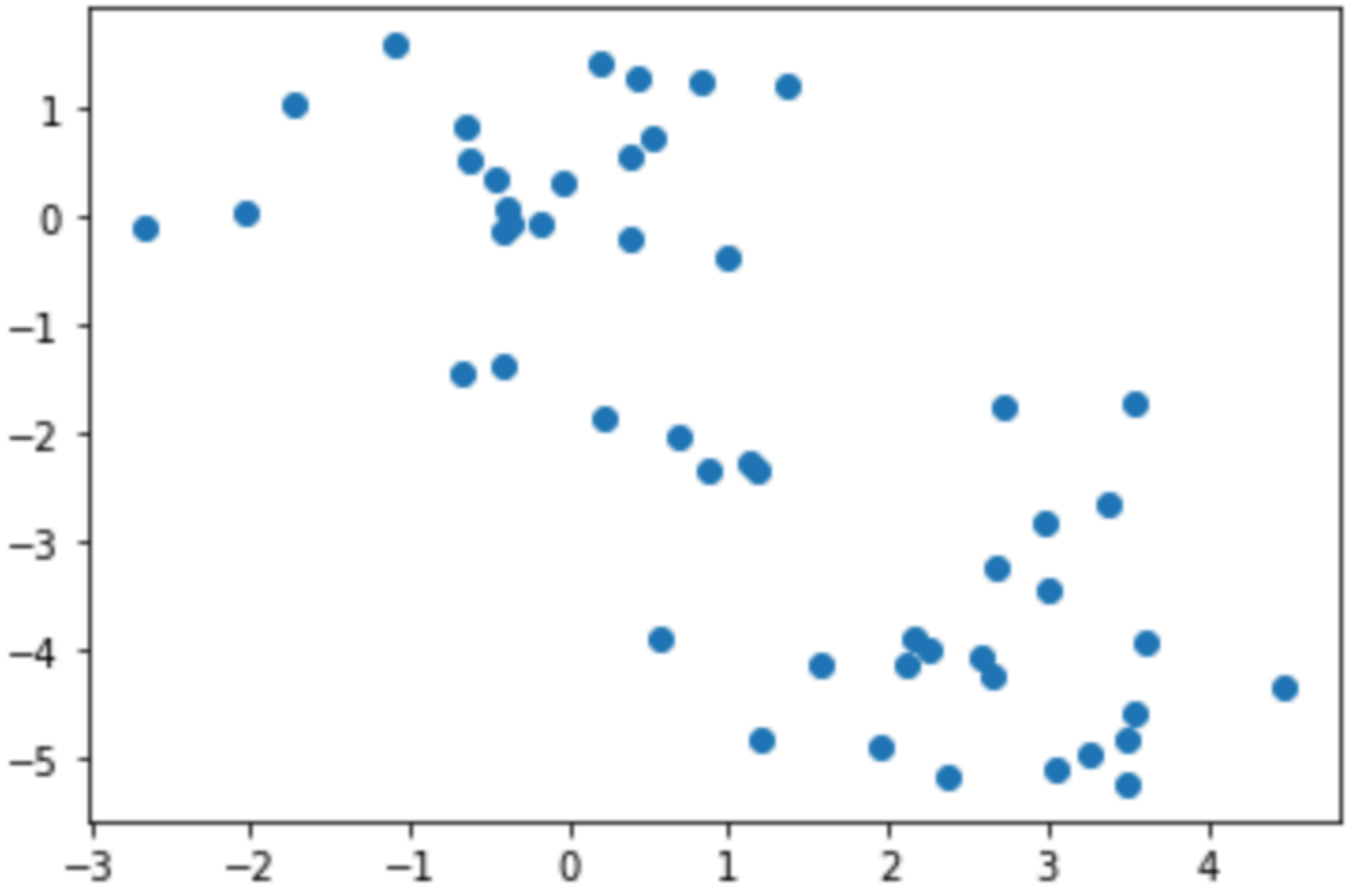

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results


1
0
−1
−2
−3
−4
−5
−3
−2
−1
0
1
2
3
4

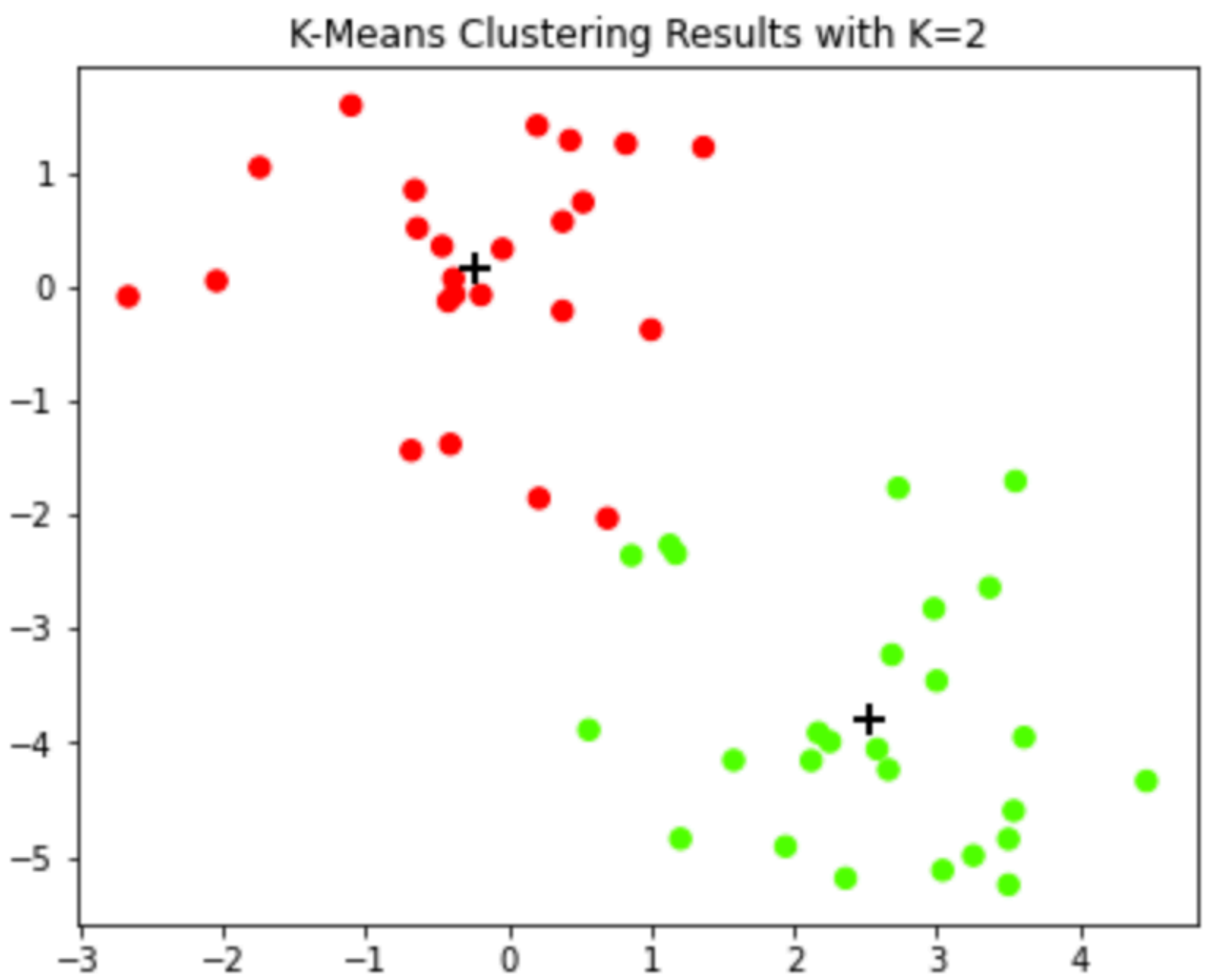


K-Means Clustering Results with K=2

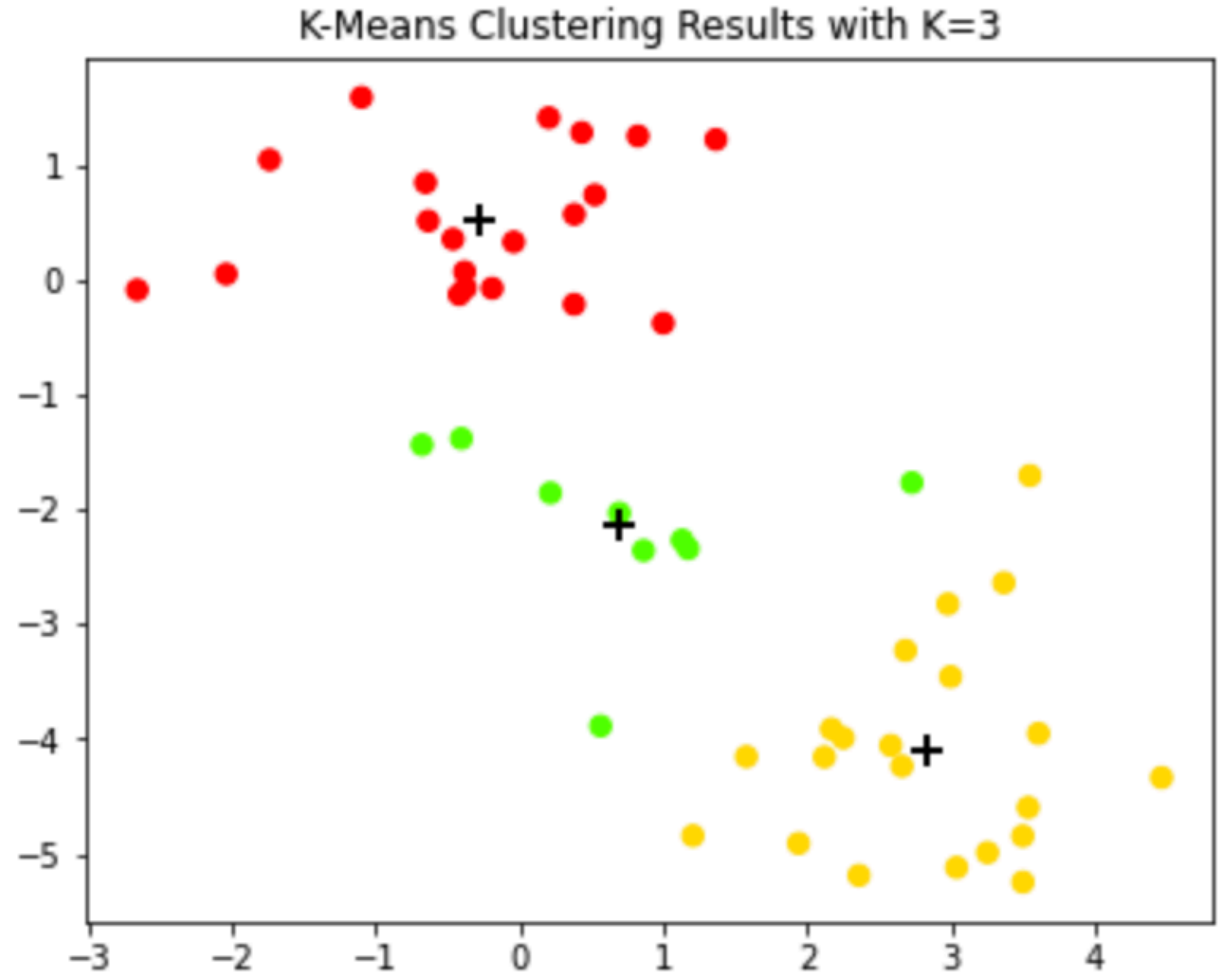




1
0
−1
−2
−3
−4
−5
−3 −2 −1 0 1 2 3 4



K-Means Clustering Results with K=2
1
0
−1
−2
−3
−4
−5
−3 −2 −1 0 1 2 3 4



K-Means Clustering Results with K=3
1
0
−1
−2
−3
−4
−5
−3 −2 −1 0 1 2 3 4

PCA: Principal component analysis

K-means: looking for minimum variance; PCA: looking for largest variance

Feature1	Feature2	Feature3	Feature4	....	Feature100

What is the problem with many features

What is the problem with many features

Very difficult to visualize (no one knows how to plot >4 dimensions)

What is the problem with many features

Very difficult to visualize (no one knows how to plot >4 dimensions)

We care about the label (?) for the data, many of the features are redundant!

What is the problem with many features

Very difficult to visualize (no one knows how to plot >4 dimensions)

We care about the label (?) for the data, many of the features are redundant!

A feature with large variance is a good feature!

What is the problem with many features

Very difficult to visualize (no one knows how to plot >4 dimensions)

We care about the label (?) for the data, many of the features are redundant!

A feature with large variance is a good feature!

PCA is a very simple yet elegant way of compressing data features.

PCA: using linear algebra to find new features with largest variance

Feature1	Feature2	Feature3	Feature4	……..	Feature100

PCA: using linear algebra to find new features with largest variance

Feature1	Feature2	Feature3	Feature4	……..	Feature100

PCA

PCA: using linear algebra to find new features with largest variance

Feature1	Feature2	Feature3	Feature4	....	Feature100

PCA

PCA: using linear algebra to find new features with largest variance

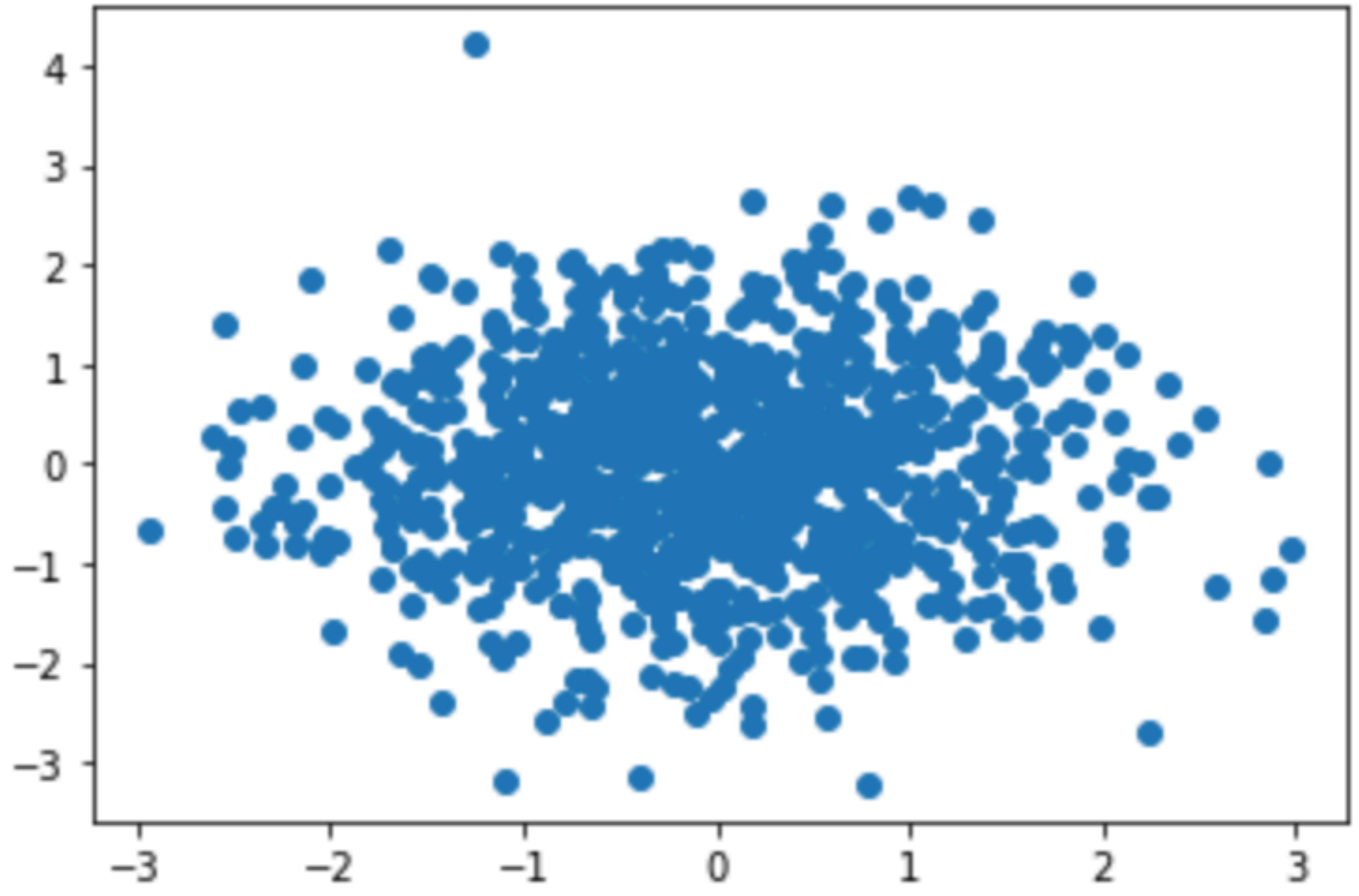
Feature1	Feature2	Feature3	Feature4	……..	Feature100

PCA

New feat.1	New feat.2

Very simple case of PCA

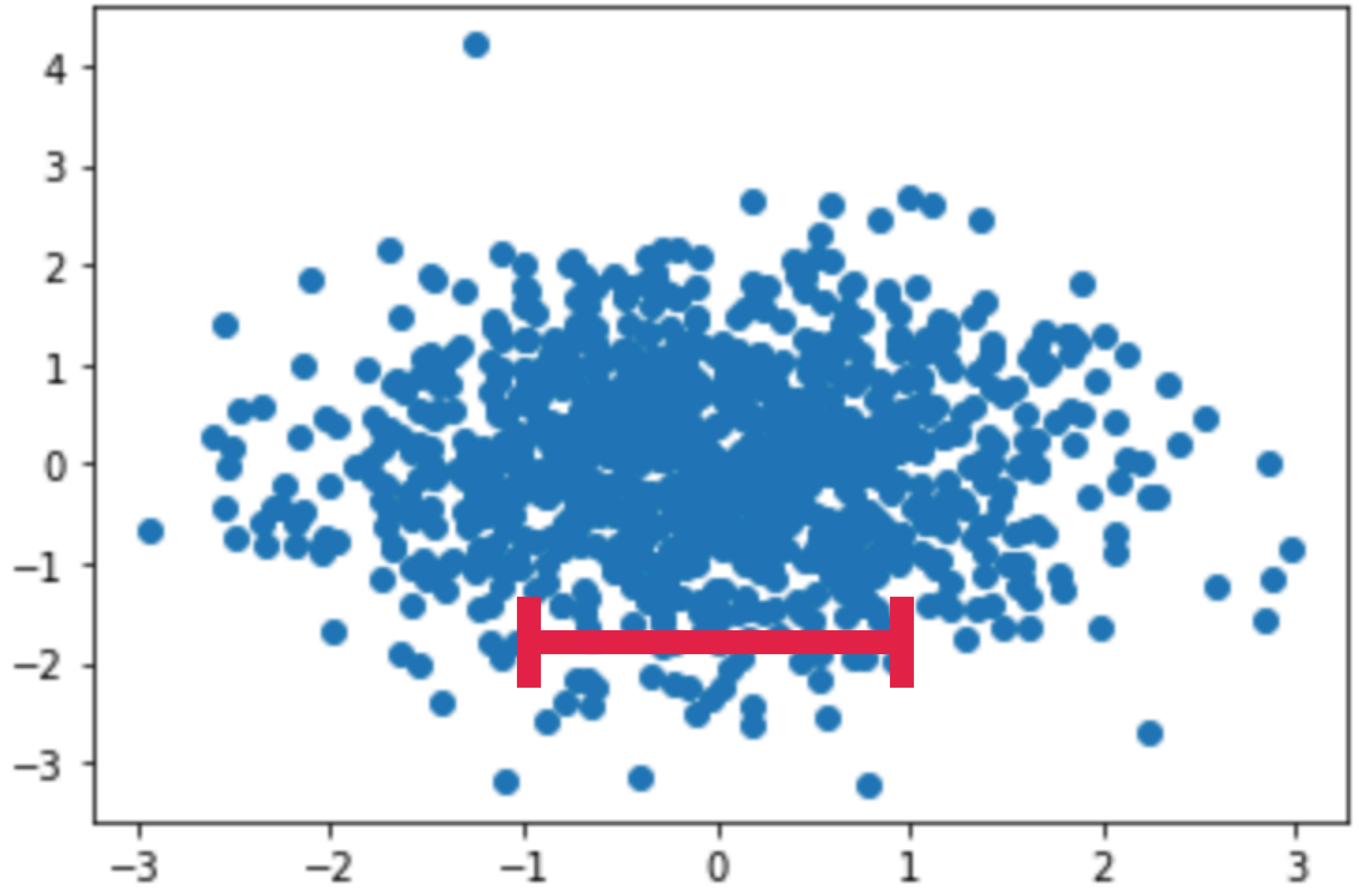
Two dimensions to two dimensions




```
In [13]: np.var(X0,axis=0)
Out[13]: array([0.9569638 , 0.97715073])
```

Very simple case of PCA

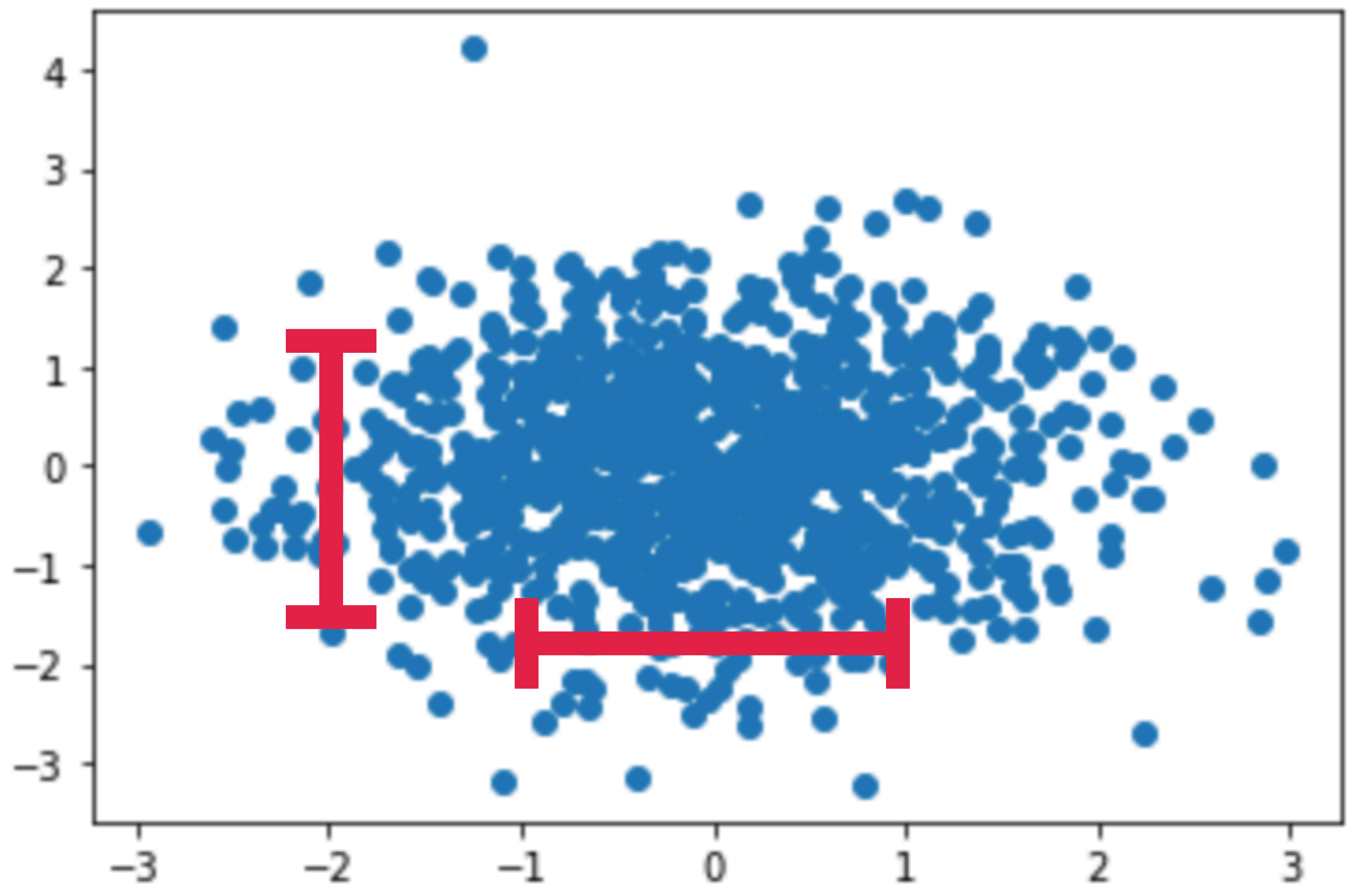
Two dimensions to two dimensions




```
In [13]: np.var(X0,axis=0)
Out[13]: array([0.9569638 , 0.97715073])
```

Very simple case of PCA

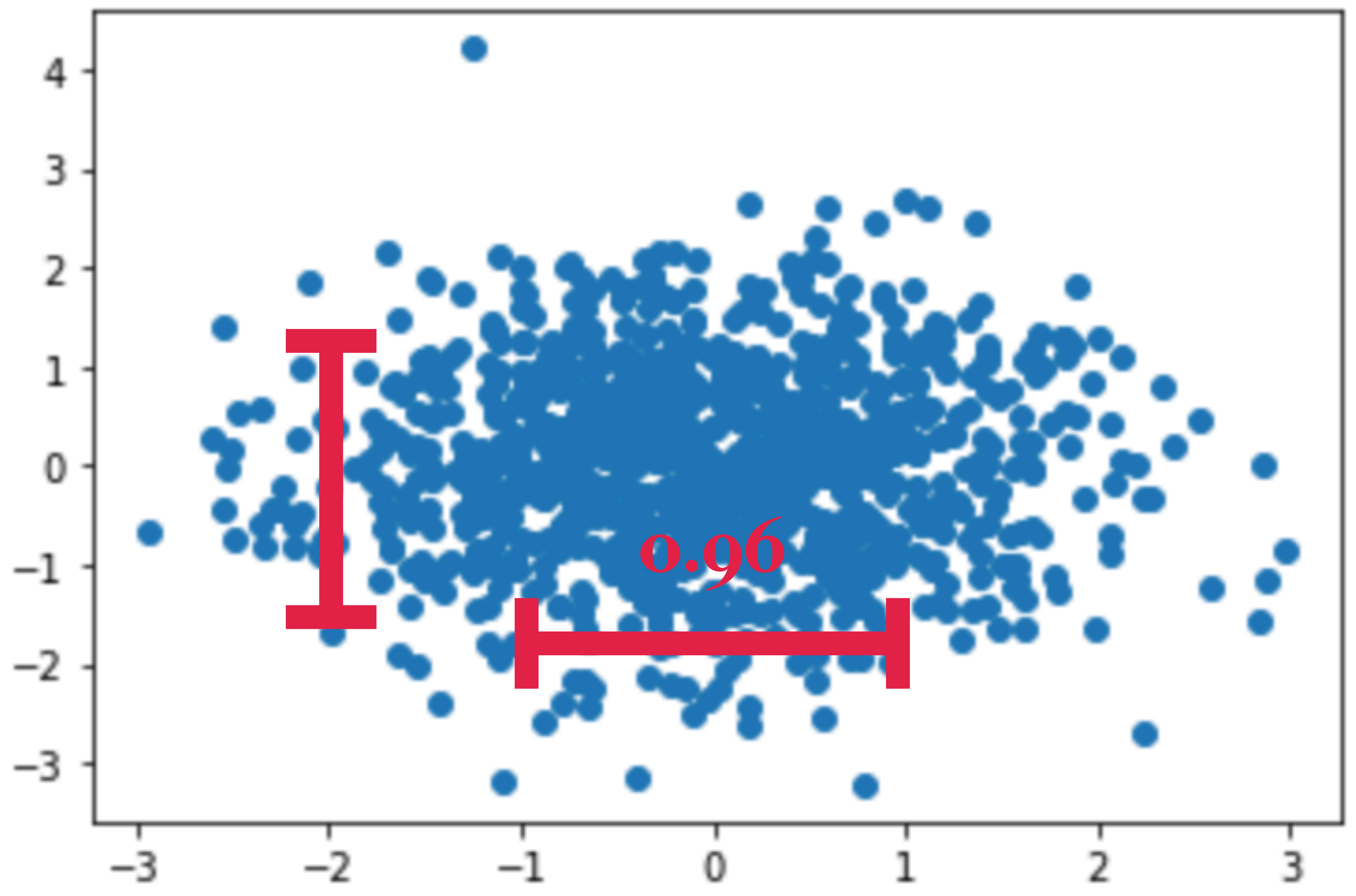
Two dimensions to two dimensions




```
In [13]: np.var(X0,axis=0)
Out[13]: array([0.9569638 , 0.97715073])
```

Very simple case of PCA

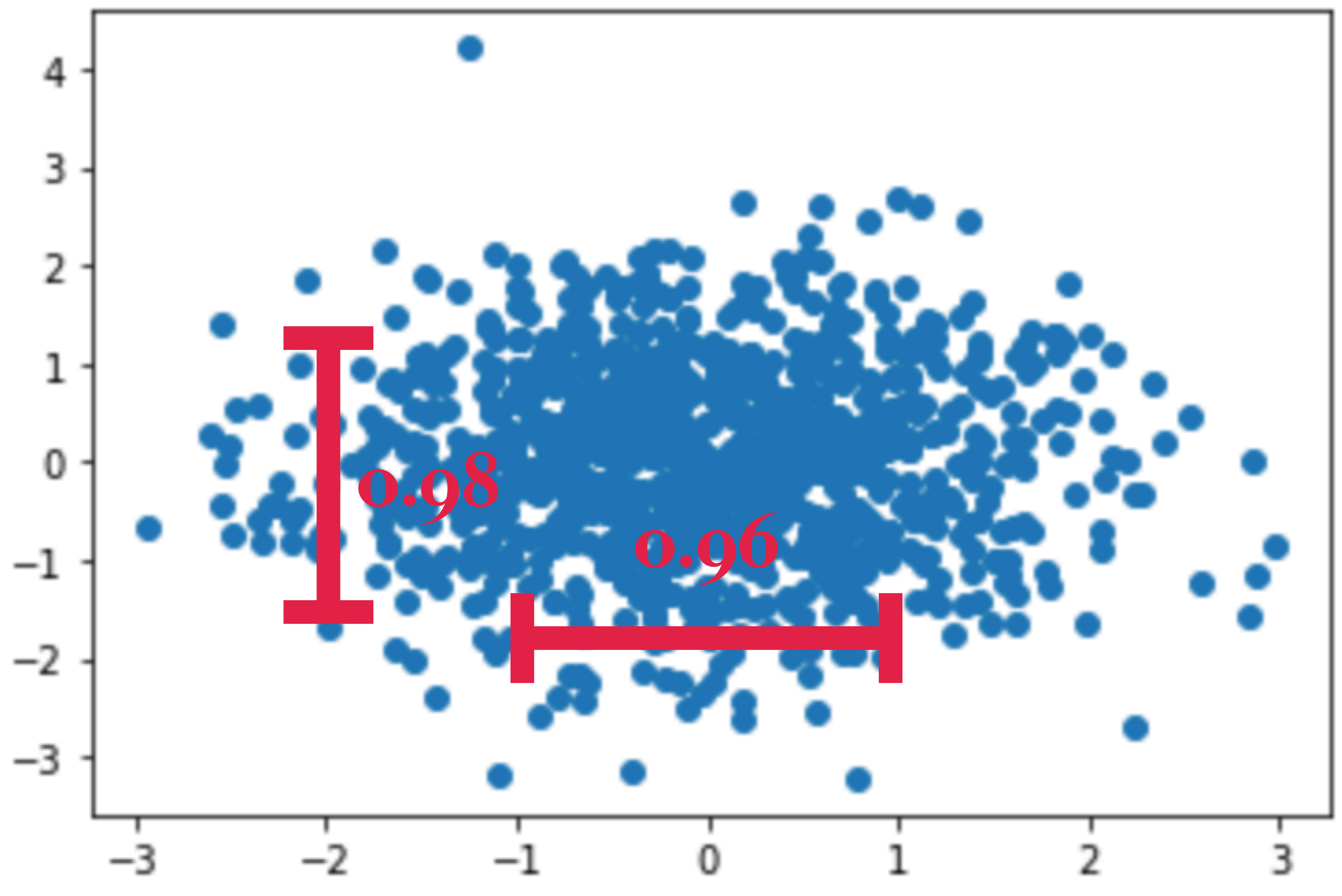
Two dimensions to two dimensions



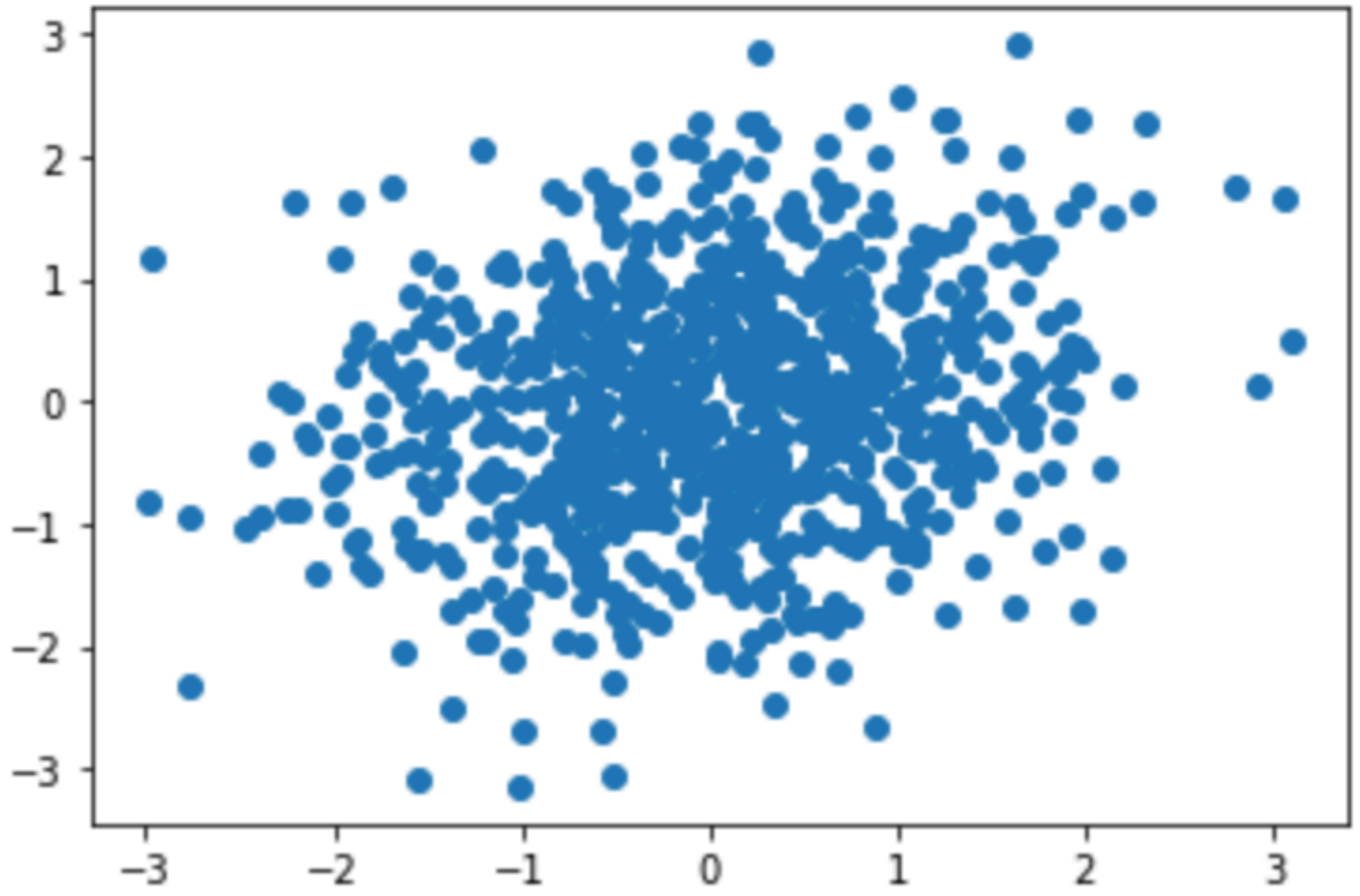

```
In [13]: np.var(X0,axis=0)
Out[13]: array([0.9569638 , 0.97715073])
```

Very simple case of PCA

Two dimensions to two dimensions

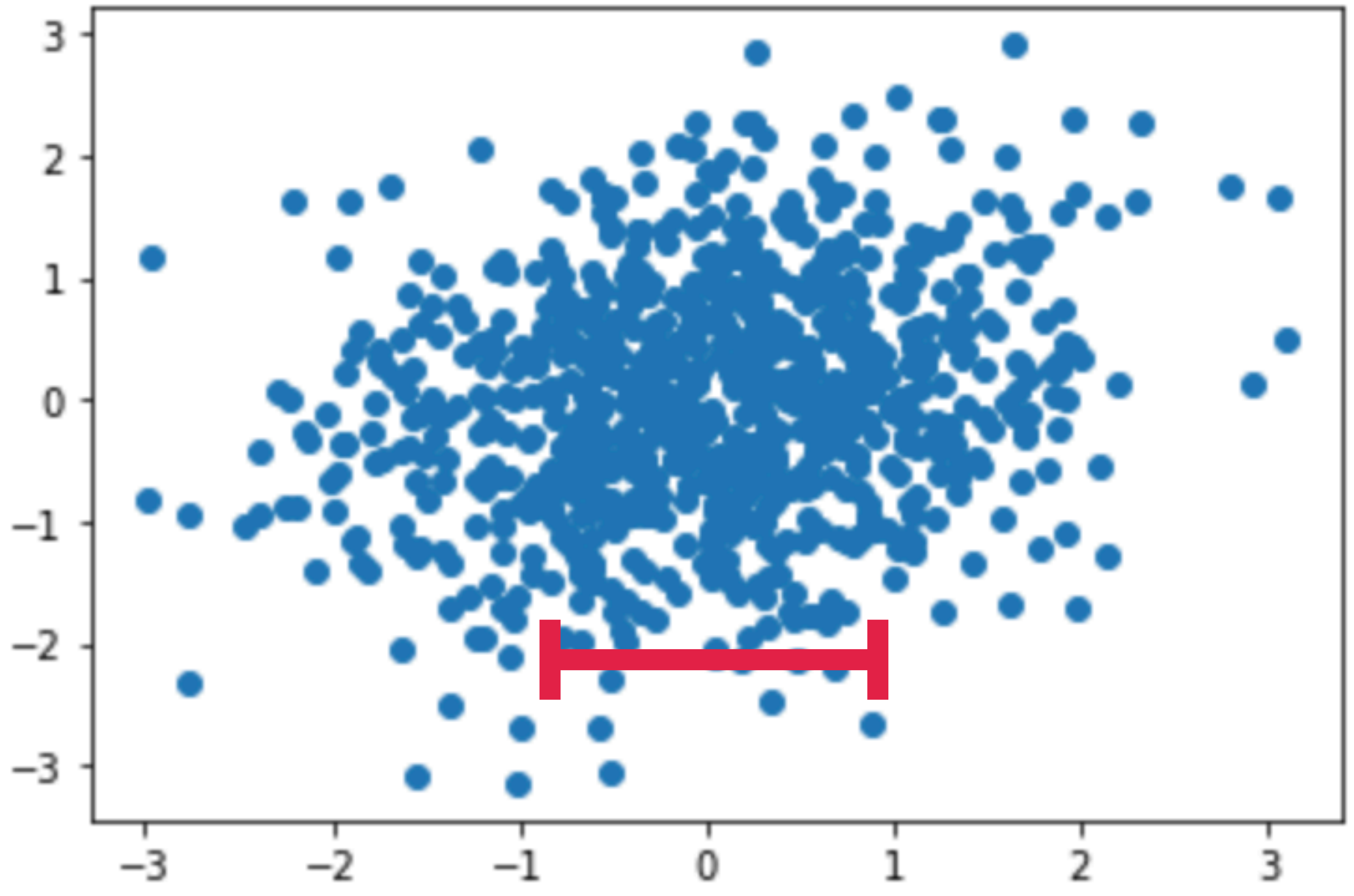



```
In [13]: np.var(X0,axis=0)
Out[13]: array([0.9569638 , 0.97715073])
```

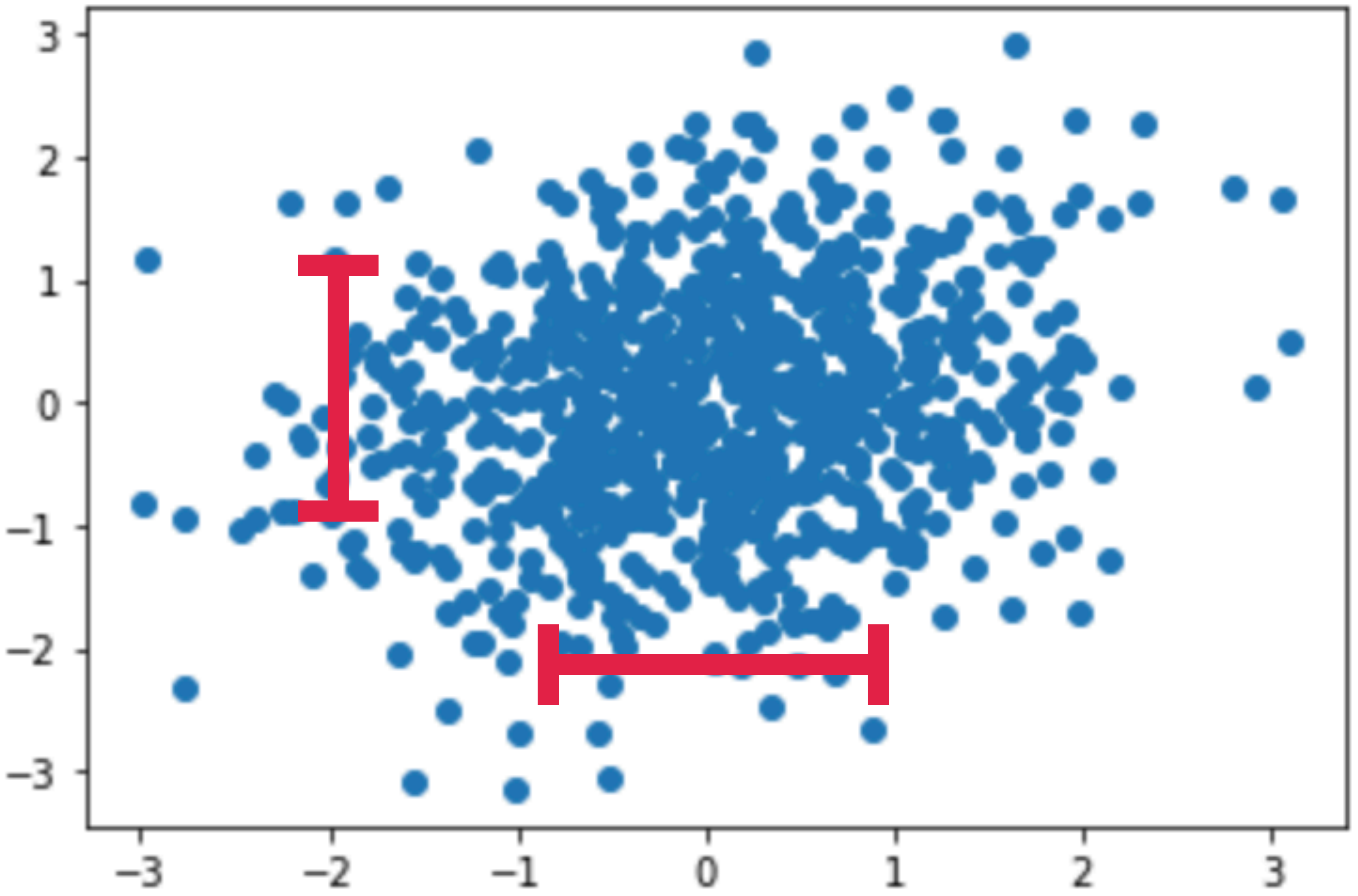


In [15]:
```
np.var(X1,axis=0)
```

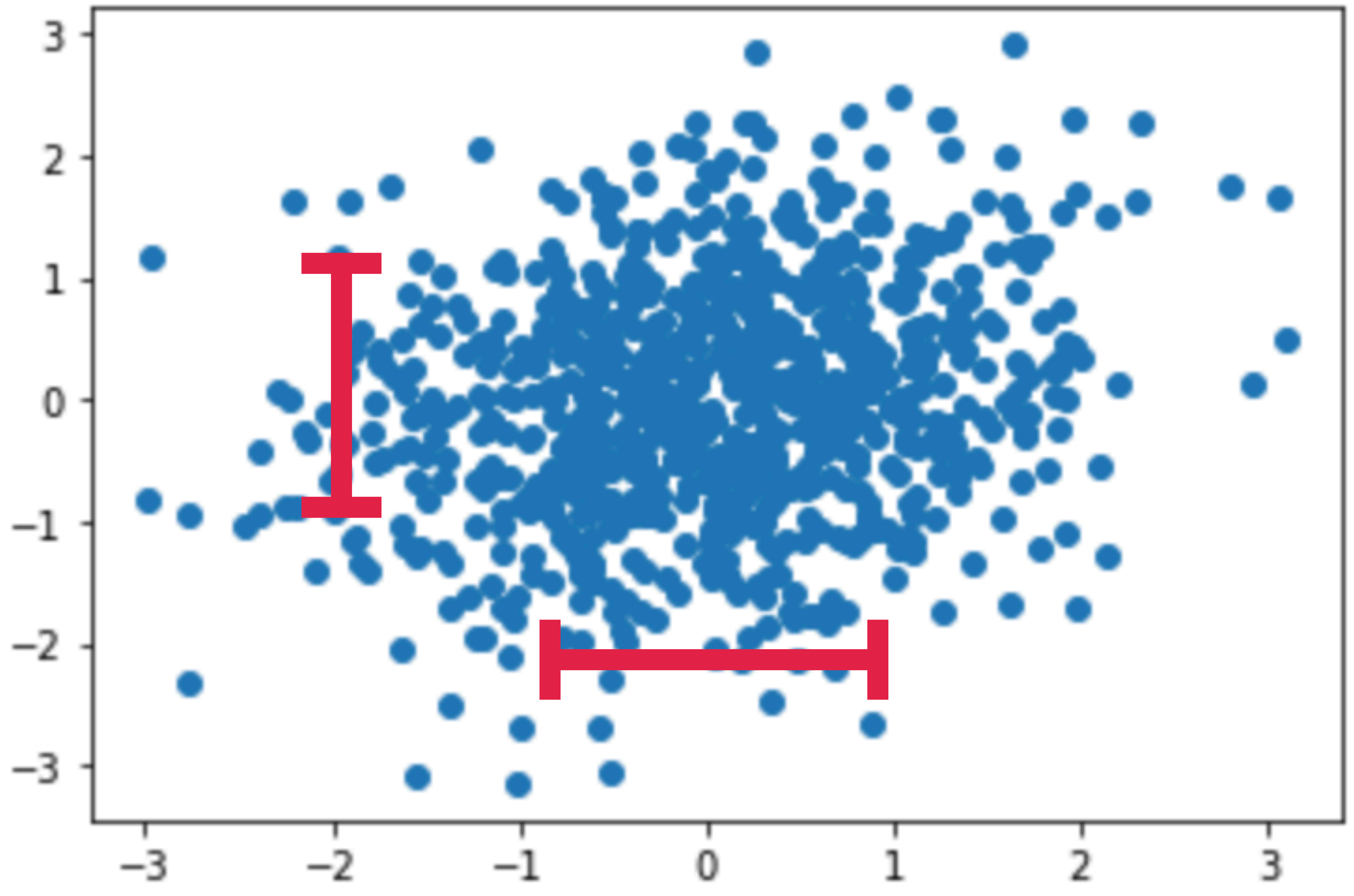
Out[15]:
```
array([0.9654444 , 0.98708471])
```




```
In [15]: np.var(X1,axis=0)
Out[15]: array([0.9654444 , 0.98708471])
```




```
In [15]: np.var(X1,axis=0)
Out[15]: array([0.9654444 , 0.98708471])
```

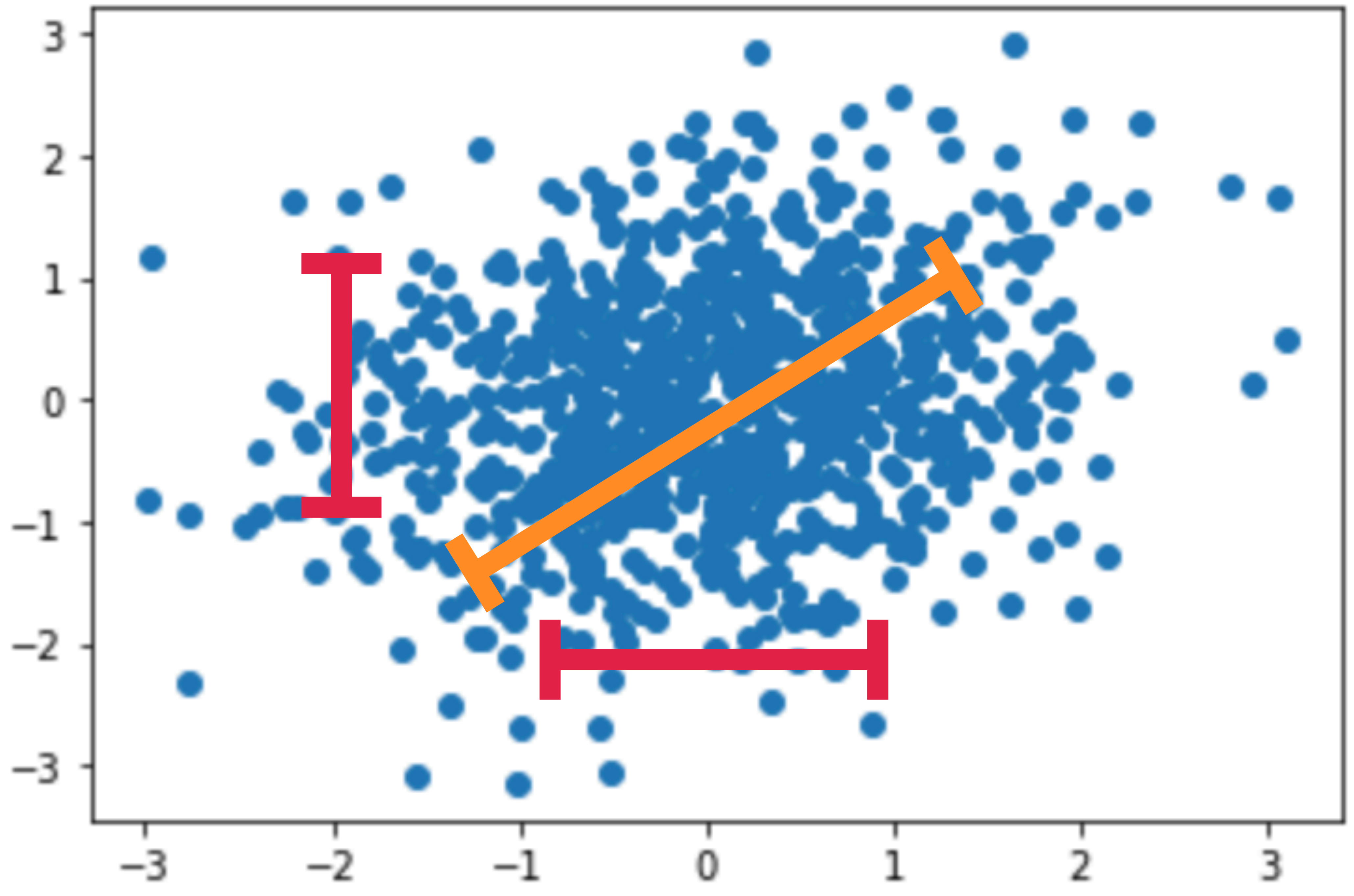


In [15]:
```
np.var(X1,axis=0)
```
Out[15]: `array([0.9654444 , 0.98708471])`

In [16]:
```
from sklearn.decomposition import PCA

model = PCA()
model.fit(X)
print(model.components_)
```
```
[[-0.56934021  0.82210201]
 [-0.82210201 -0.56934021]]
```

In [19]:
```
X1_transform = model.transform(X1)
np.var(X1_transform,axis=0)
```
Out[19]: `array([0.78612429, 1.16640482])`

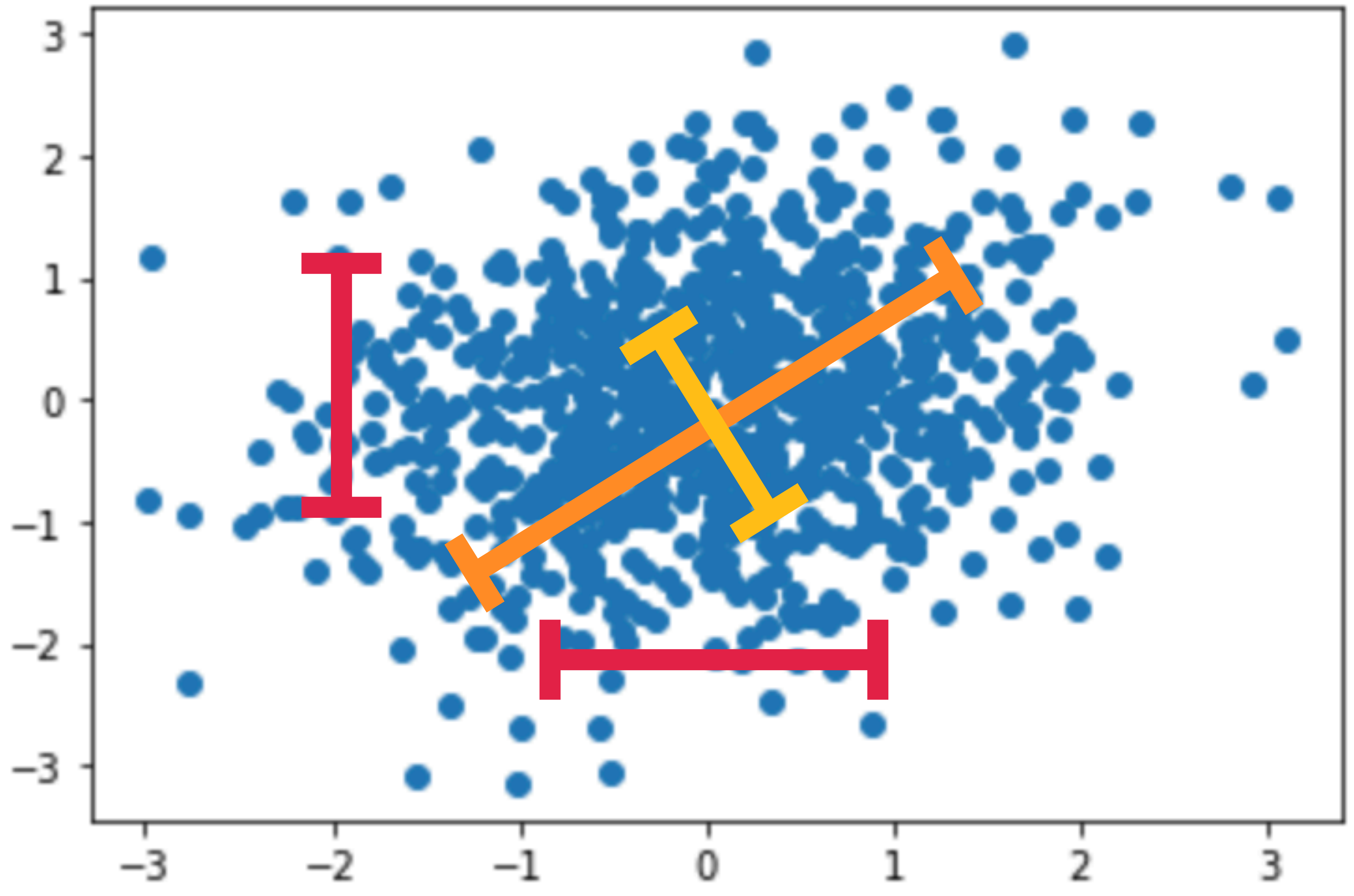


In [15]:
```
np.var(X1,axis=0)
```
Out[15]:
```
array([0.9654444 , 0.98708471])
```

In [16]:
```
from sklearn.decomposition import PCA

model = PCA()
model.fit(X)
print(model.components_)
```
```
[[-0.56934021  0.82210201]
 [-0.82210201 -0.56934021]]
```

In [19]:
```
X1_transform = model.transform(X1)
np.var(X1_transform,axis=0)
```
Out[19]:
```
array([0.78612429, 1.16640482])
```




```
In [15]: np.var(X1,axis=0)

Out[15]: array([0.9654444 , 0.98708471])
```

```
In [16]: from sklearn.decomposition import PCA

         model = PCA()
         model.fit(X)
         print(model.components_)

[[-0.56934021  0.82210201]
 [-0.82210201 -0.56934021]]
```

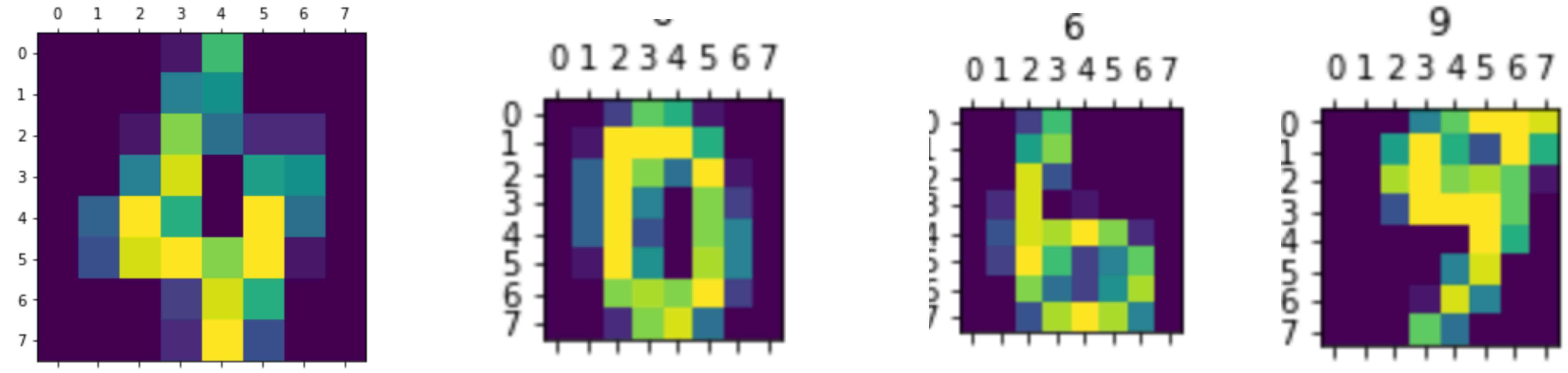
```
In [19]: X1_transform = model.transform(X1)
         np.var(X1_transform,axis=0)

Out[19]: array([0.78612429, 1.16640482])
```

Example: MNIST dataset

Each datum has 64 pixels (features)

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel55	pixel56	pixel57	pixel58	pixel59	pixel60	pixel61	pixel62	pixel63	label
	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0	0.0	0
	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0	0.0	4
	0.0	0.0	0.0	12.0	13.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	9.0	15.0	11.0	3.0	0.0	6
	0.0	0.0	11.0	12.0	0.0	0.0	0.0	0.0	0.0	2.0	...	0.0	0.0	0.0	9.0	12.0	13.0	3.0	0.0	0.0	9
	0.0	0.0	1.0	9.0	15.0	11.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	10.0	13.0	3.0	0.0	0.0	0



Example: MNIST dataset

Each datum has 64 pixels (features)

pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel55	pixel56	pixel57	pixel58	pixel59	pixel60	pixel61	pixel62	pixel63	label
0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0	0.0	0
0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0	0.0	4
0.0	0.0	0.0	12.0	13.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	9.0	15.0	11.0	3.0	0.0	6
0.0	0.0	11.0	12.0	0.0	0.0	0.0	0.0	0.0	2.0	...	0.0	0.0	0.0	9.0	12.0	13.0	3.0	0.0	0.0	9
0.0	0.0	1.0	9.0	15.0	11.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	1.0	10.0	13.0	3.0	0.0	0.0	0