



UCL ARTIFICIAL
INTELLIGENCE SOCIETY

TUTORIAL #4

Session 4

Introduction to unsupervised models and data pre-processing

SUPERVISED V. UNSUPERVISED LEARNING

Supervised L.

- It learns using data X and labels Y
- Learns $X \rightarrow Y$



Unspervised L.

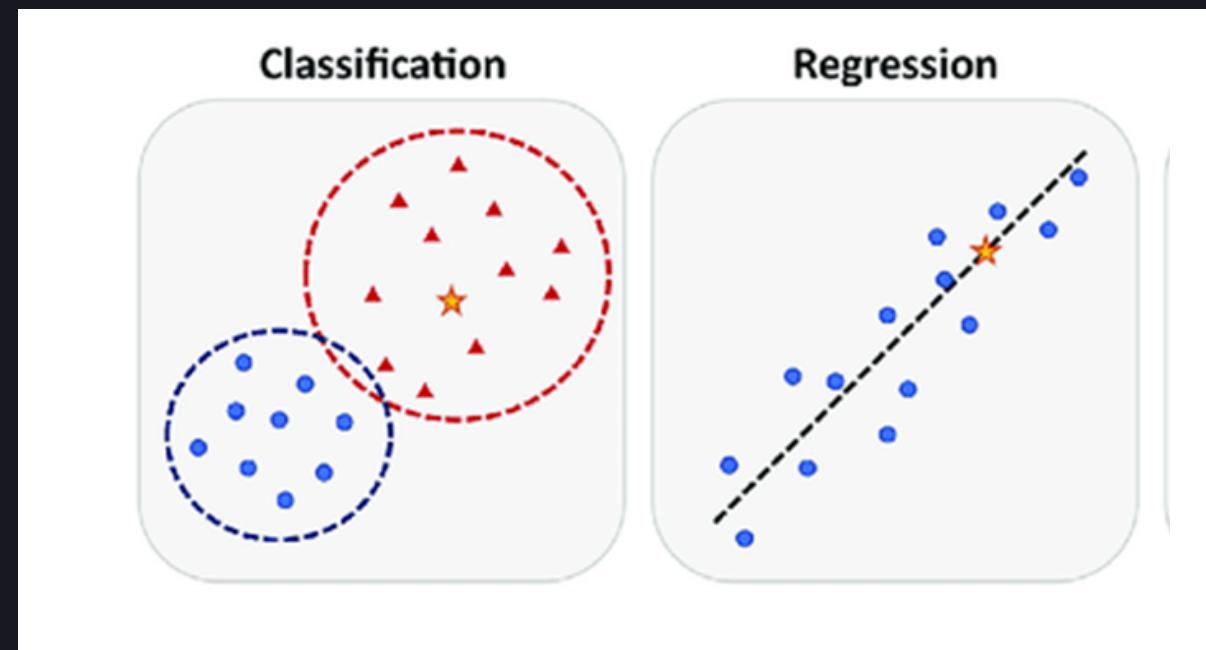
- It learns using just data X
- Observes underlying structures



SUPERVISED V. UNSUPERVISED LEARNING

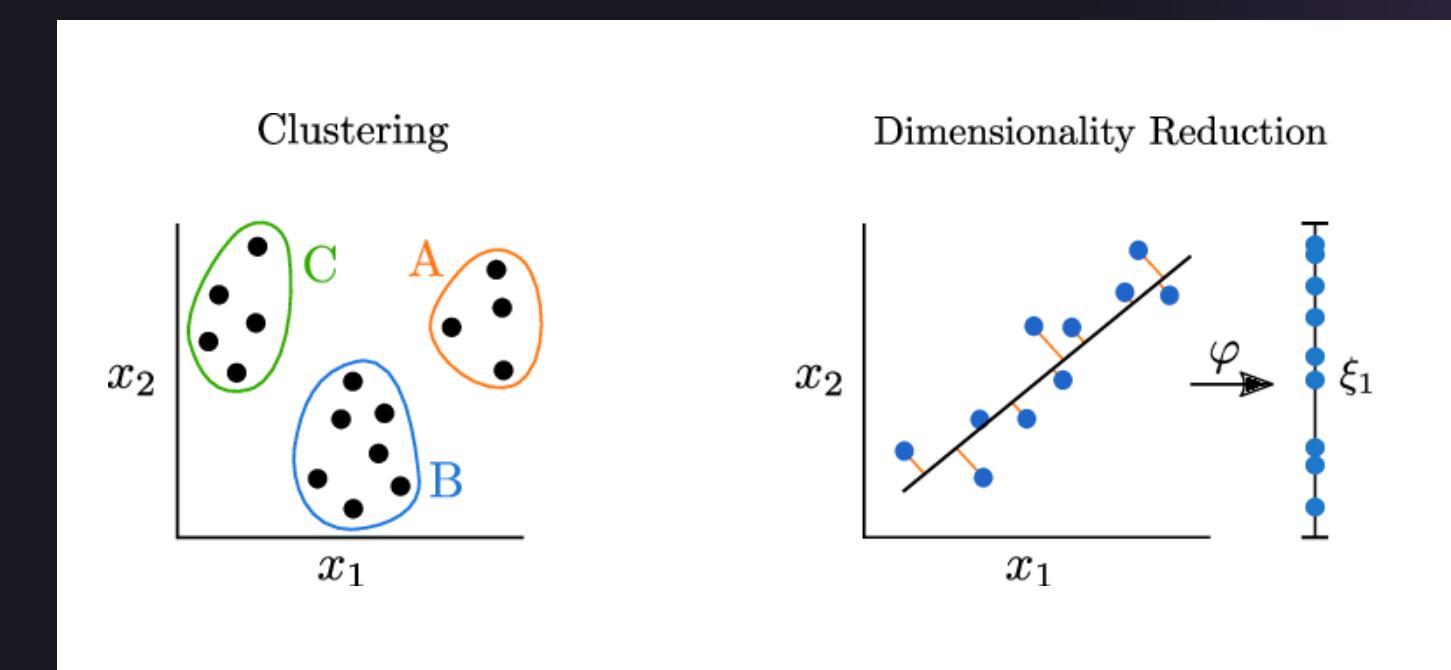
Supervised L.

- Fits some function
- Regression and Classification



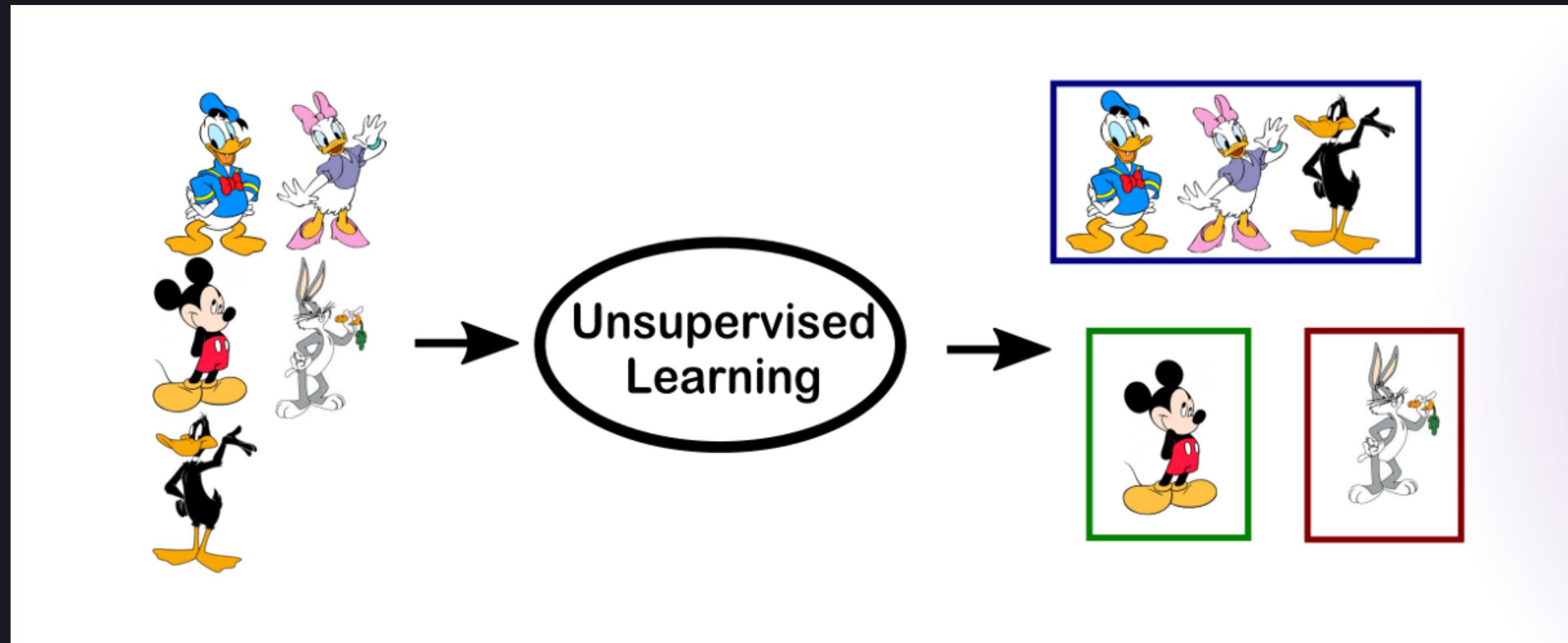
Unspervised L.

- Works with and transforms data
- Clustering and Dimensional Reduction



Theory

SUPERVISED V. UNSUPERVISED LEARNING



SUPERVISED V. UNSUPERVISED LEARNING

	<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>	classification or categorization	clustering
<i>Continuous</i>	regression	dimensionality reduction

Theory

DATA PRE-PROCESSING

Raw Data:

Imagination:

- No inconsistencies
- No noise
- Complete
- No missing values

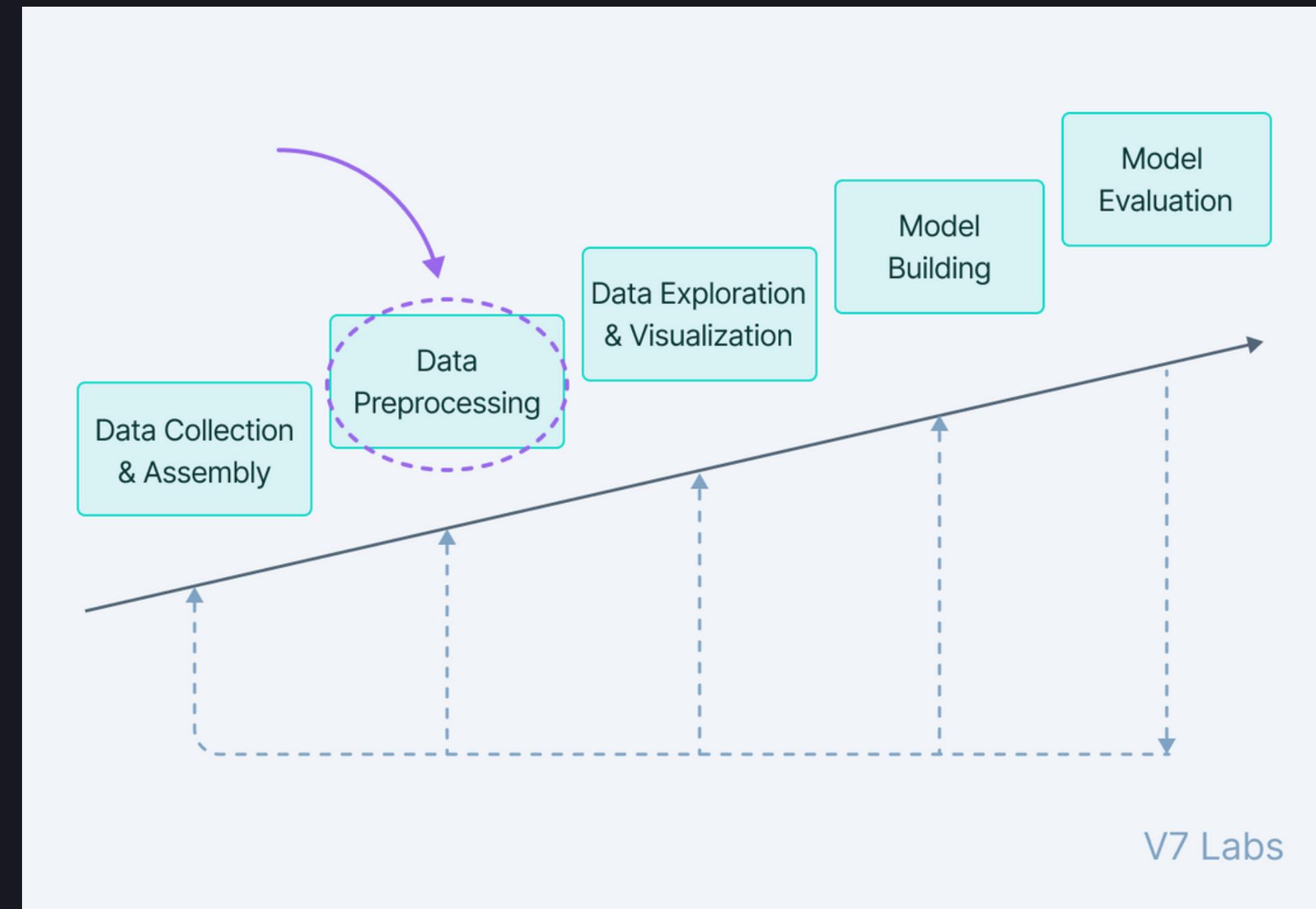
etc

Reality:



Theory

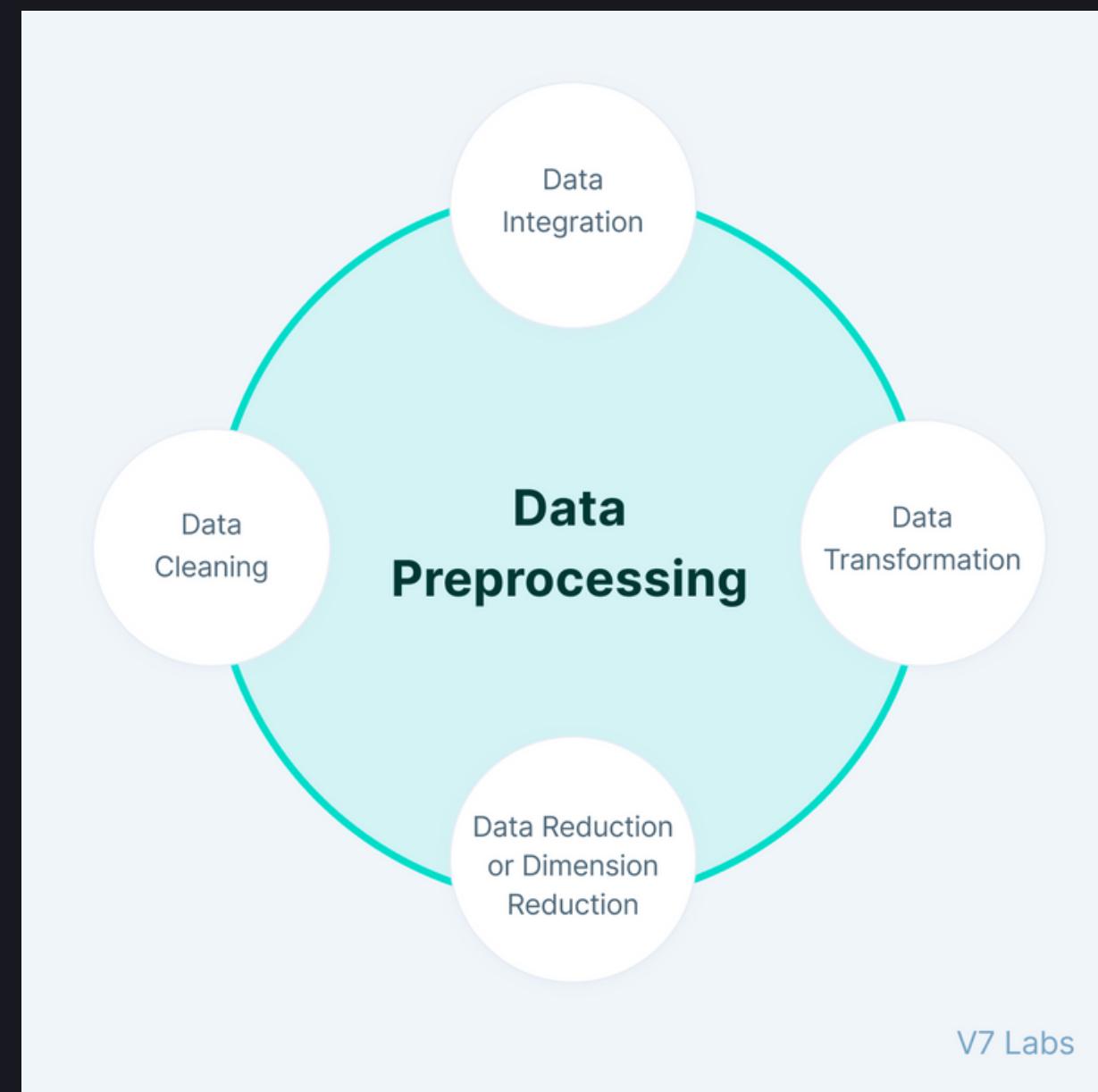
DATA PRE-PROCESSING



Theory

DATA PRE-PROCESSING

4 Steps in Data Preprocessing



Theory

DATA PRE-PROCESSING

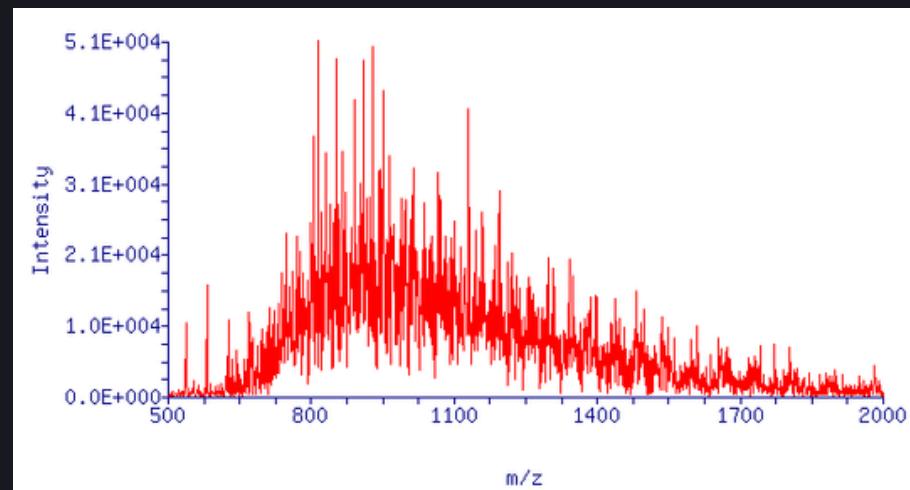
Data cleaning

1) Missing values

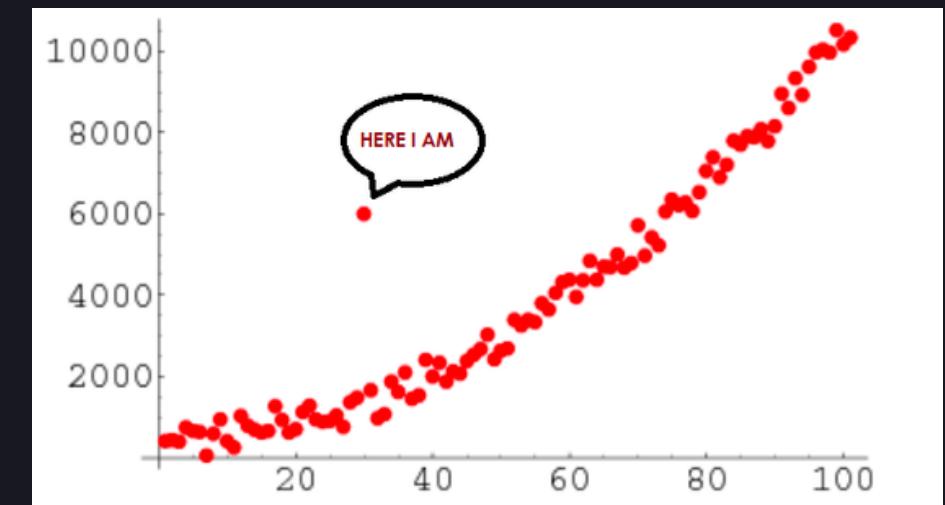
Missing values

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2033	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male	0	0	0	330877	8.4583		Q

2) Noisy data



3) Removing Outliers



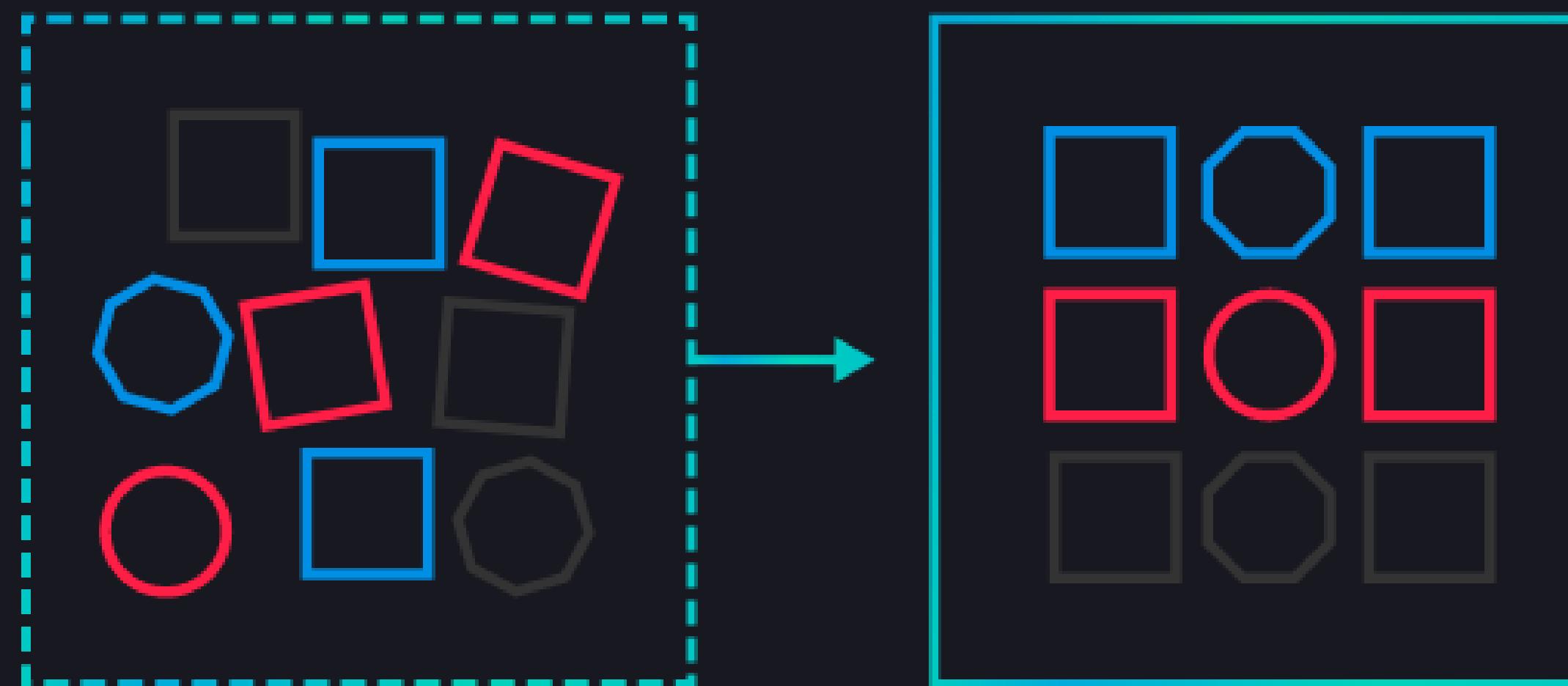
Theory

DATA INTEGRATION



Theory

DATA TRANSFORMATION



- Generalisation
- Normalisation
- Attribute Selection
- Aggregation

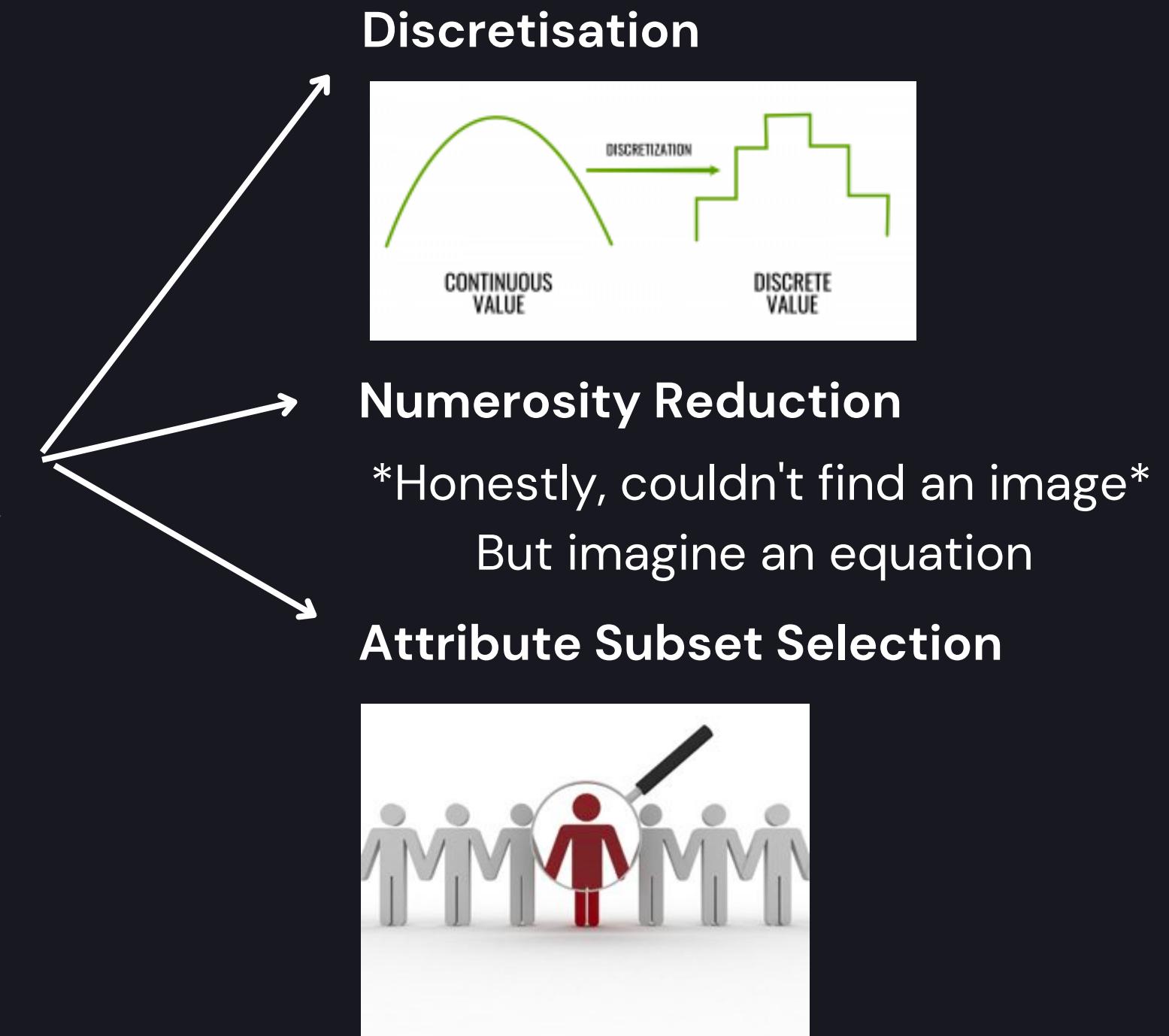
Basically, just making the data
actually usable

Theory

DATA REDUCTION



How? Lot's of ways.
Perhaps even too many.
But here's a couple:



Theory

VISUALISATION

Why would we need to visualise data?

VISUALISATION

Why would we need to visualise data?

01

Identify
patterns

03

Feature selection

02

Identify corrupt
data/outliers

04

Understanding high
dimensional data

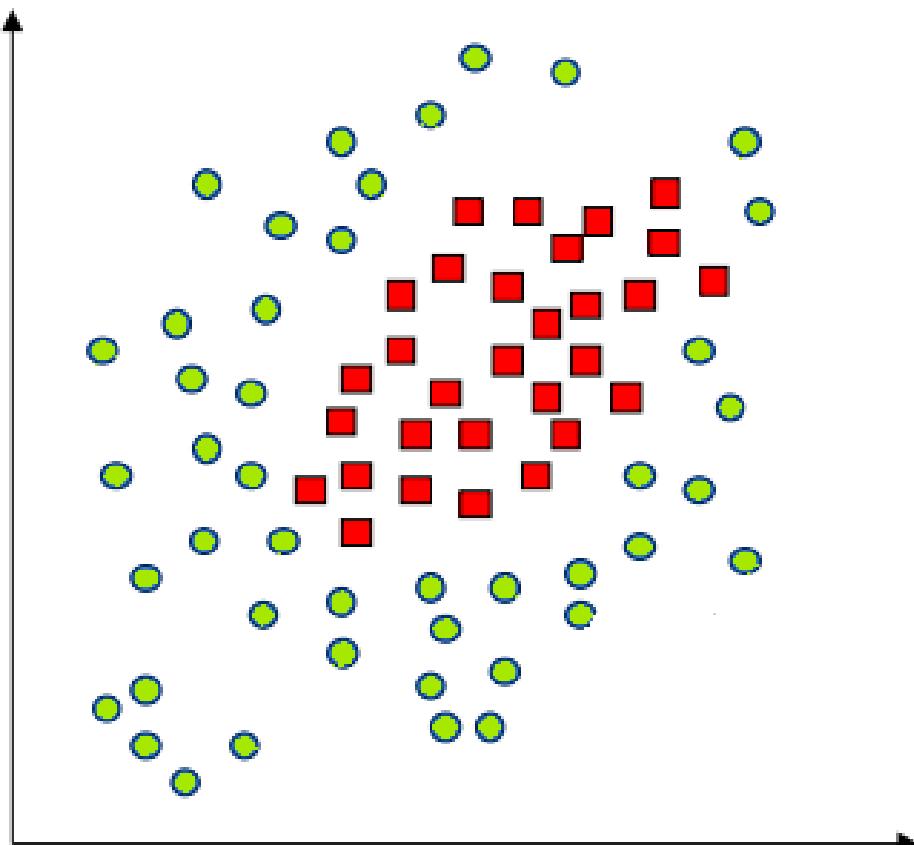
Theory

VISUALISATION

Why would we need to visualise data?

01

Identify patterns



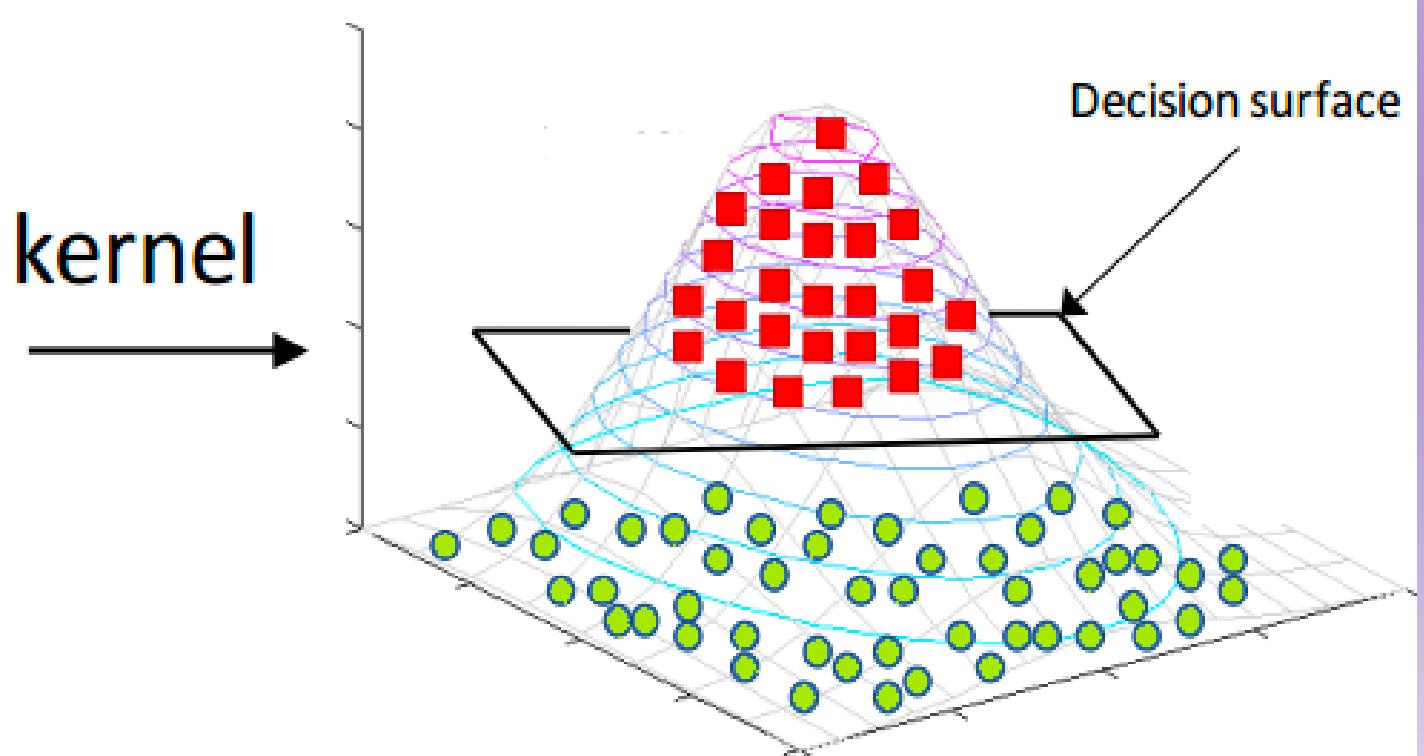
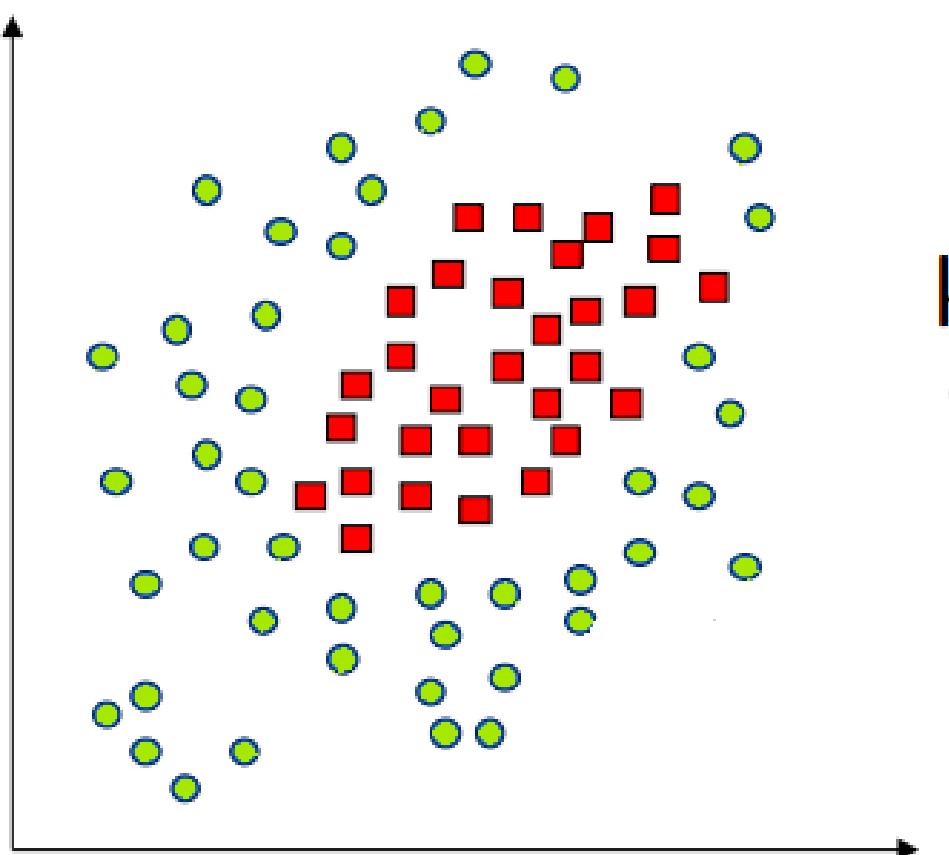
Theory

VISUALISATION

Why would we need to visualise data?

O1

Identify patterns



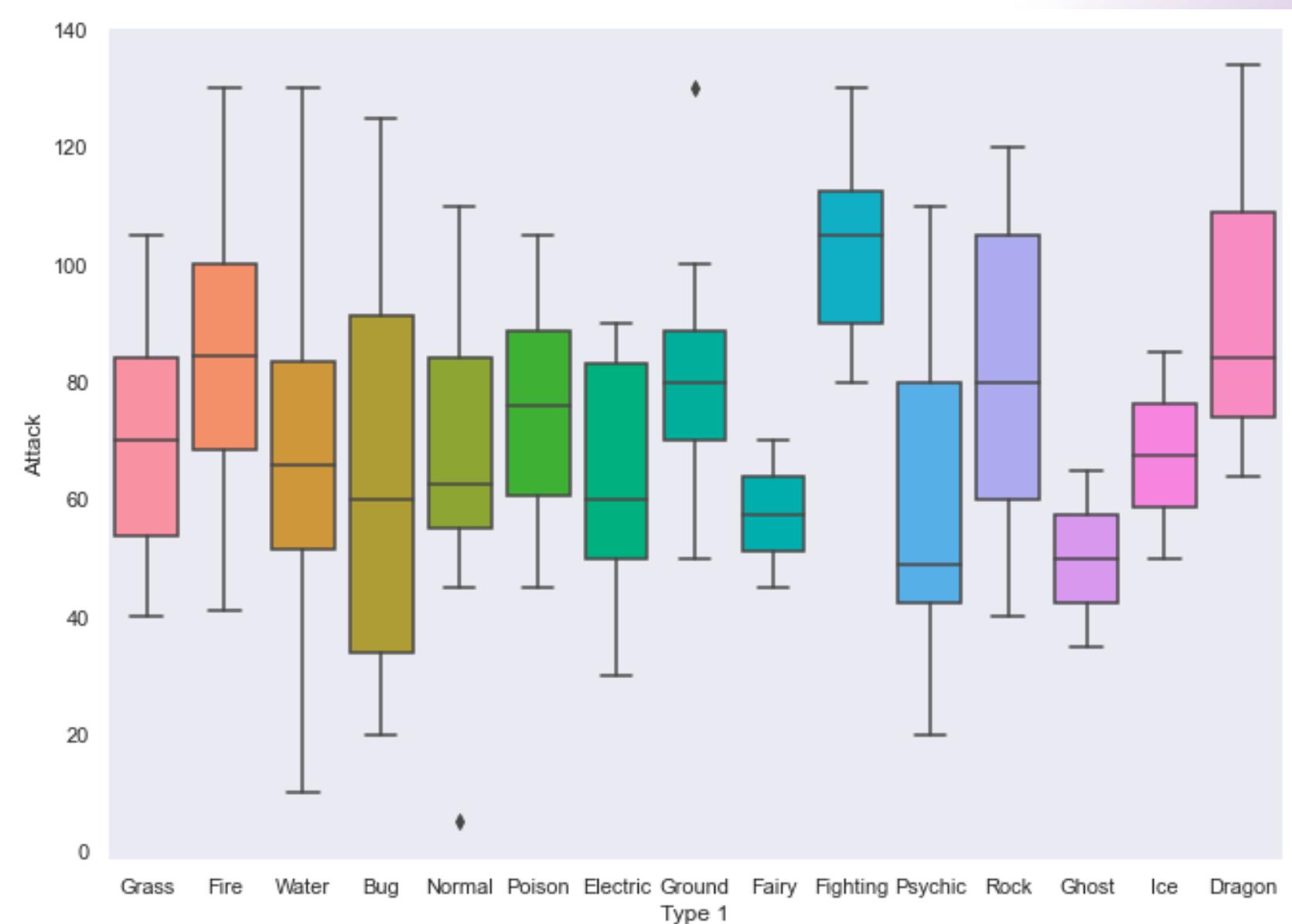
Theory

VISUALISATION

Why would we need to visualise data?

02

Identify outliers



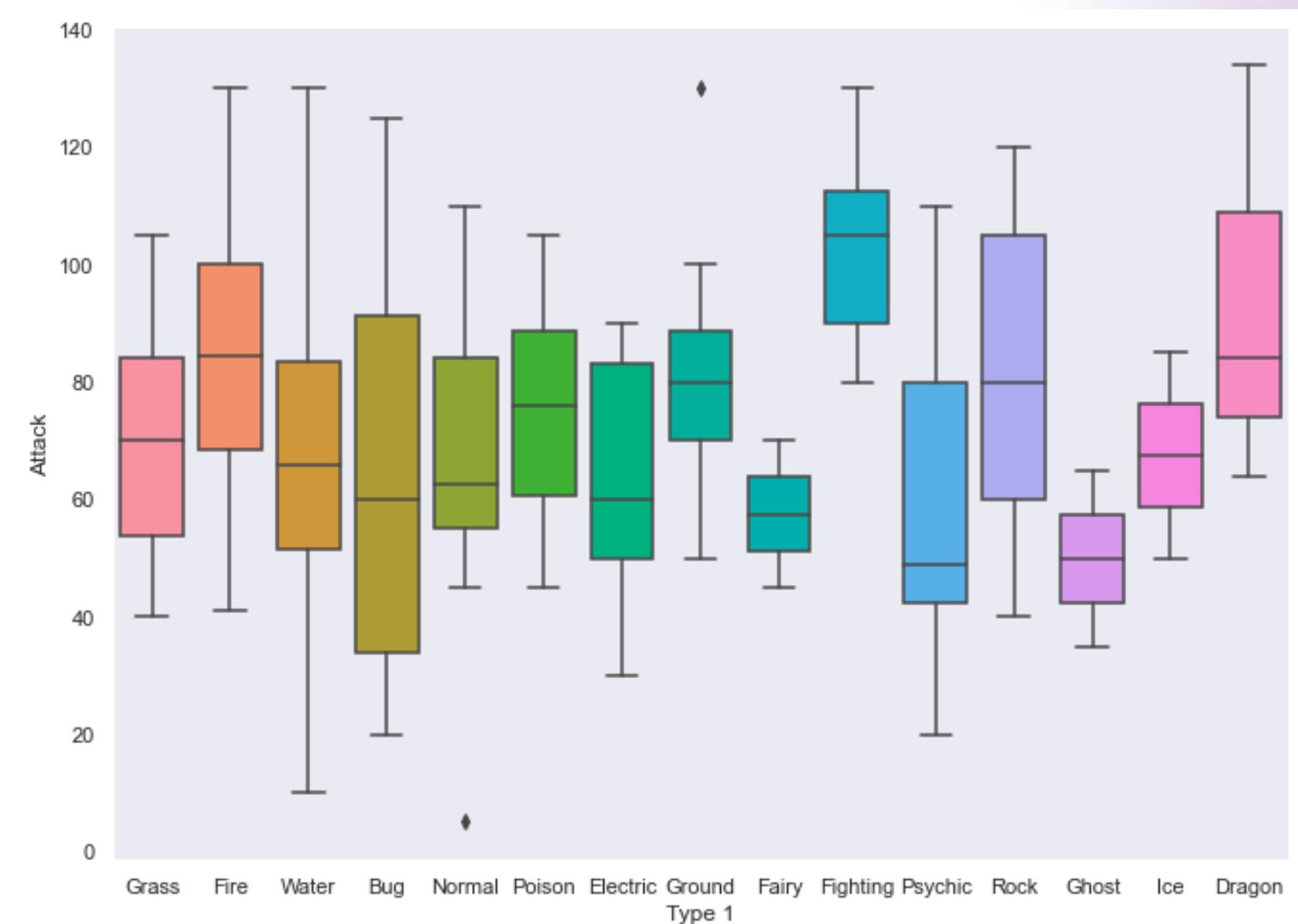
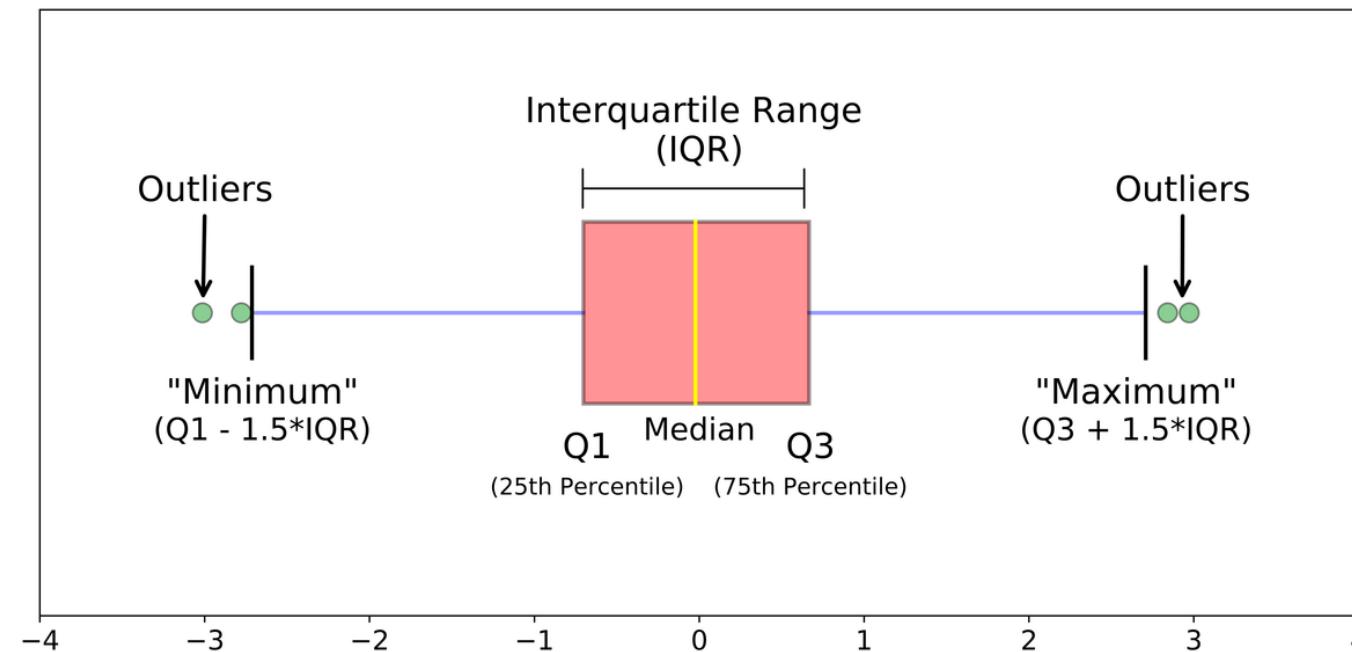
Theory

VISUALISATION

Why would we need to visualise data?

02

Identify outliers



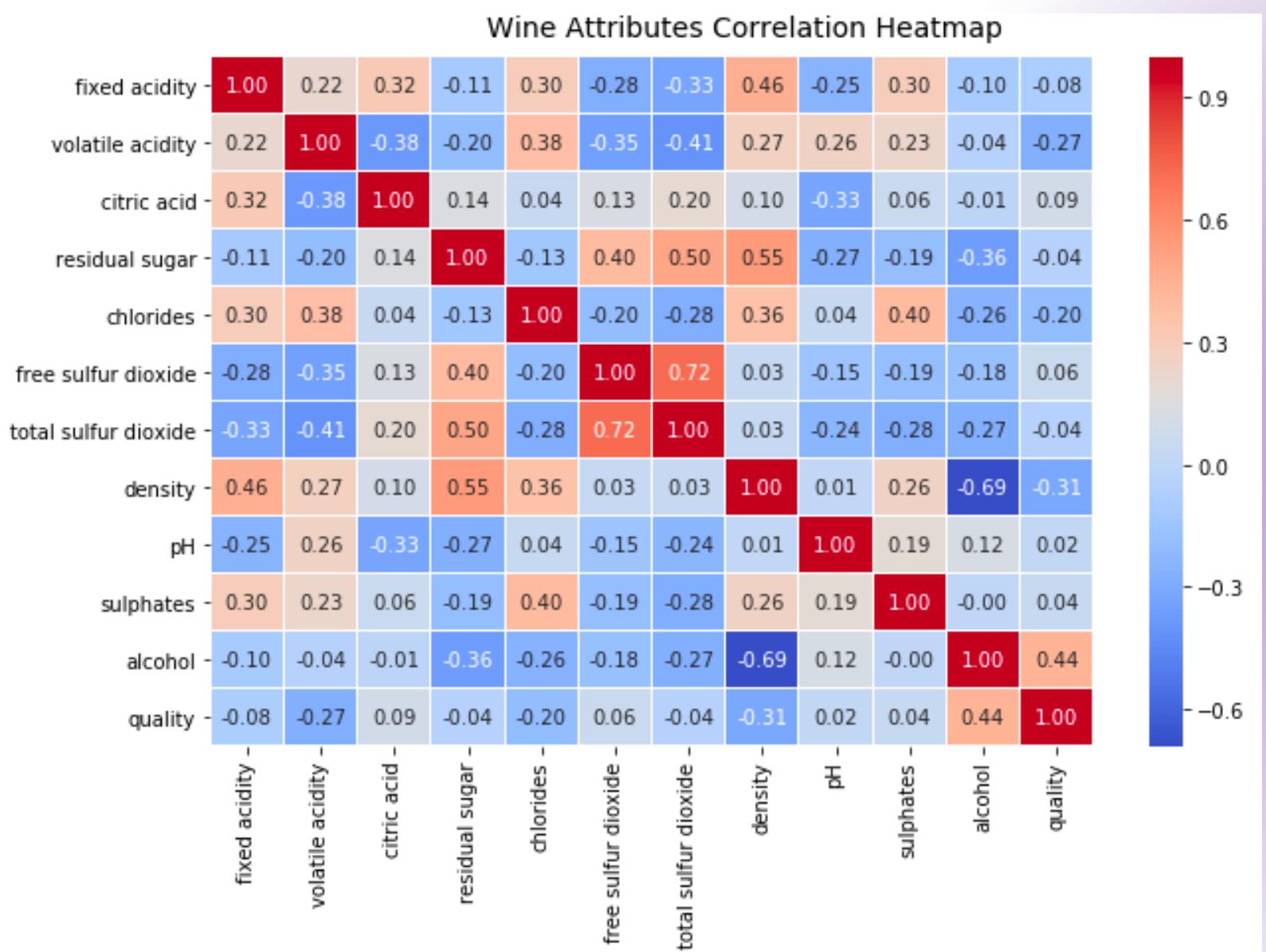
Theory

VISUALISATION

Why would we need to visualise data?

03

Feature selection



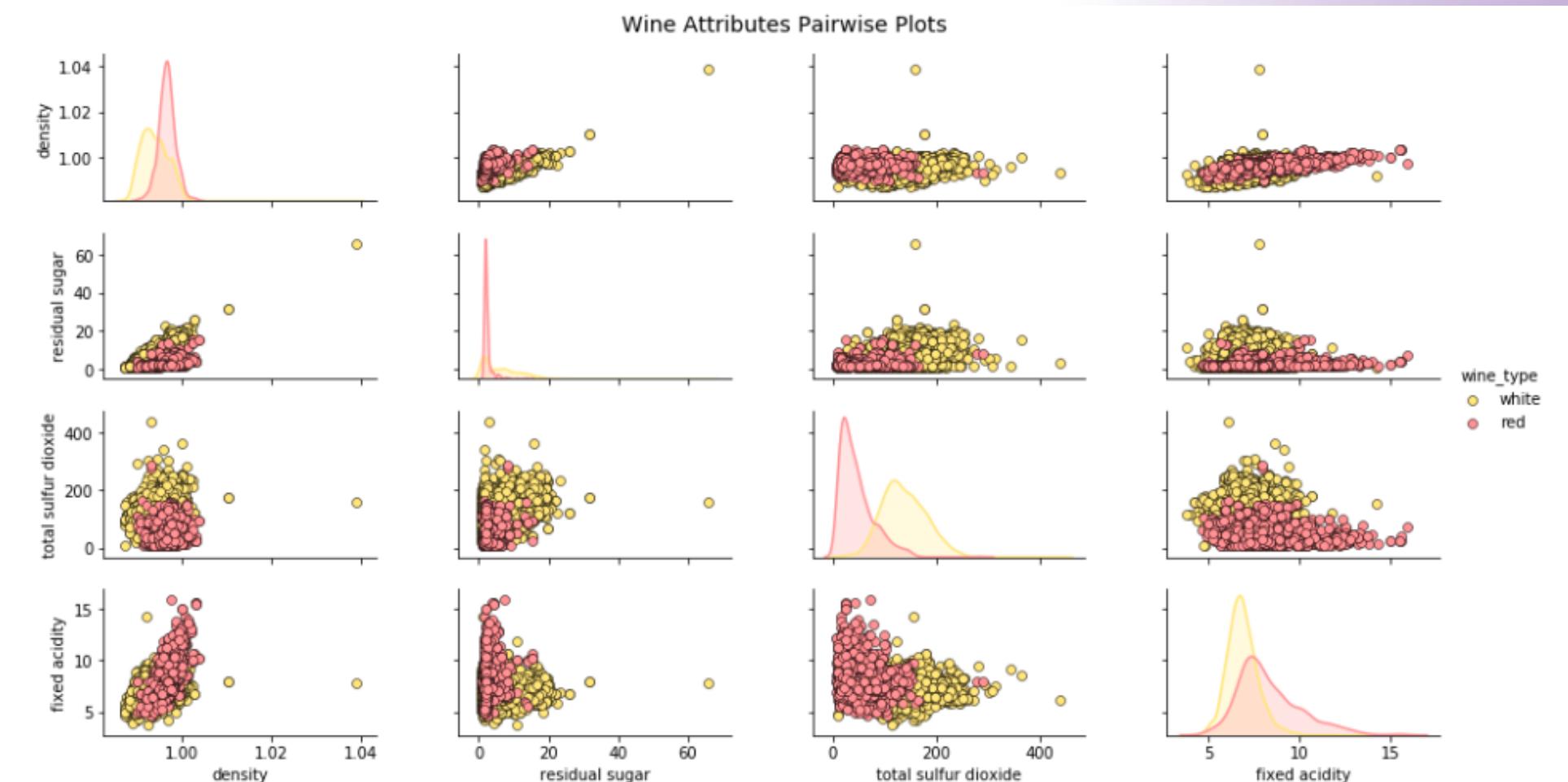
Theory

VISUALISATION

Why would we need to visualise data?

04

Understanding high dimensional data



Theory

PCA

Dimensionality reduction

- Curse of dimensionality
 - Sparse and dissimilar data

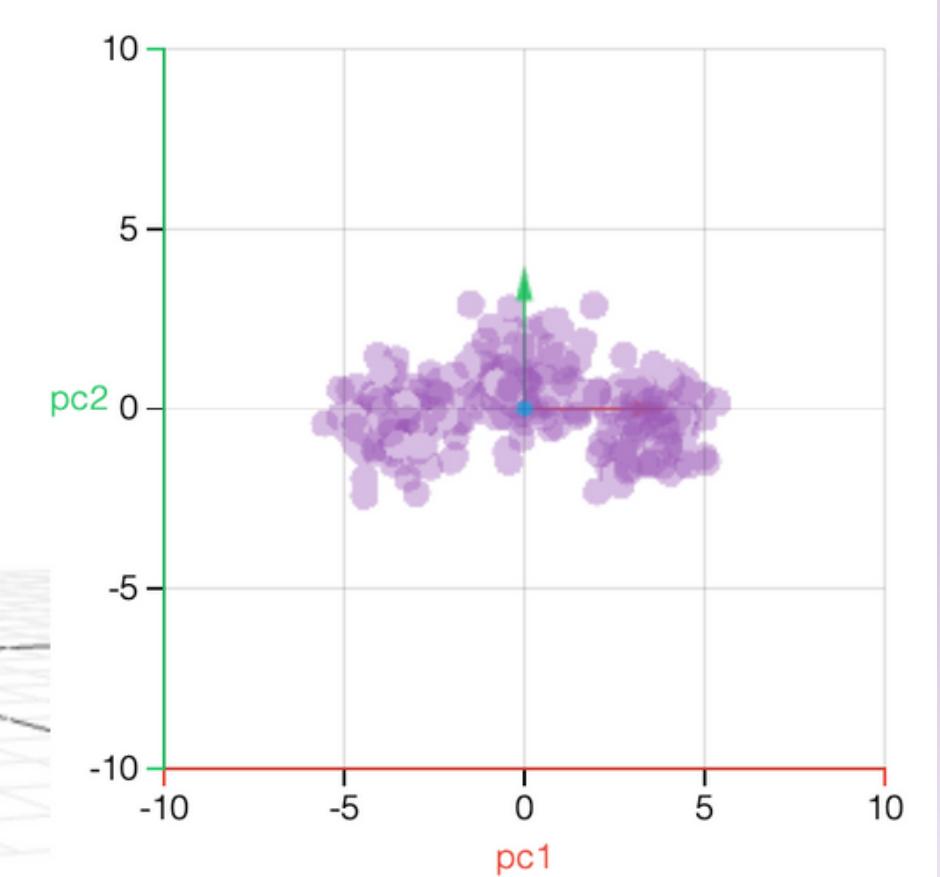
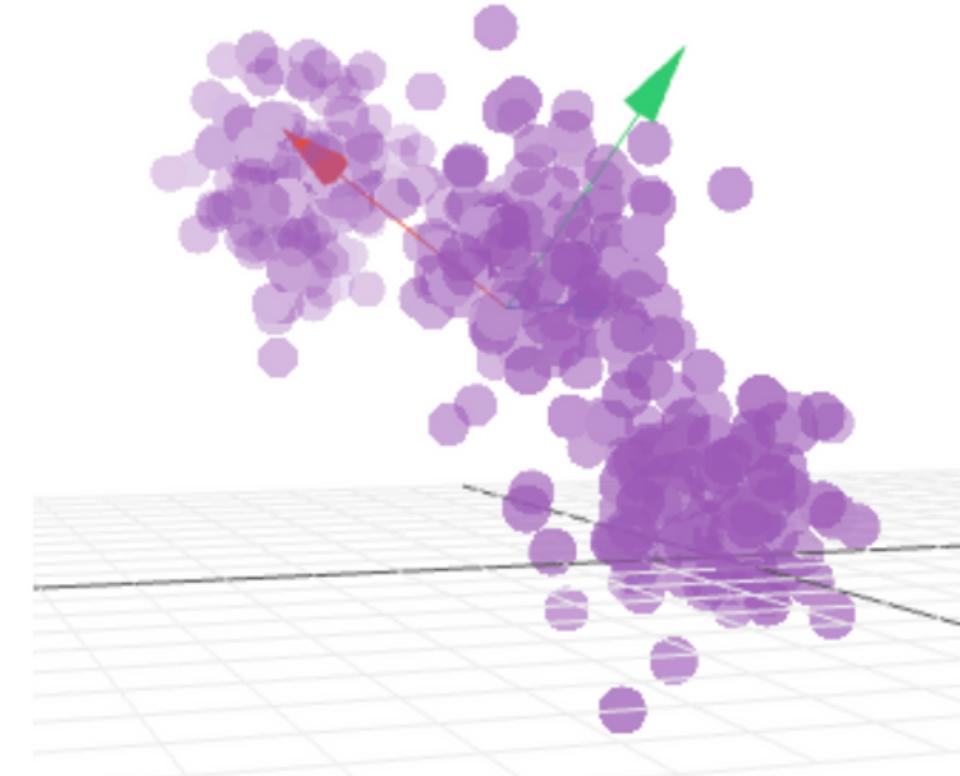
2D

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

3D

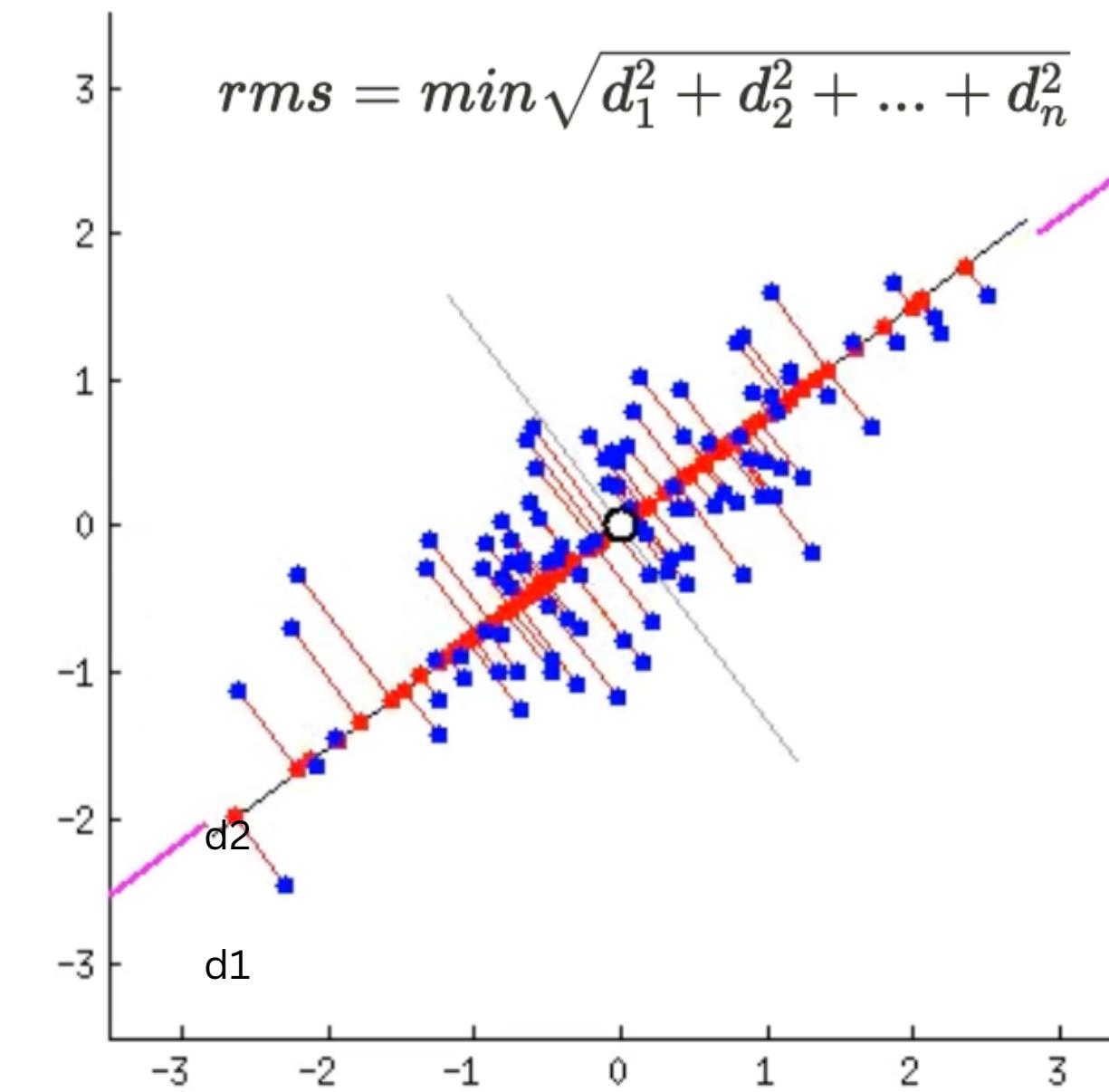
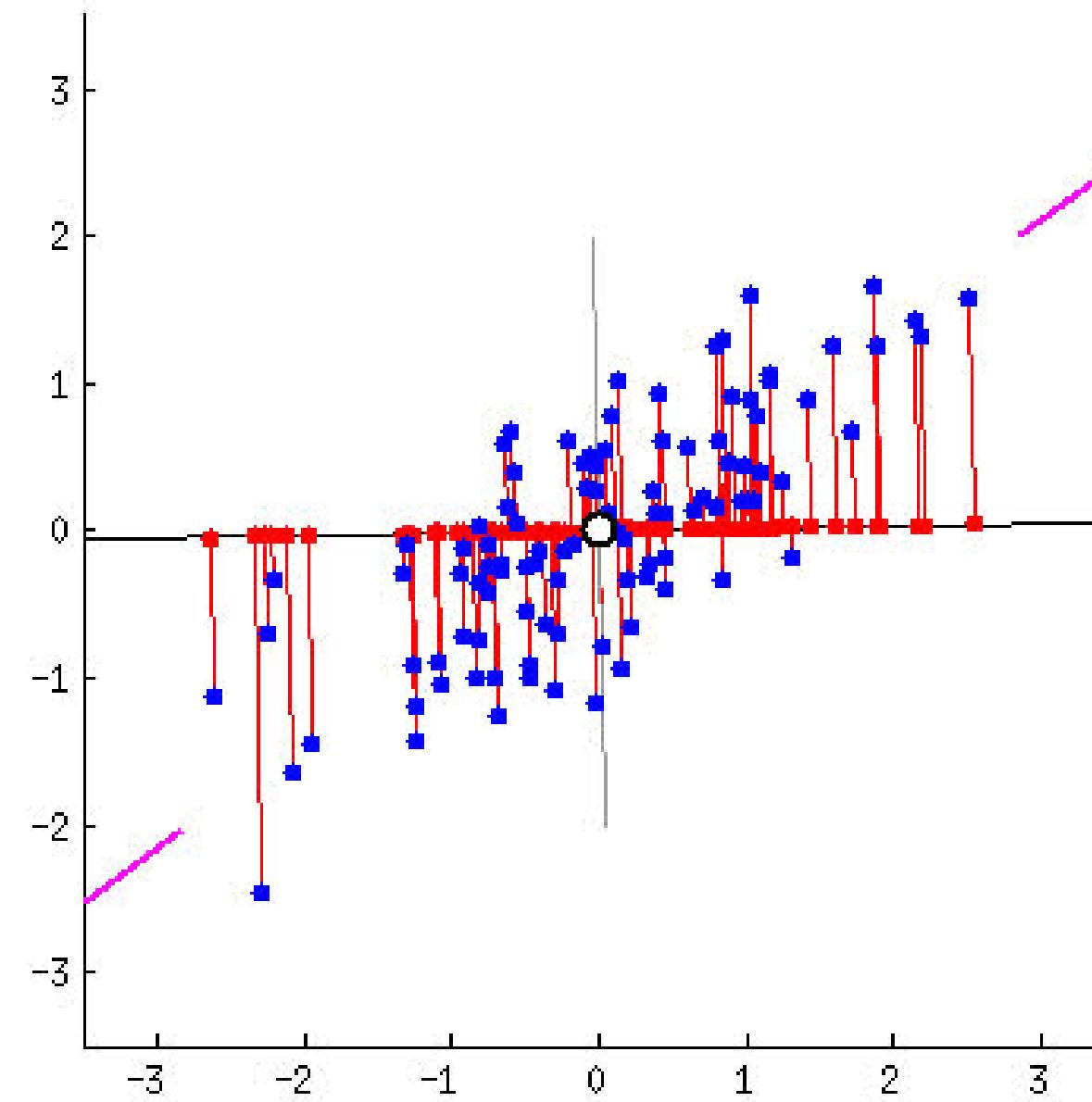
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

- Reduces multicollinearity



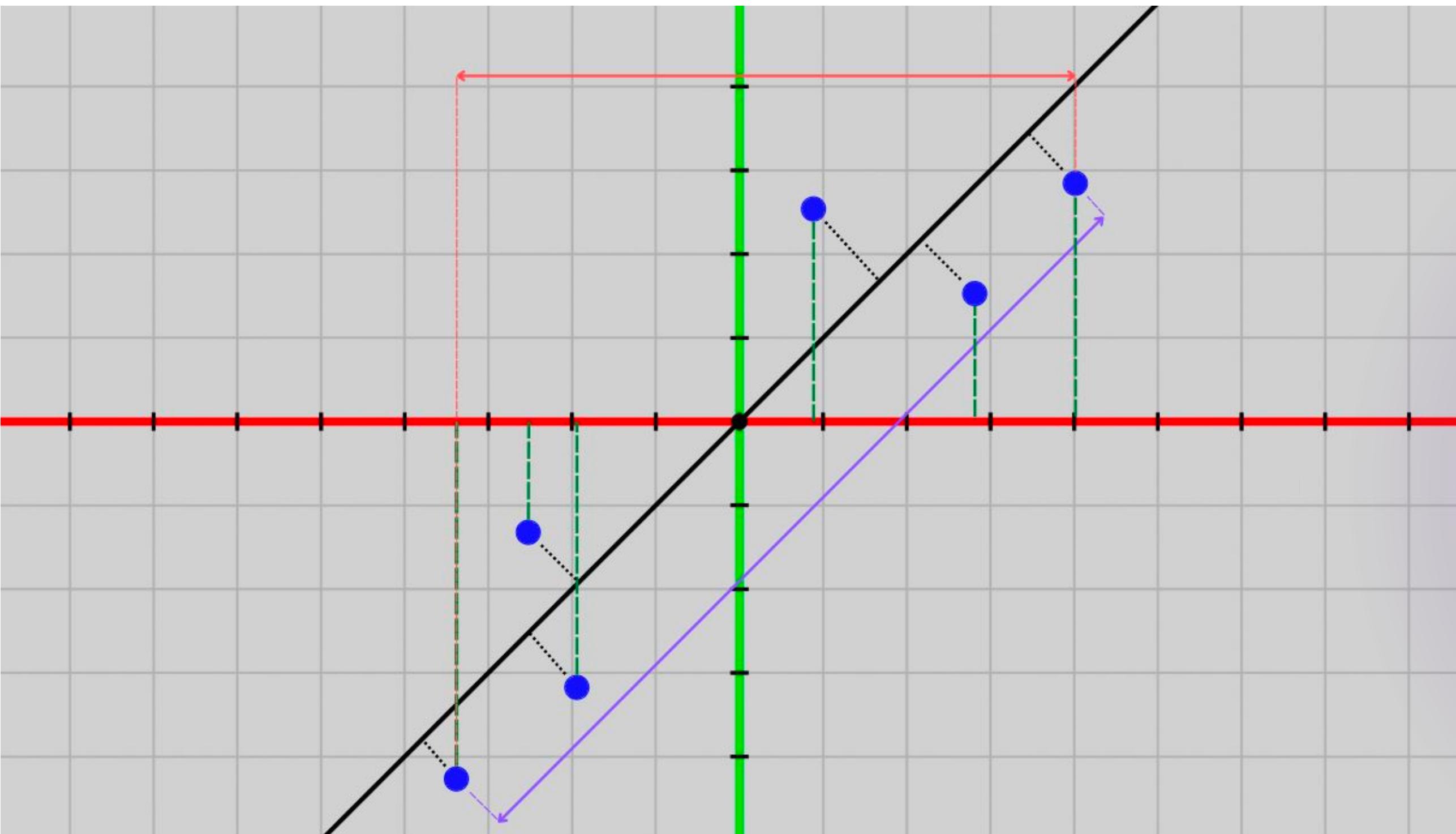
Theory

PCA



Theory

PCA



Theory

PCA

Reducing
dimensions

Choose n
best PCs

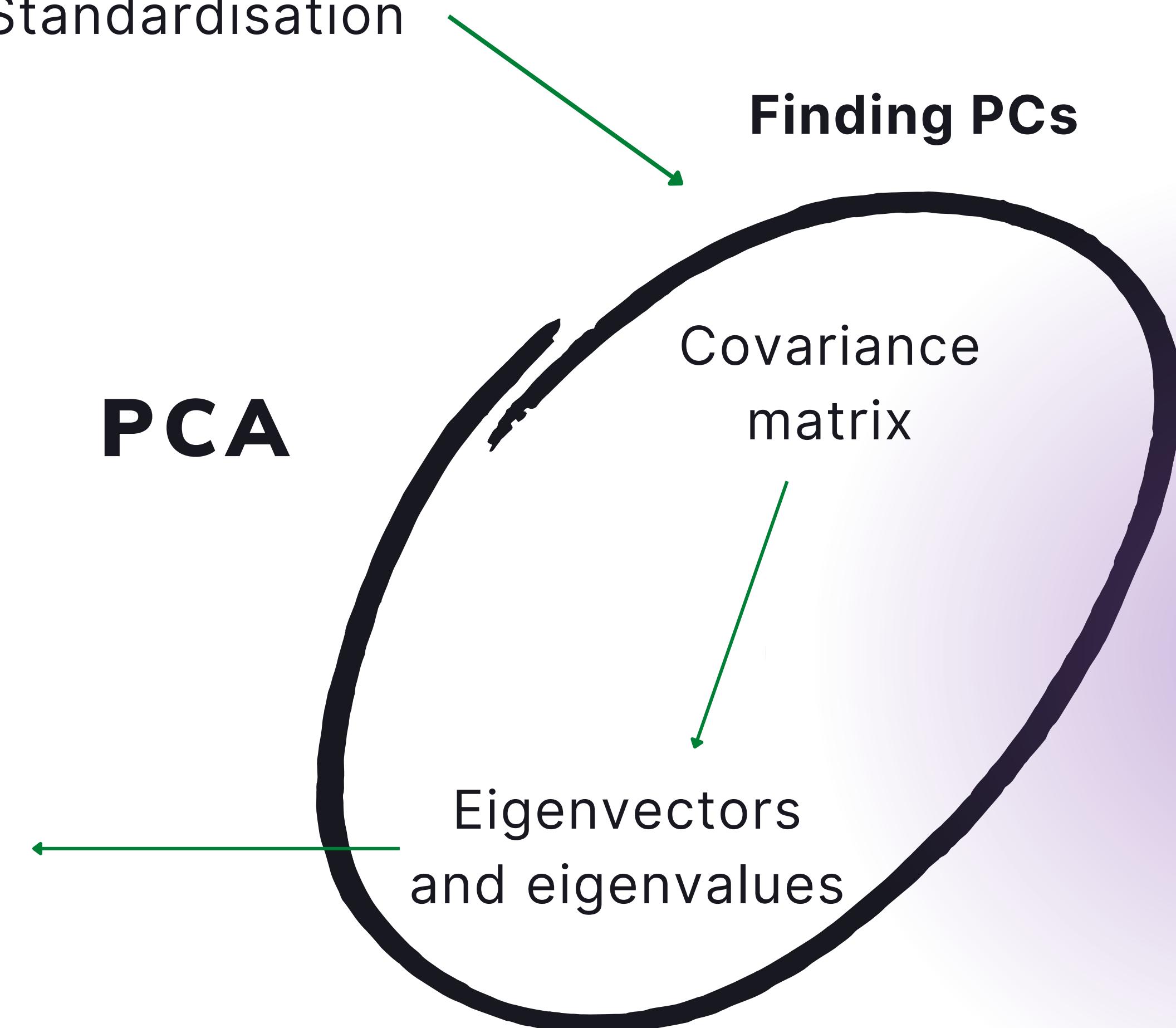
Standardisation

PCA

Finding PCs

Covariance
matrix

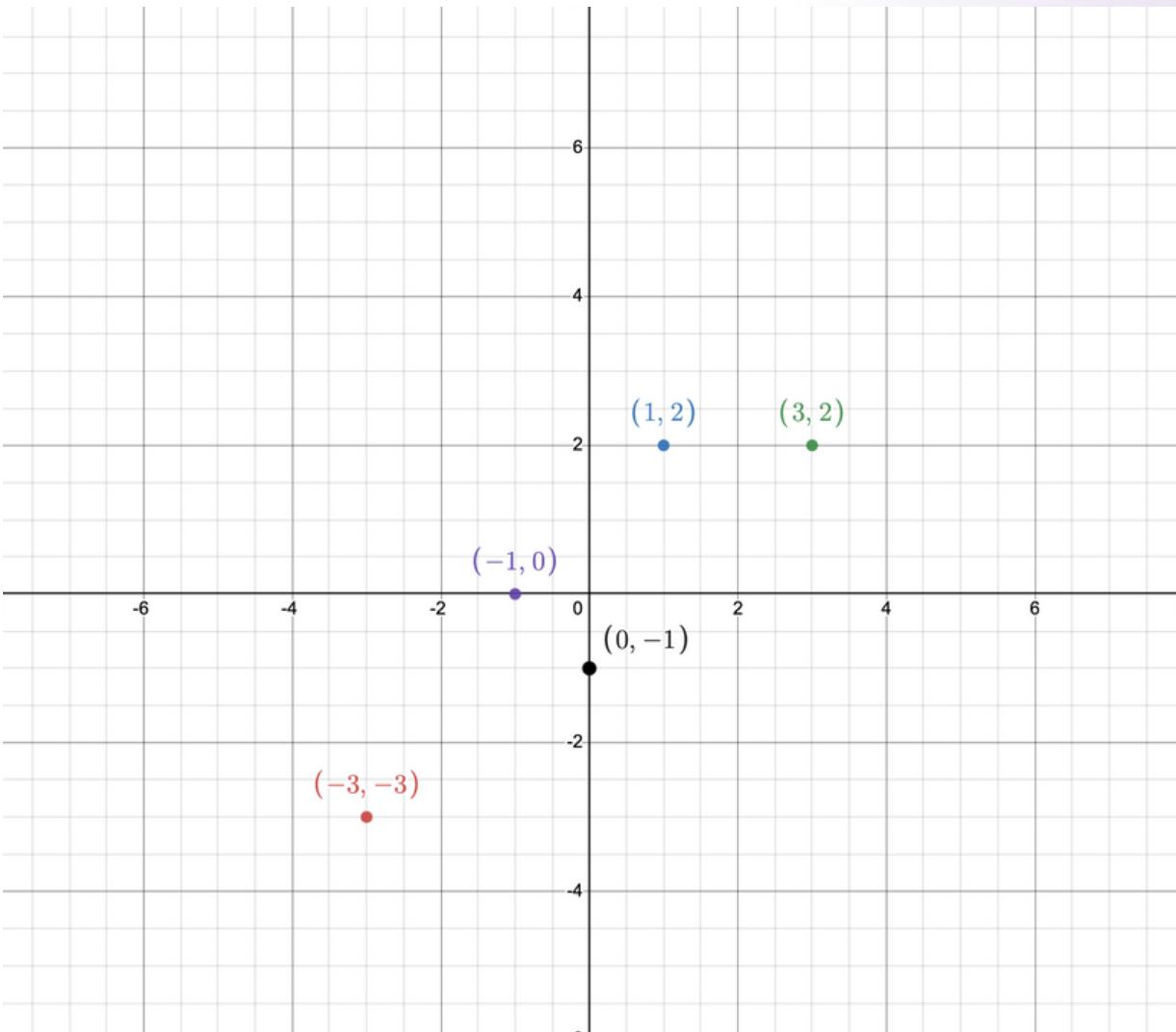
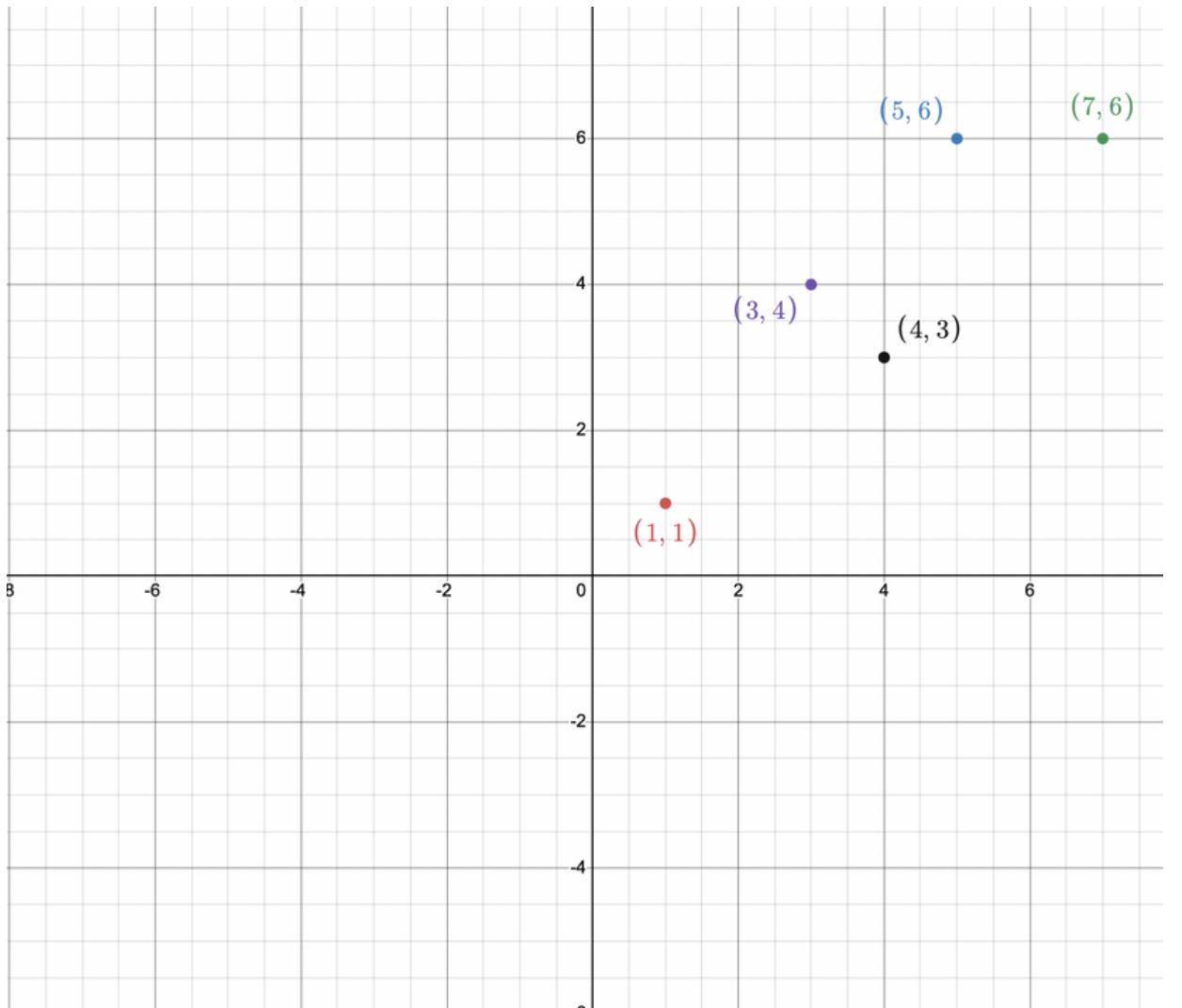
Eigenvectors
and eigenvalues



Theory

PCA

Standardisation



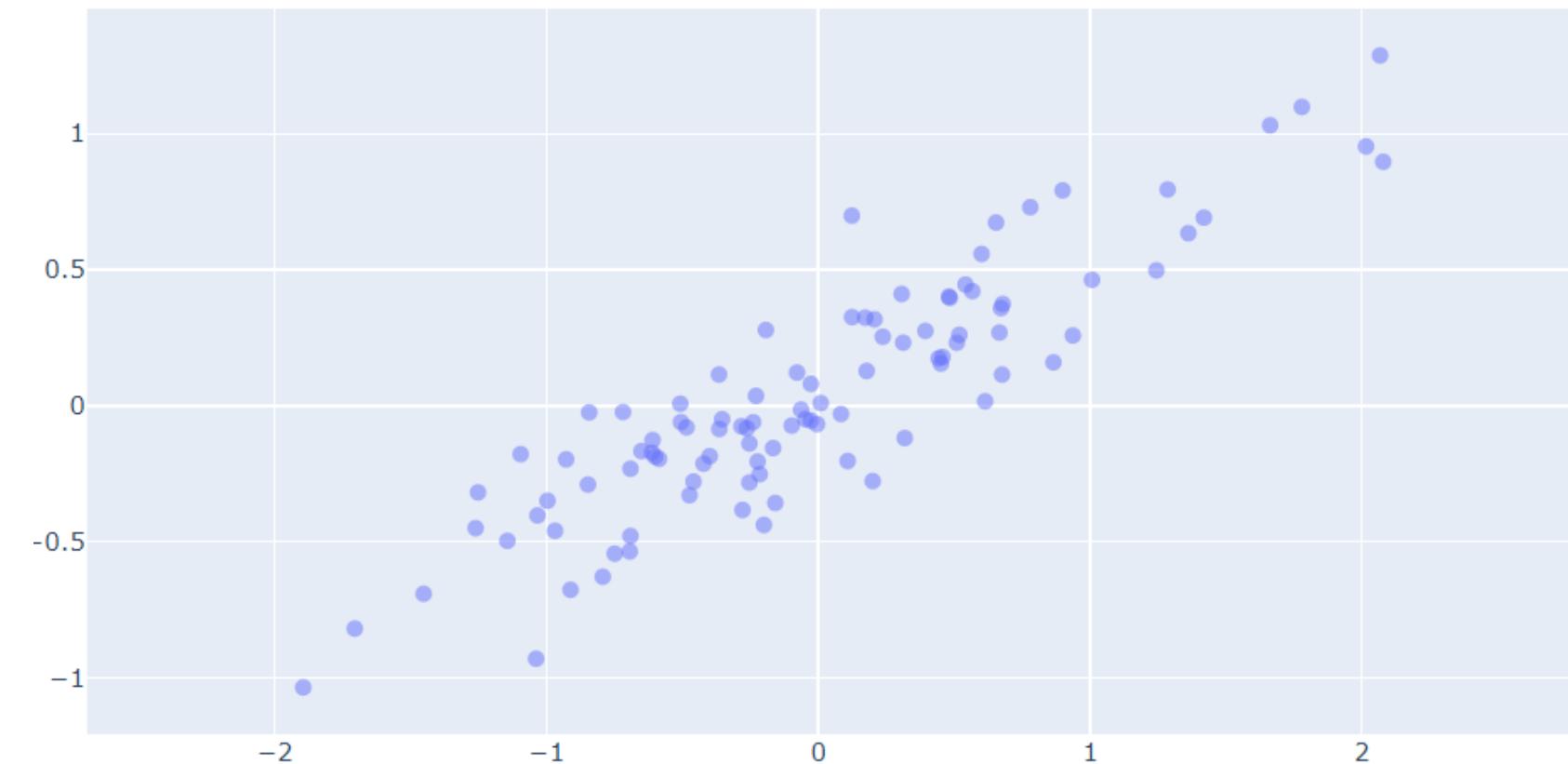
Theory

PCA

Covariance matrix

- Measuring the levels by which each pair of variables/attributes vary

$$\text{Cov}(x, y) = \frac{1}{N-1} \sum_{j=1}^N (x_j - \mu_x)(y_j - \mu_y)$$



→ $\begin{pmatrix} 2.5 & 3.0 \\ 3.0 & 10.0 \end{pmatrix}$

Theory

PCA

Eigenvalues and eigenvectors

$$\mathbf{A}\vec{v}_i = \lambda_i\vec{v}_i.$$

$$\mathbf{A} = \begin{bmatrix} -2 & 2 & 1 \\ -5 & 5 & 1 \\ -4 & 2 & 3 \end{bmatrix}$$



$$\det(\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} -2 - \lambda & 2 & 1 \\ -5 & 5 - \lambda & 1 \\ -4 & 2 & 3 - \lambda \end{vmatrix} = \mathbf{0}$$



$$\lambda = 3, 2, 1.$$



$$\mathbf{A}\vec{v}_i = \lambda_i\vec{v}_i$$

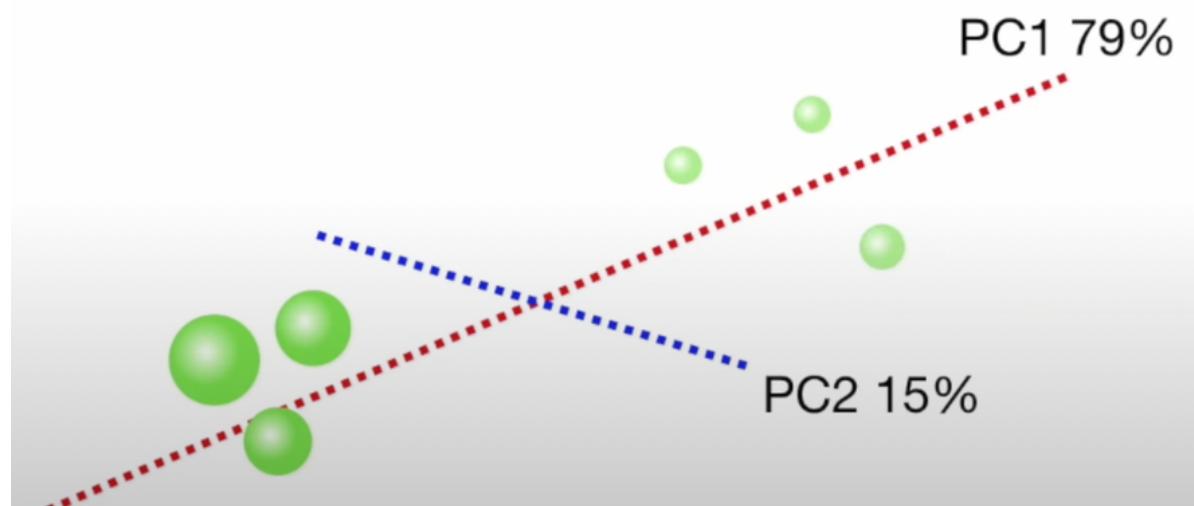
$$\lambda_1 = 3 : \hat{v}_1 = (1/\sqrt{6}, 2/\sqrt{6}, 1/\sqrt{6})^T$$

$$\lambda_2 = 2 : \hat{v}_2 = (1/\sqrt{6}, 1/\sqrt{6}, 2/\sqrt{6})^T$$

$$\lambda_3 = 1 : \hat{v}_3 = (1, 1, 1)^T / \sqrt{3}$$

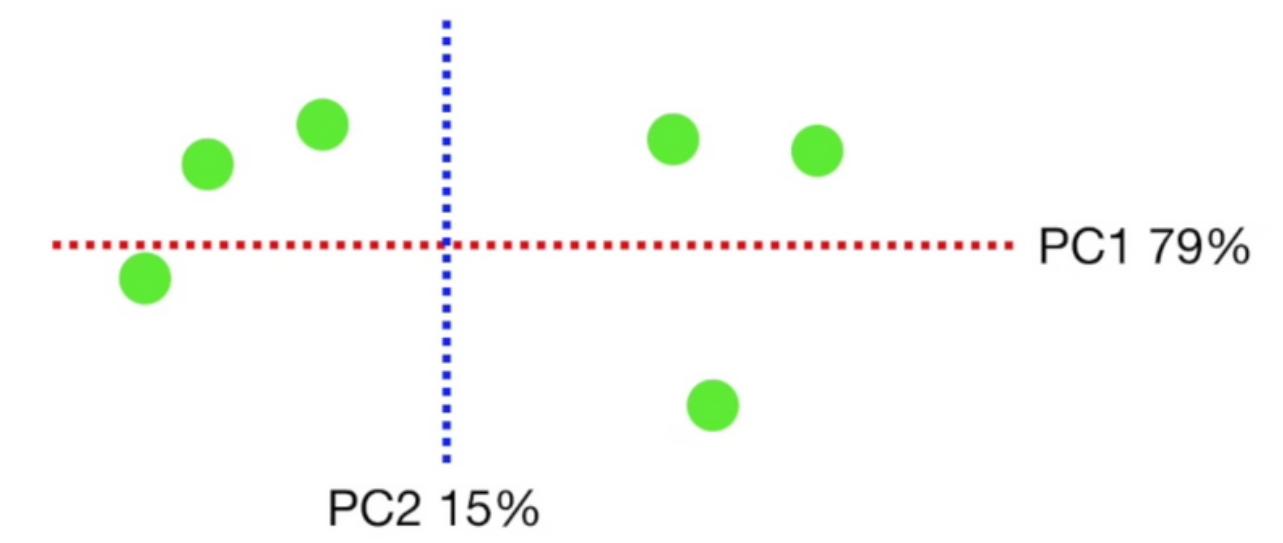
Theory

PCA

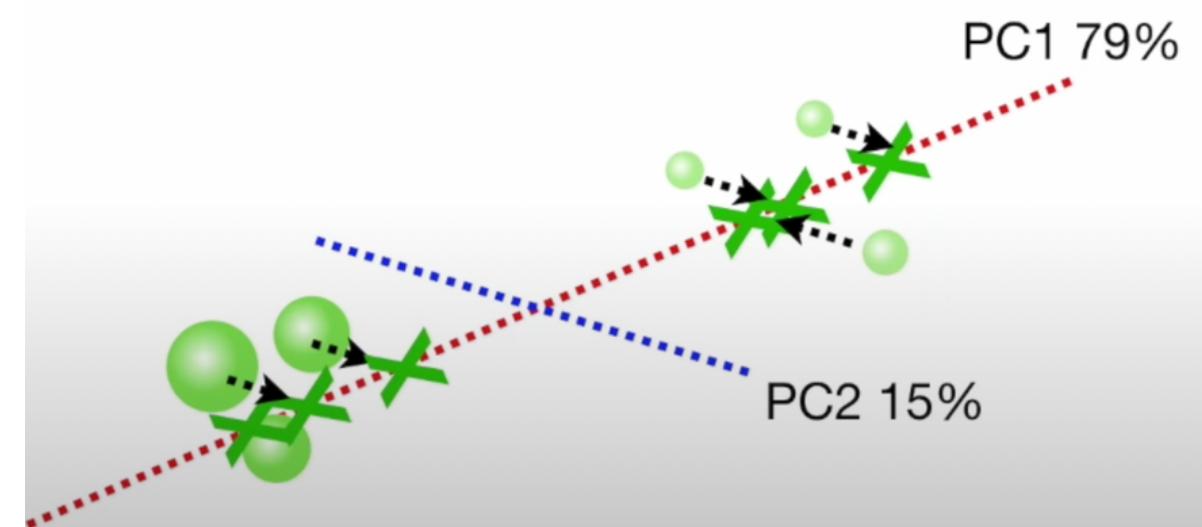


Find principal
components

Project all the
datapoints to the
new PCs



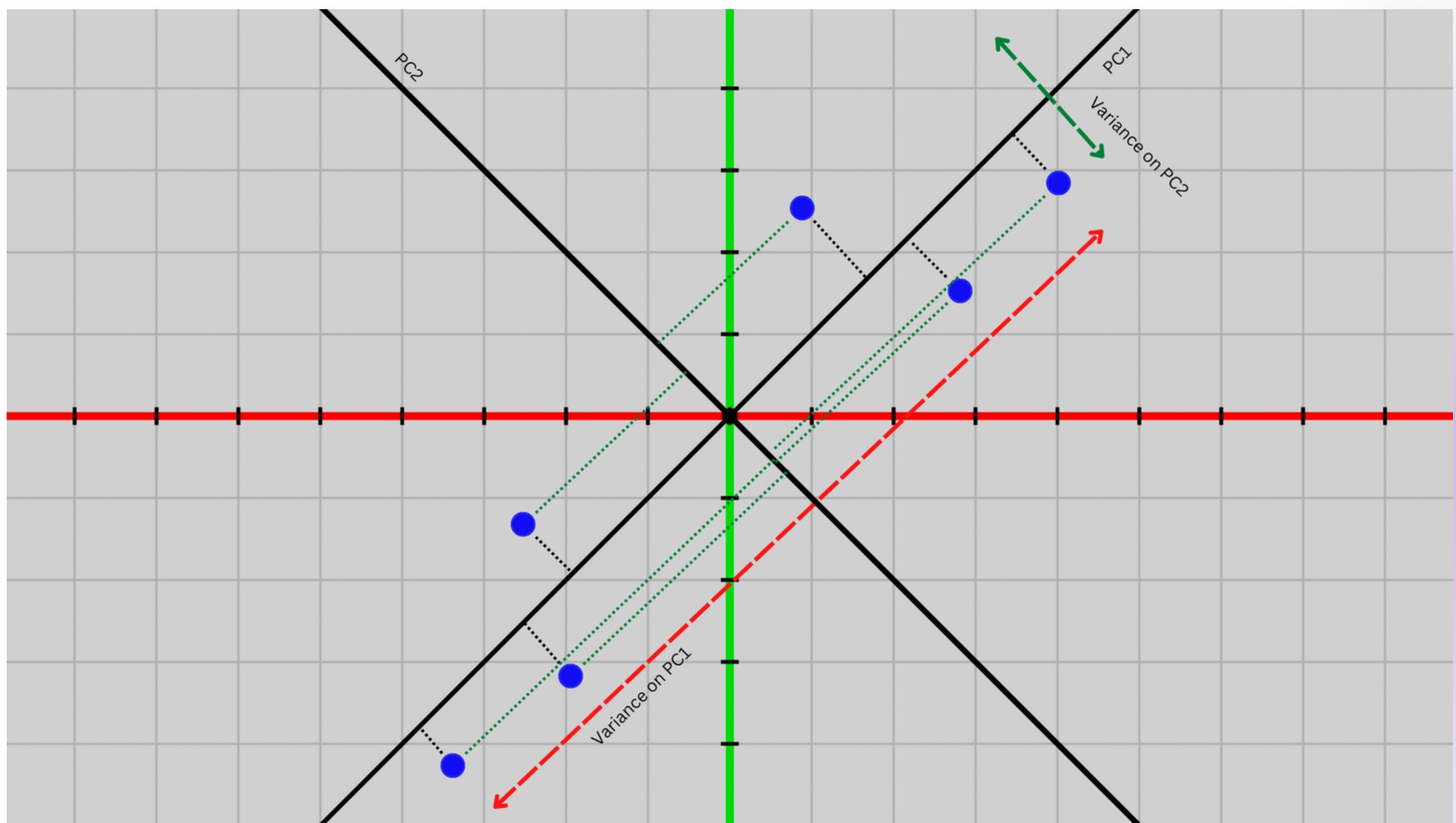
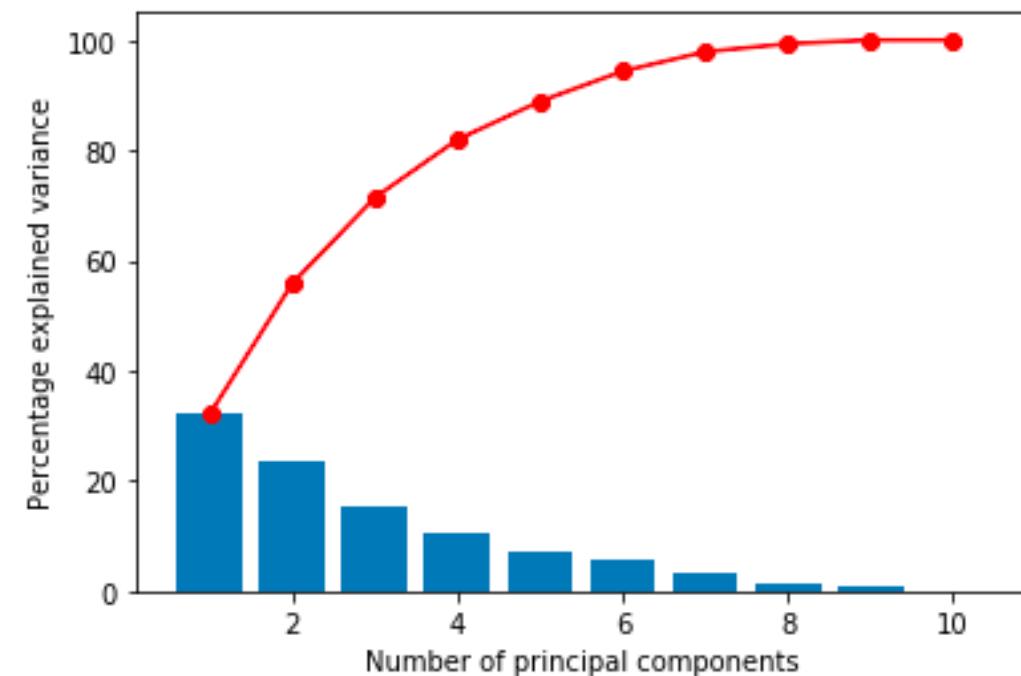
Final dataset



Theory

PCA

Explained variance!



Theory

PCA

PCA from scratch

```
import numpy as np

# Step 1: Standardisation
X_meaned = X - np.mean(X, axis=0)

# Step 2: Computing covariance matrix
cov_mat = np.cov(X_meaned, rowvar=False)

# Step 3: Finding eigenvalues and eigenvectors
eigenvalues, eigenvectors = np.linalg.eigh(cov_mat)
```

Theory

PCA

PCA from scratch

```
import numpy as np

# Step 4: Sorting the eigenvectors based on highest eigenvalues
sorted_index = np.argsort(eigenvalues)[::-1]
sorted_eigenvectors = eigenvectors[:, sorted_index]

# Step 5: Choosing n best eigenvectors
eigenvector_subset = sorted_eigenvectors[:, 0:num_components]

# Step 6: Creating the new dataset with n features
X_reduced = np.dot(eigenvector_subset.transpose(), X_meaned.transpose()).transpose()
```

Theory

PCA

PCA using libraries

```
from sklearn.preprocessing import StandardScaler  
  
sc = StandardScaler()  
X_train = sc.fit_transform(X_train)  
X_test = sc.transform(X_test)
```

Theory

PCA

Drawbacks

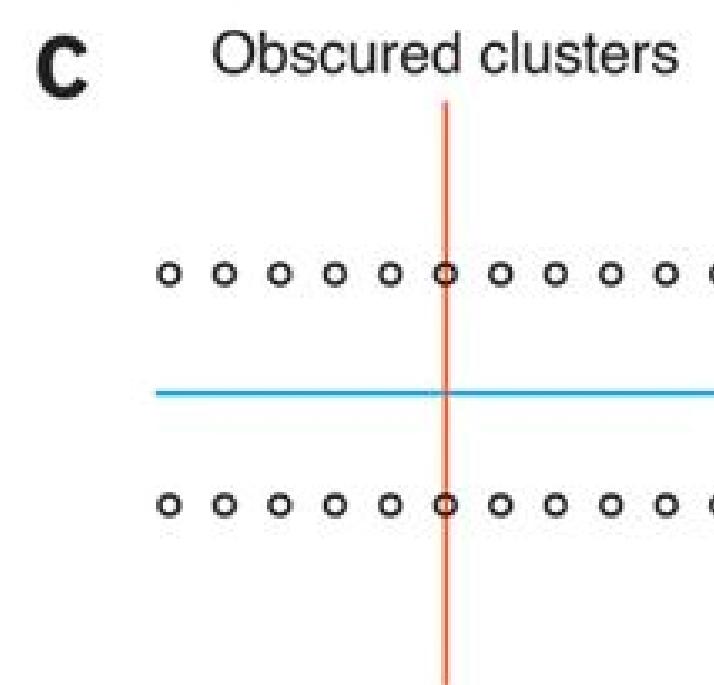
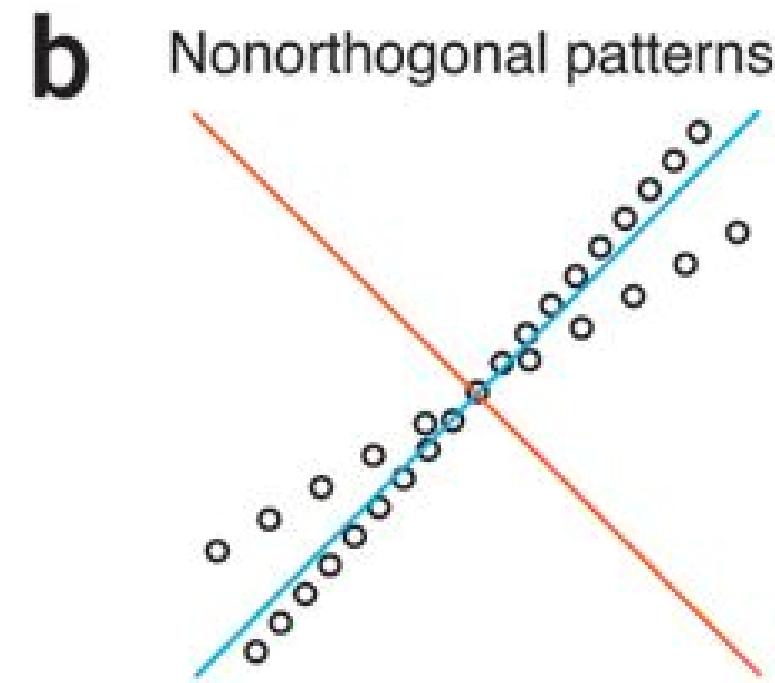
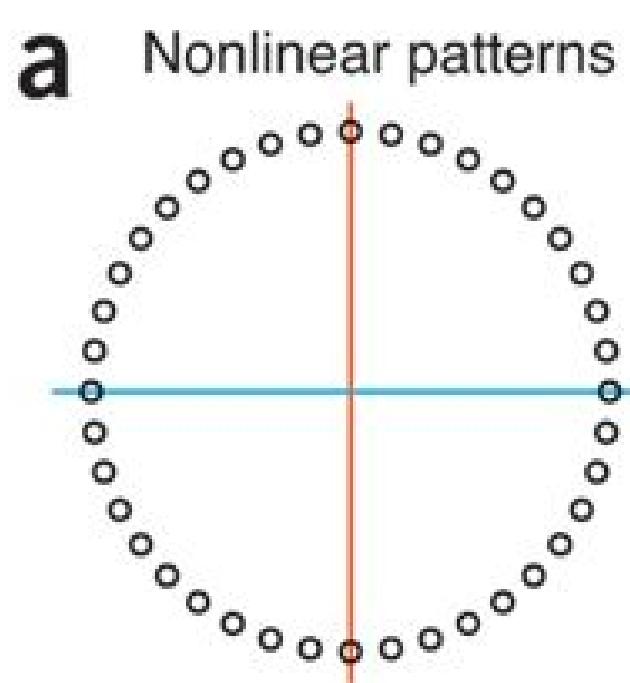
1. Doesn't work on certain data patterns

Theory

PCA

Drawbacks

1. Doesn't work on certain data patterns



2. Scaling matters
3. Interpretation of the PCs can be hard!

Theory

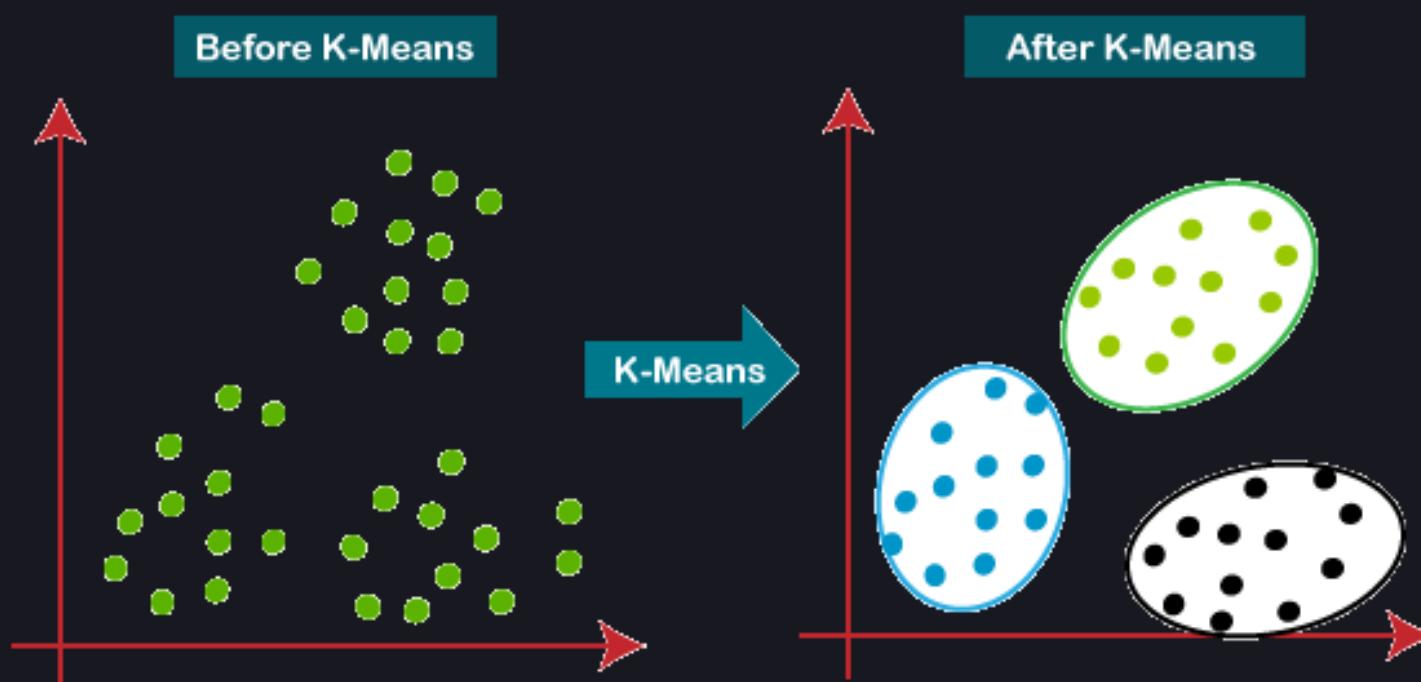
K-MEANS CLUSTERING

- Vector Quantization (Dividing vectors into groups)
- Can be used for classification
- Not to be confused with KNN!

Theory

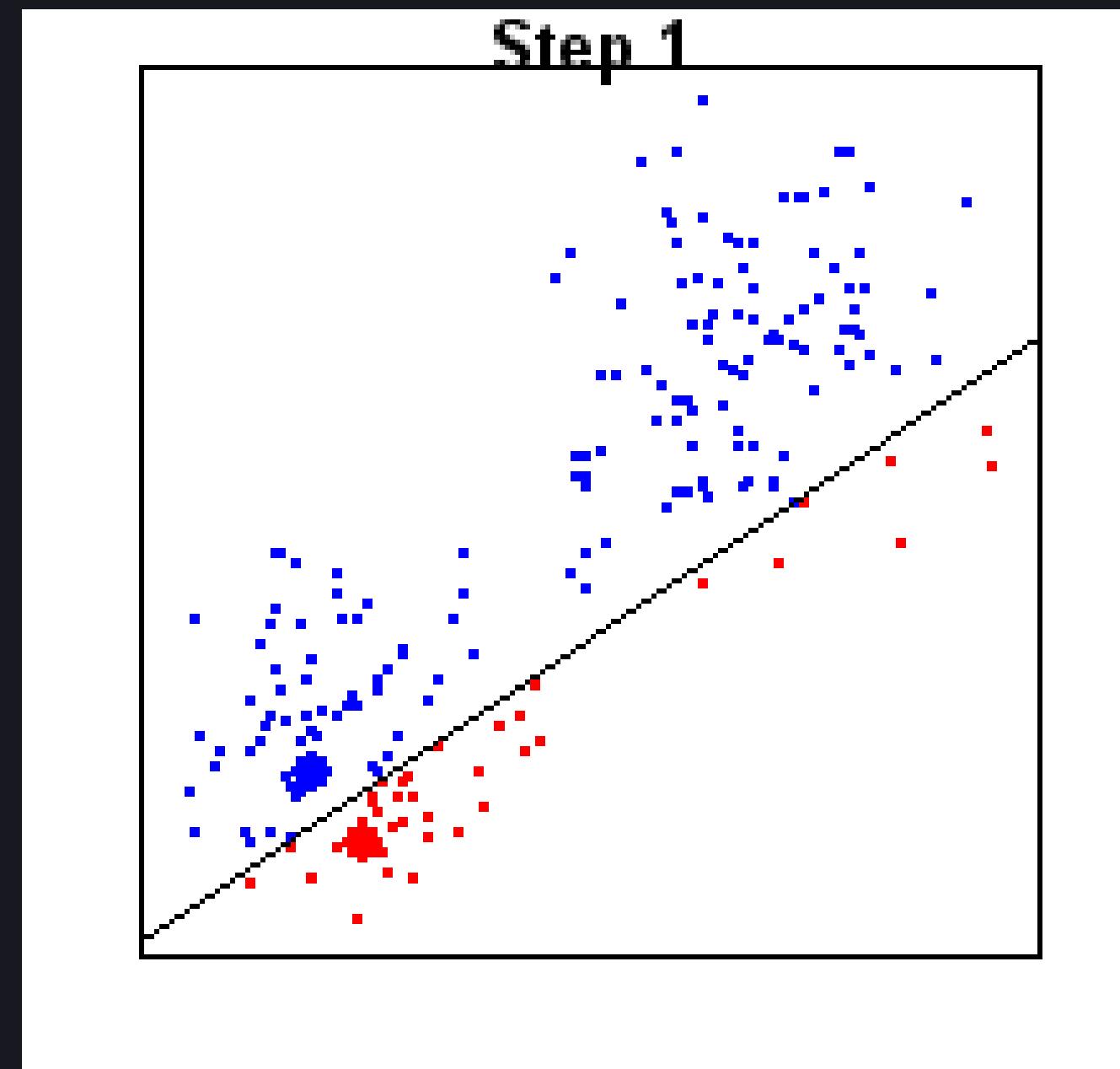
K-MEANS CLUSTERING

- Iterative process
- Partition points to their nearest mean
- Relatively simple algorithm



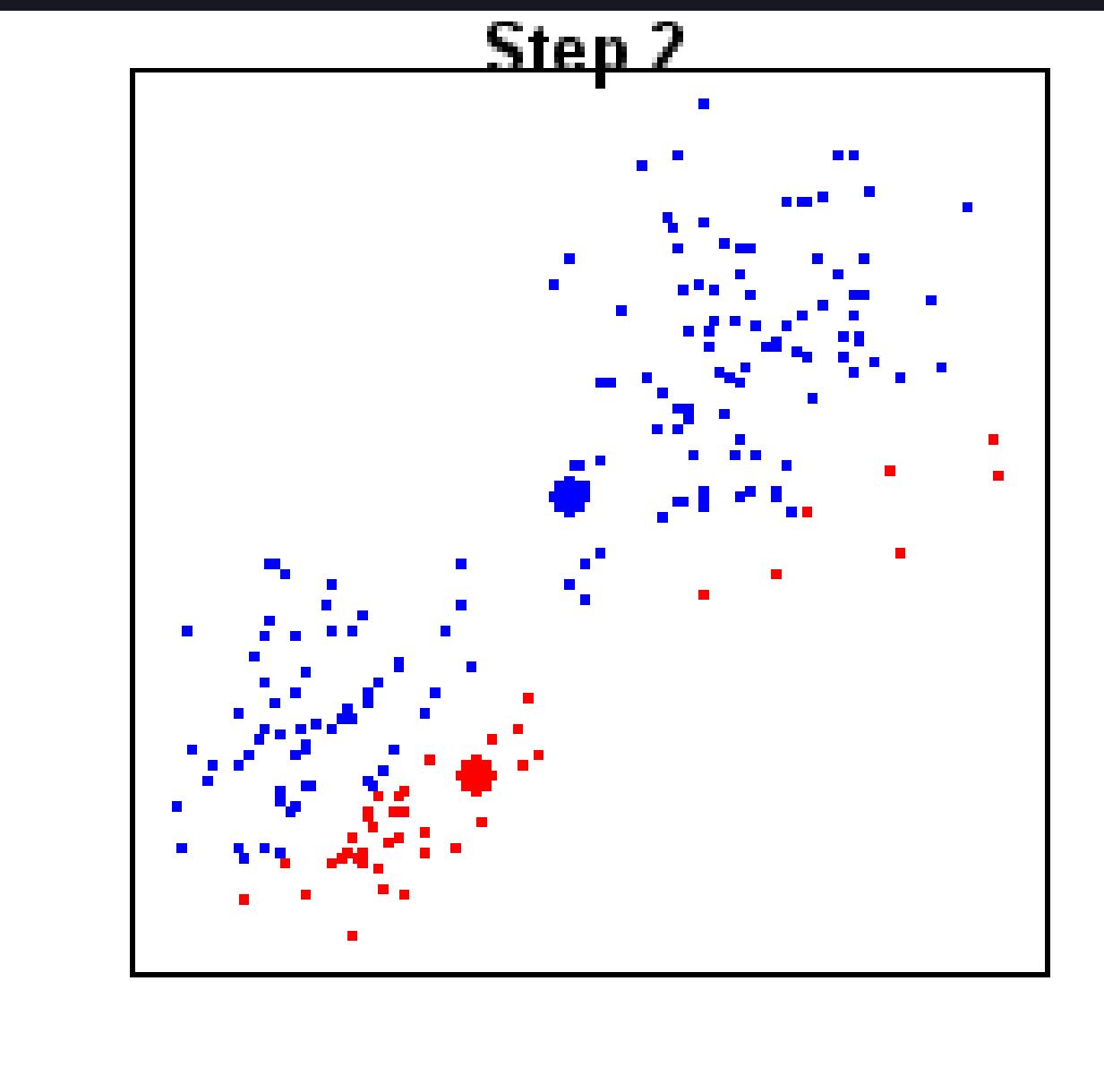
Theory

K-MEANS CLUSTERING



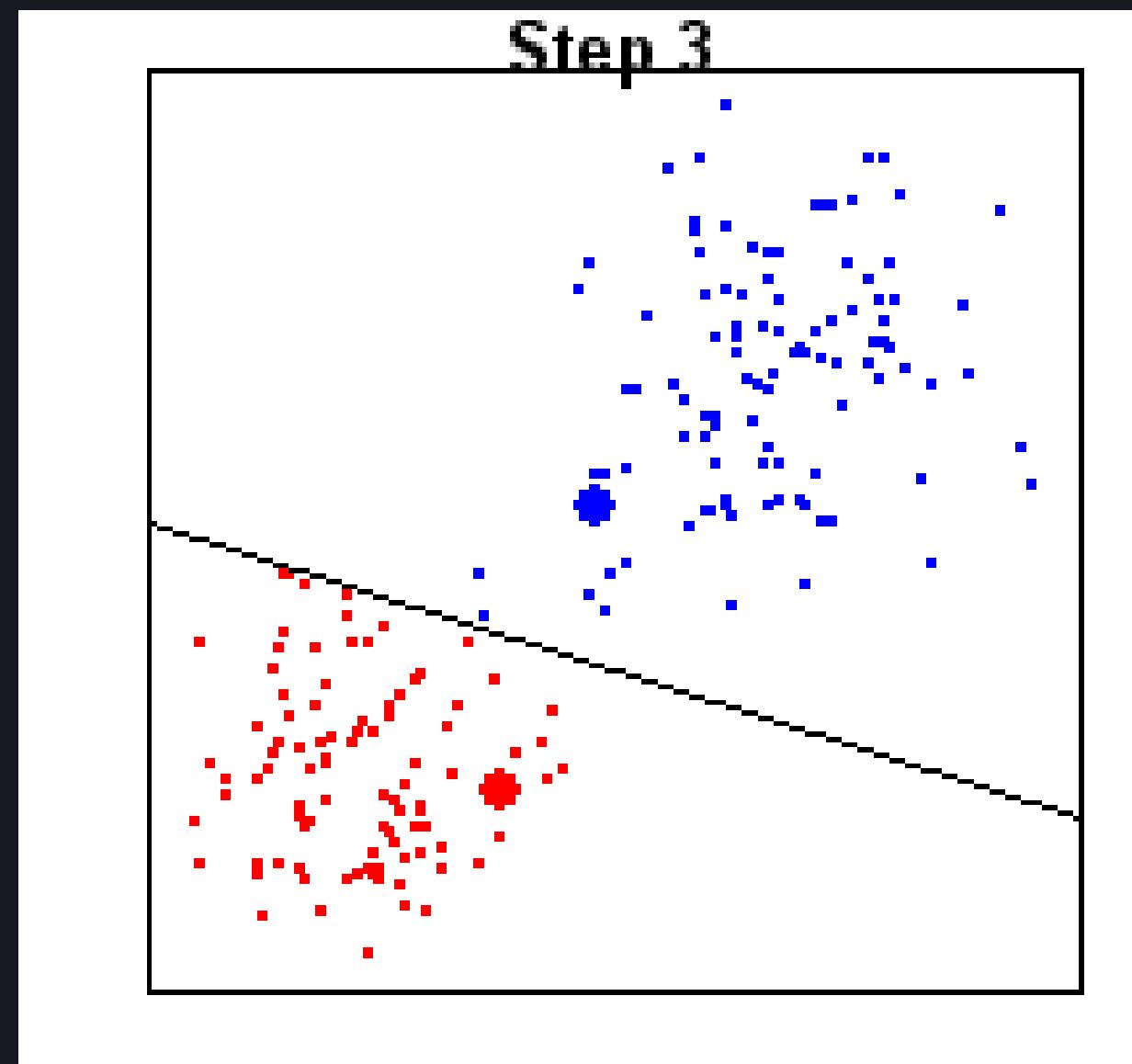
Theory

K-MEANS CLUSTERING



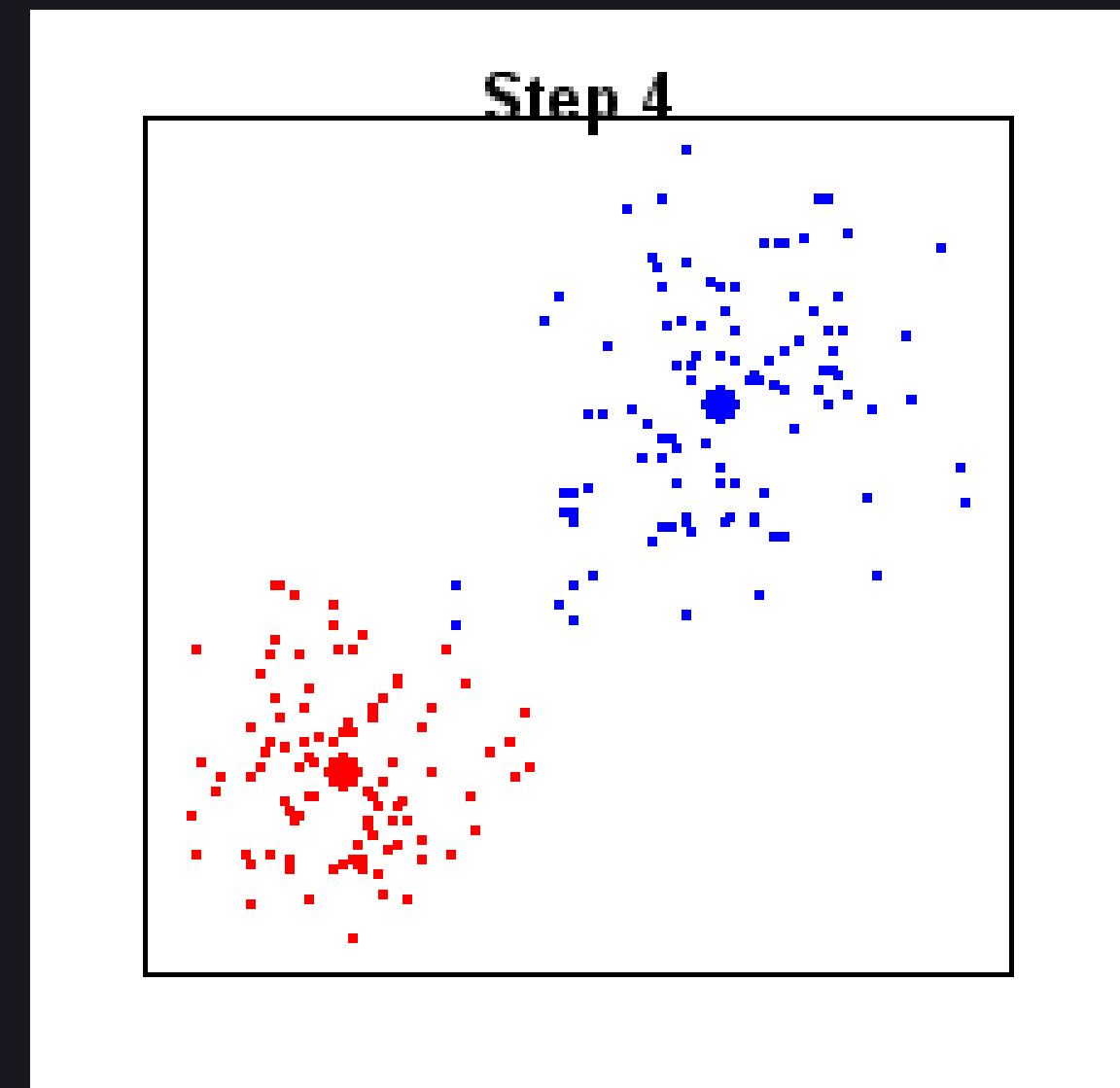
Theory

K-MEANS CLUSTERING



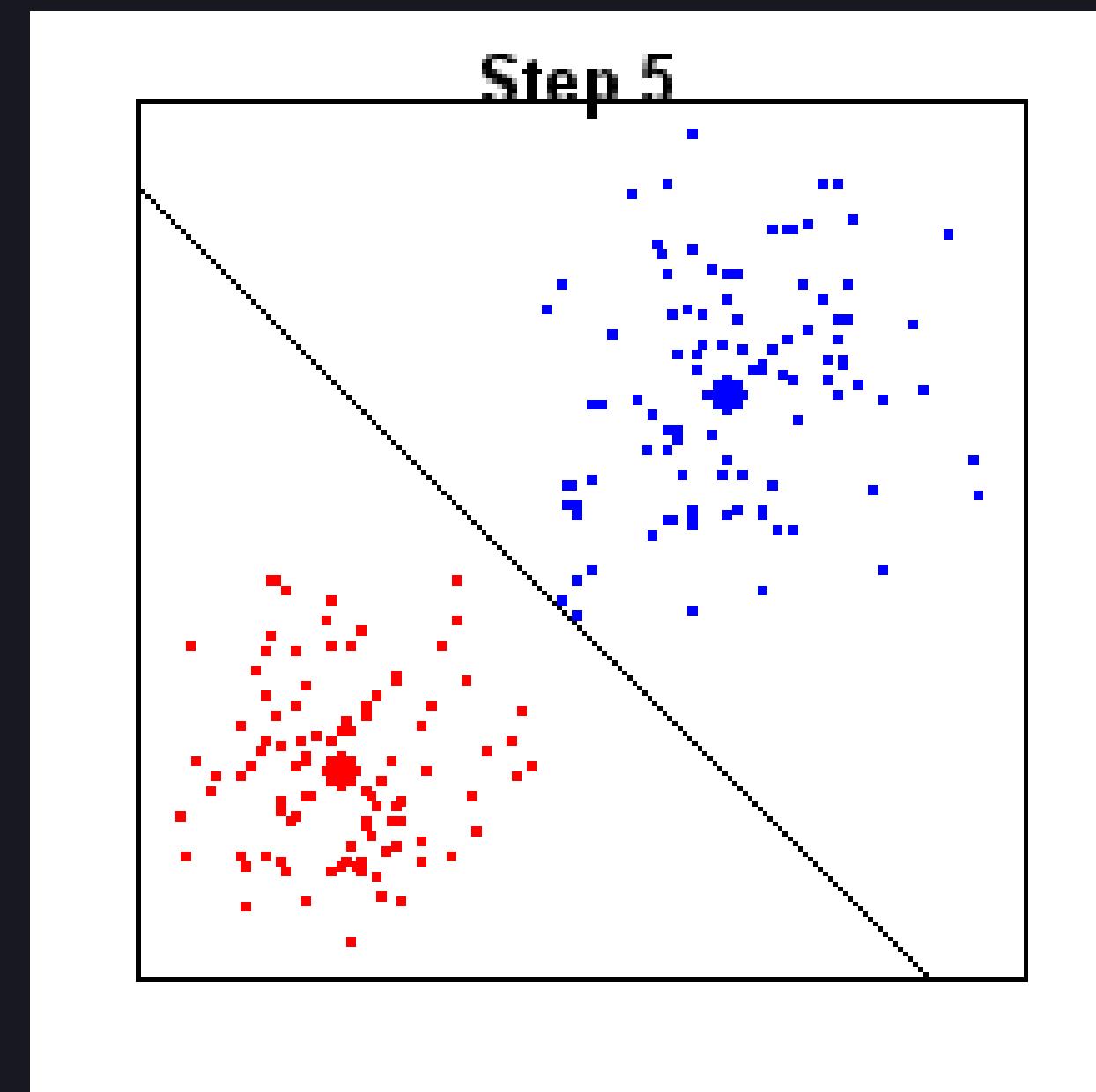
Theory

K-MEANS CLUSTERING



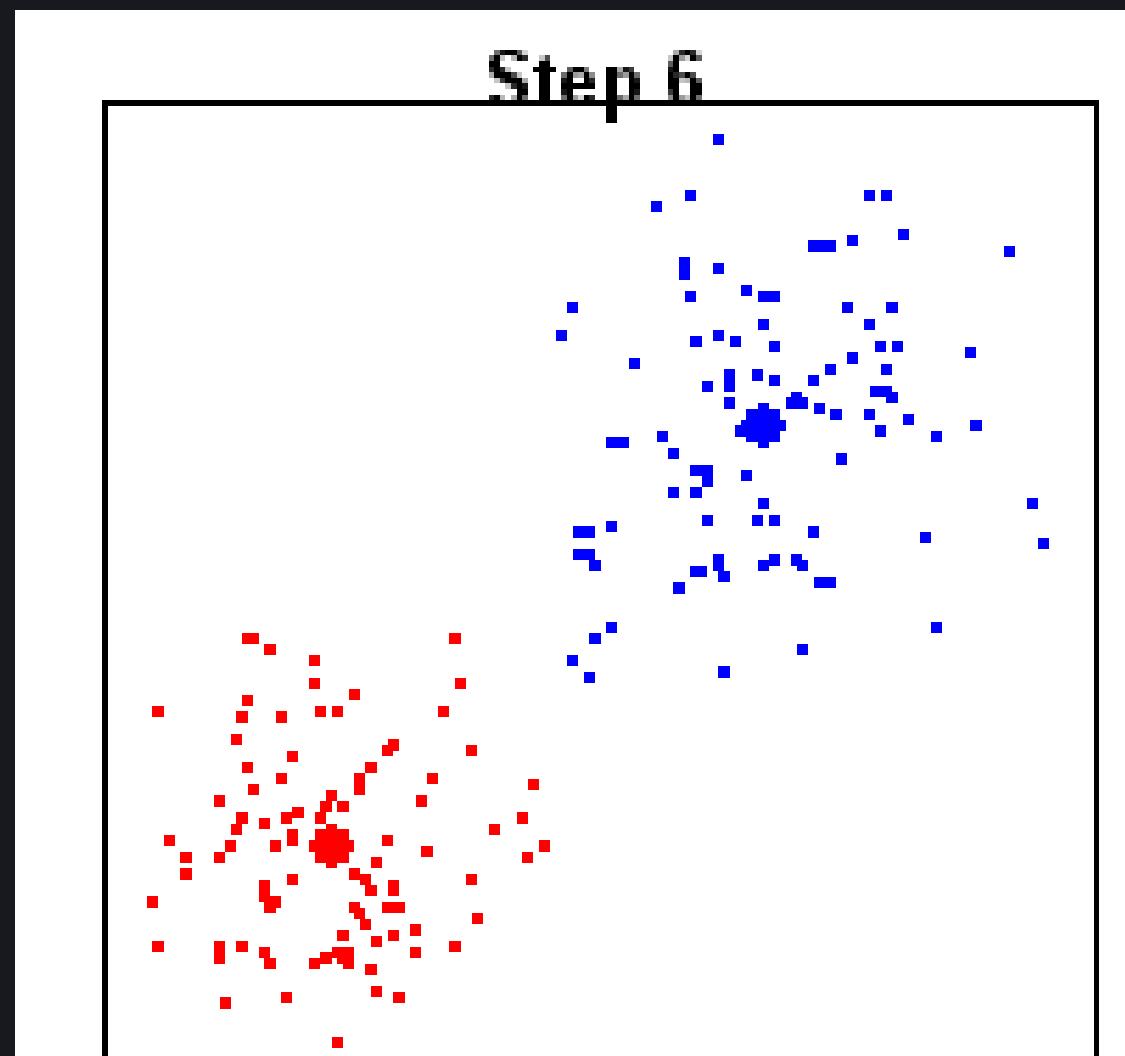
Theory

K-MEANS CLUSTERING



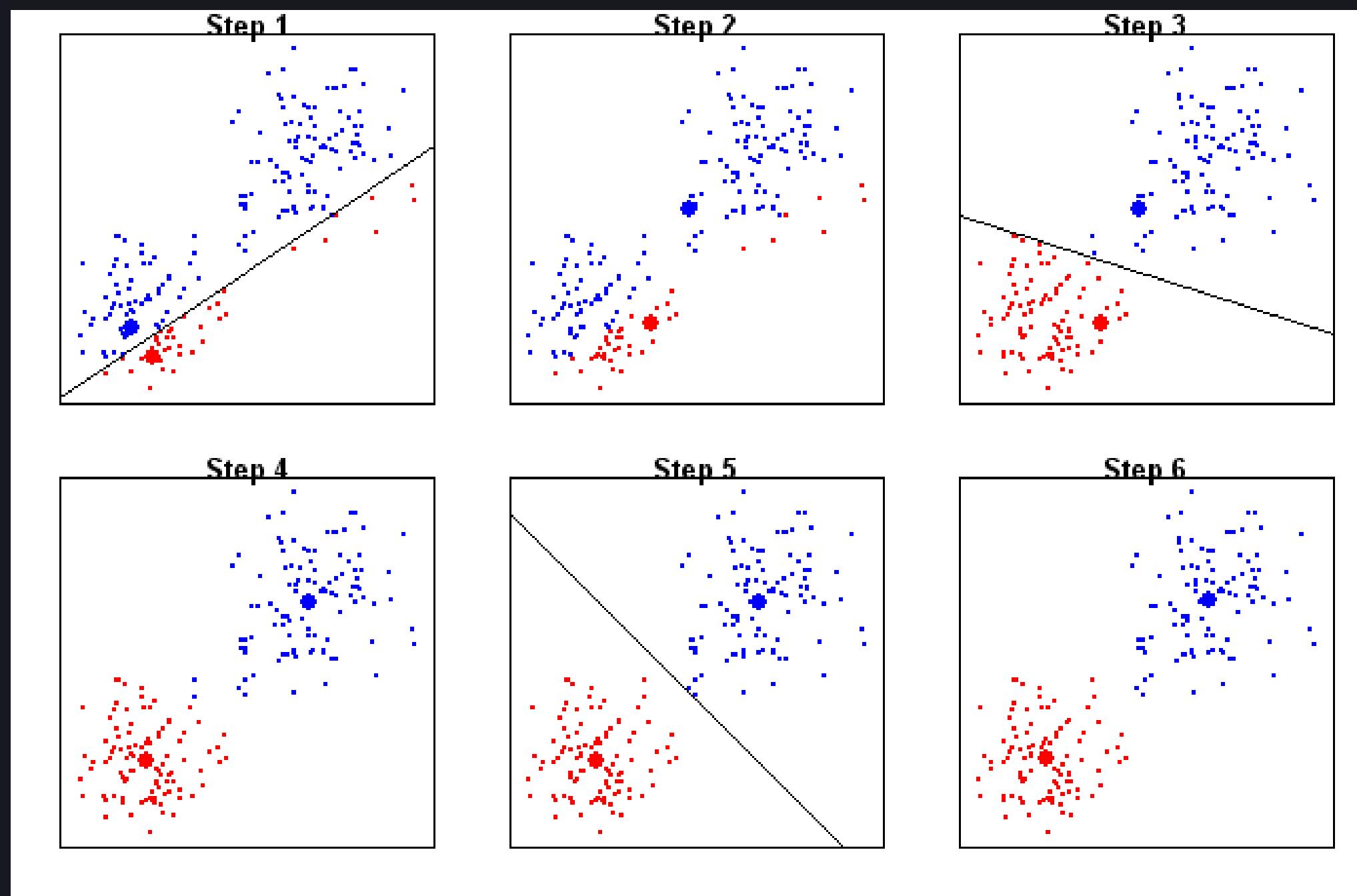
Theory

K-MEANS CLUSTERING



Theory

K-MEANS CLUSTERING



K-MEANS CLUSTERING

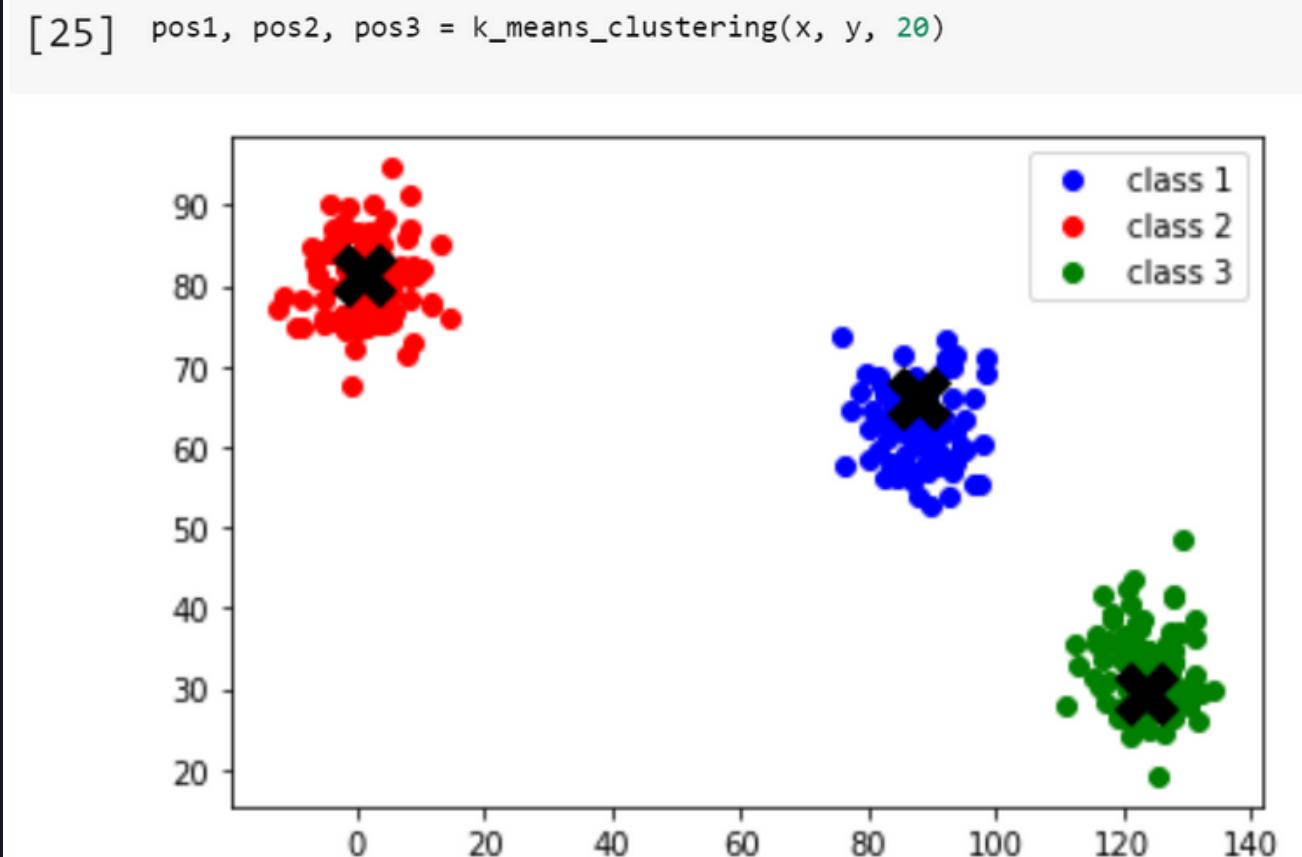
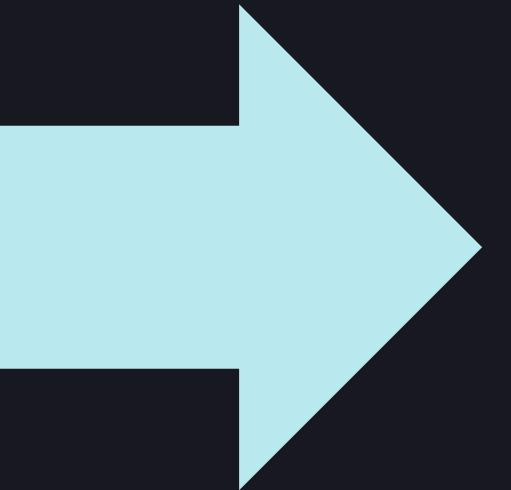
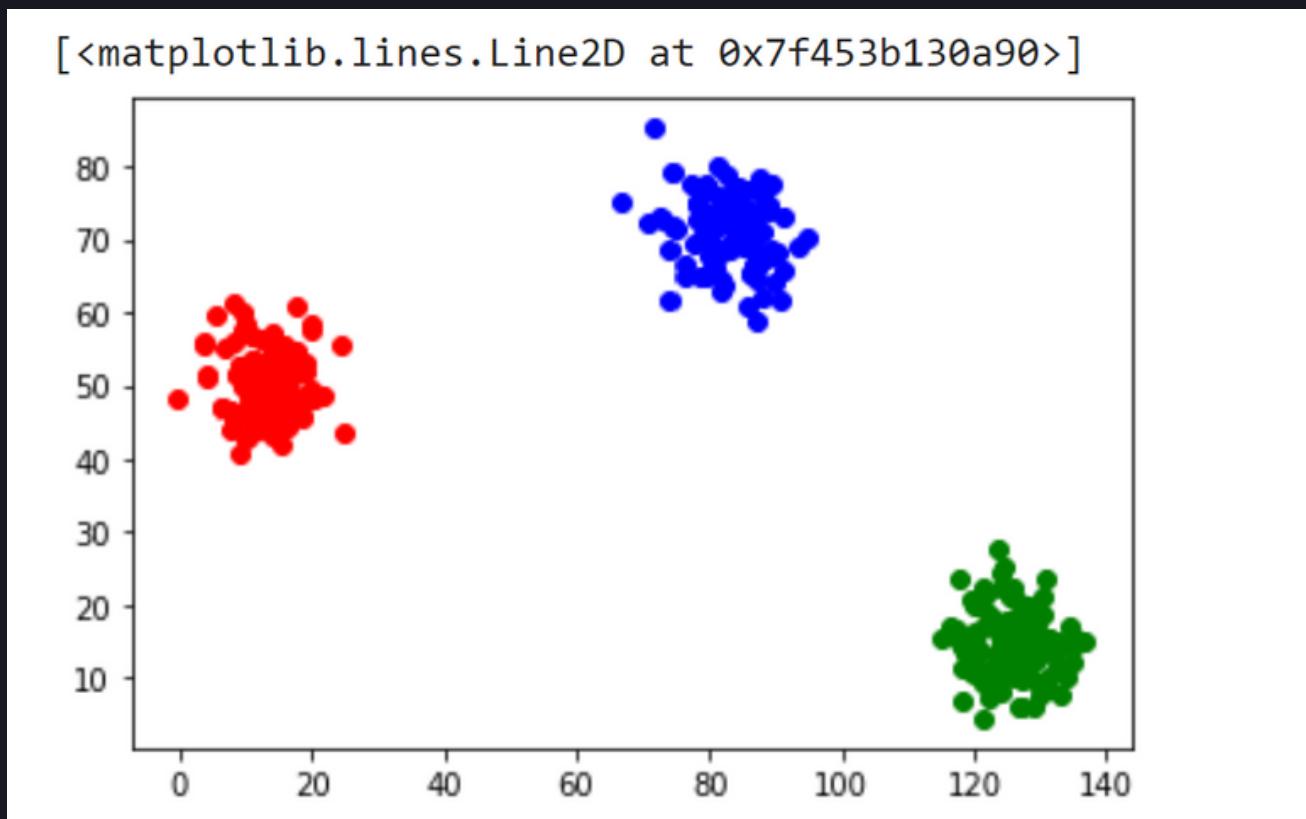
Drawbacks:

- The solution is NP-Hard, so we use heuristic algorithms to converge to local minima
- Does not guarantee global maximum
- The solution highly dependent on initialization and position of clusters
- Very difficult to choose k when number of classes unknown/data hard to visualize

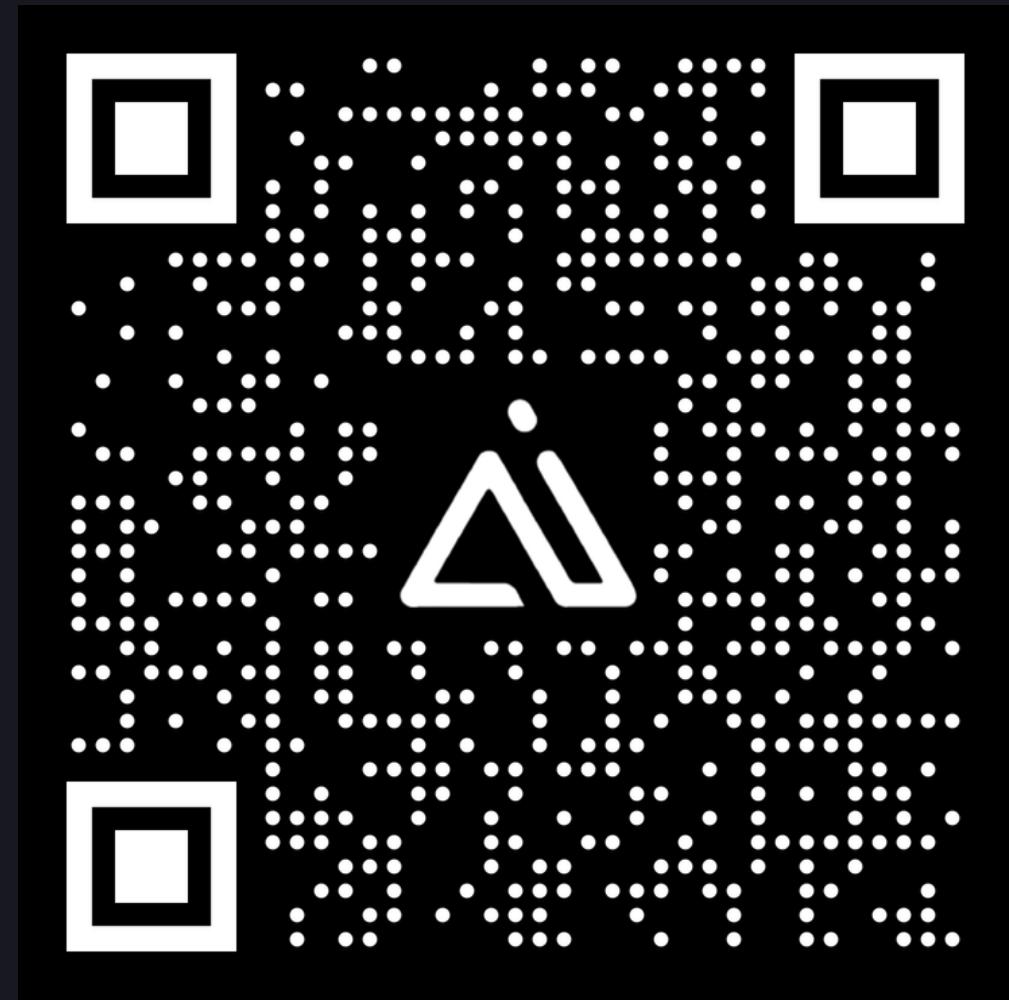
Theory

K-MEANS CLUSTERING

Implement your own algorithm!



QUESTIONS?



SEE YOU NEXT TIME