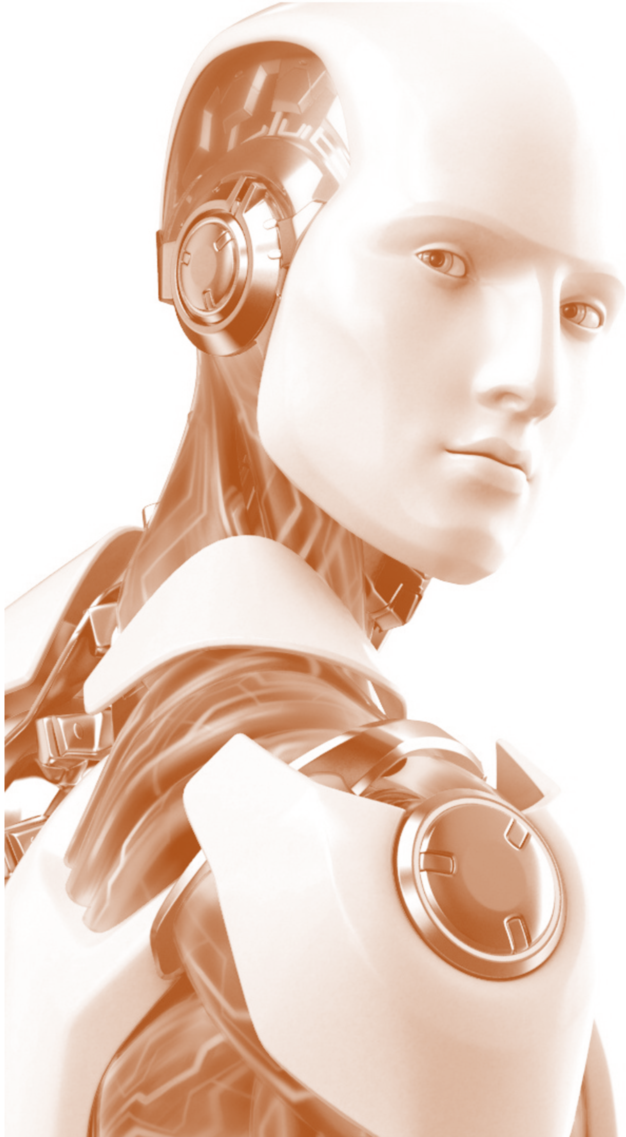


Machine Learning

Lecture 2 - Machine Learning Fundamentals

Profa. Dra. Esther Luna Colombini
esther@ic.unicamp.br

Prof. Dr. Alexandre Simoes
alexandre.simoes@unesp.br

LaRoCS – Laboratory of Robotics and Cognitive Systems

LAROCS

UNICAMP

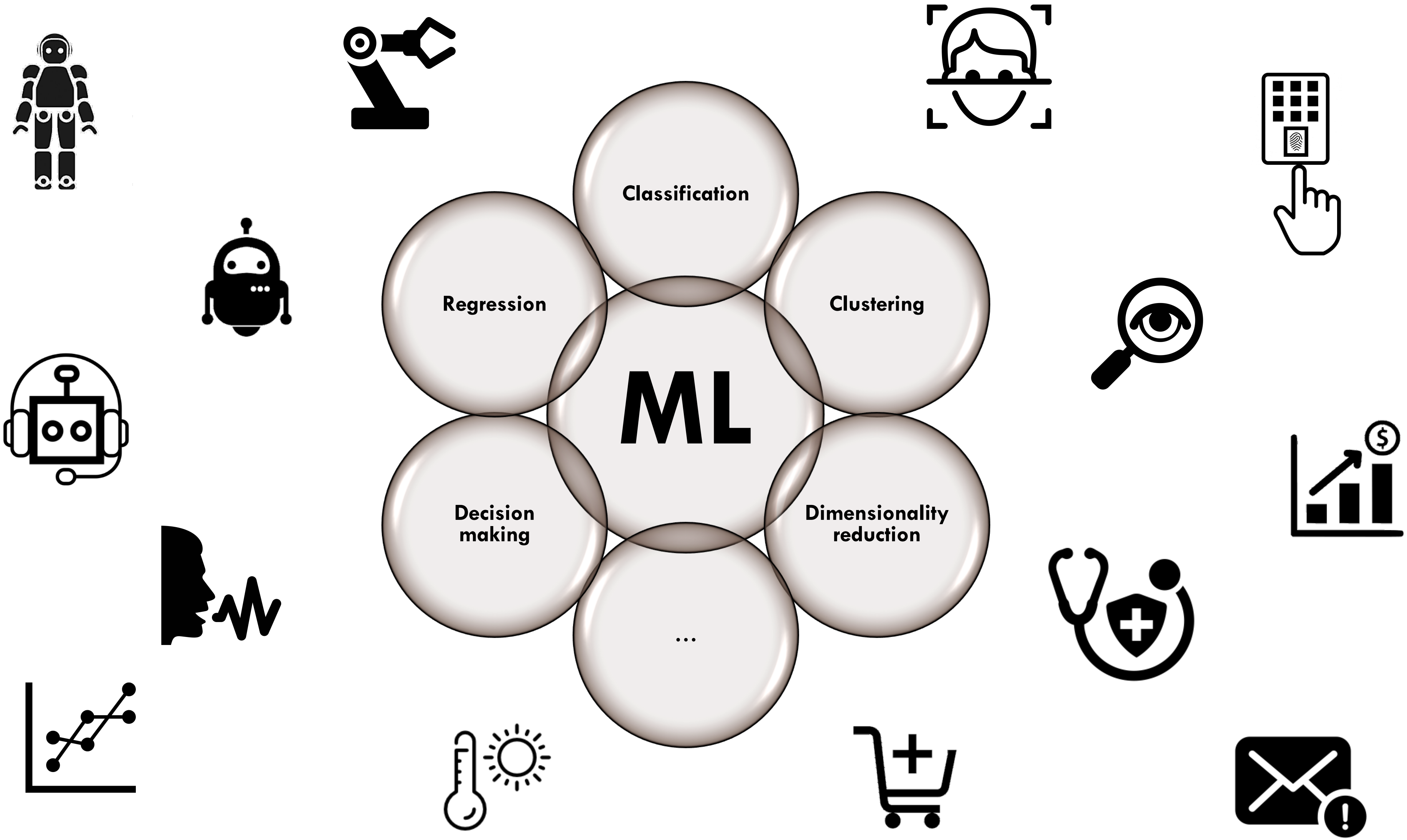
Advanced Institute for Artificial Intelligence

Summary

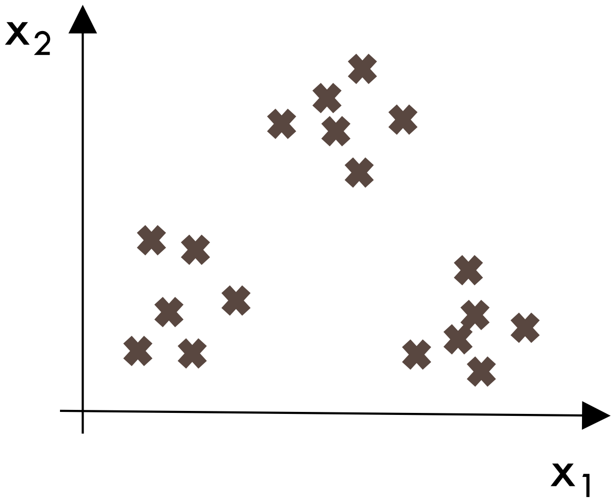
- ML typical tasks:
  - Regression
  - Clustering
  - Classification
  - Dimensionality reduction
  - Decision making
- Model training, performance and evaluation basic concepts:
  - Generalization
  - Underfitting
  - Overfitting
  - Data splitting
  - Cross-validation
  - Training early stopping
  - Data normalization

ML Typical Tasks

ML typical tasks

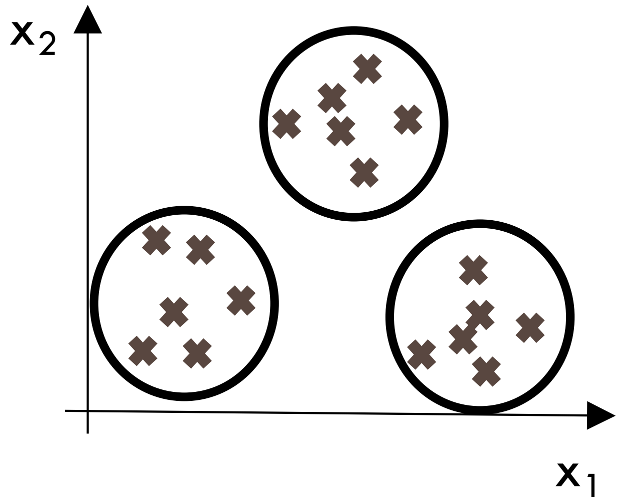


- Given some (typically unlabeled) data points, the goal is to group these points in such a way that points in the same group (or cluster) can be, **in some sense, more similar to each other** than points in other groups
- Common measures: distance between cluster members, density of clusters, statistical distribution...



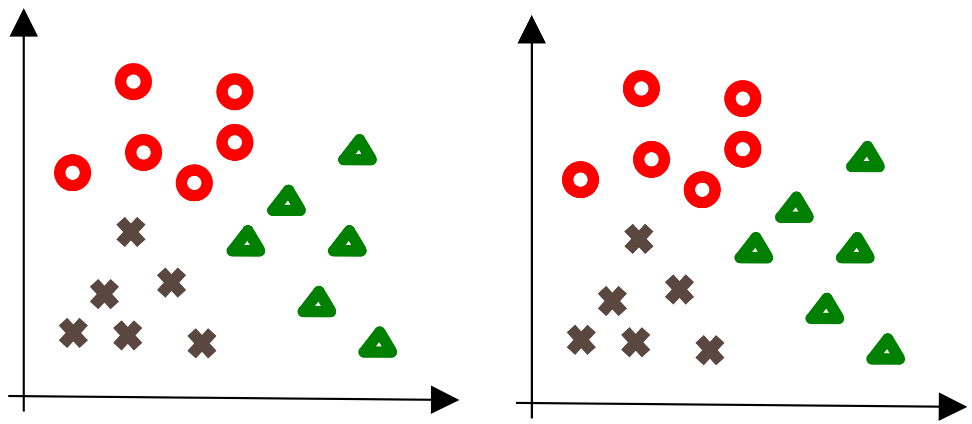
Clustering

- Given some (typically unlabeled) data points, the goal is to group these points in such a way that points in the same group (or cluster) can be, **in some sense, more similar to each other** than points in other groups
- Common measures: distance between cluster members, density of clusters, statistical distribution...



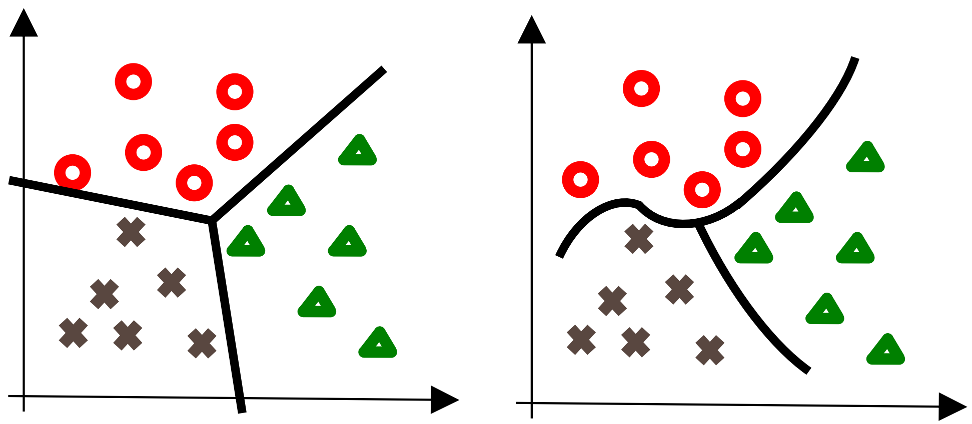
Classification

- Given a set of input data points and the classes they belong (labels), the goal is **to identify to which class a new observation belongs**



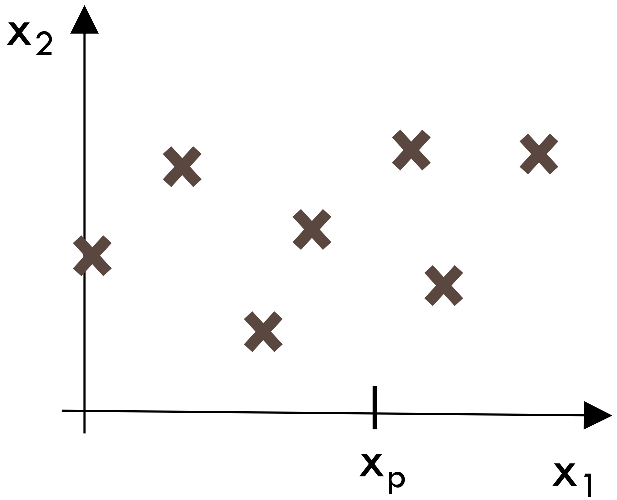
- Most of the classification methods look to establish **decision boundaries** to separate the different classes

- Given a set of input data points and the classes they belong (labels), the goal is **to identify to which class a new observation belongs**

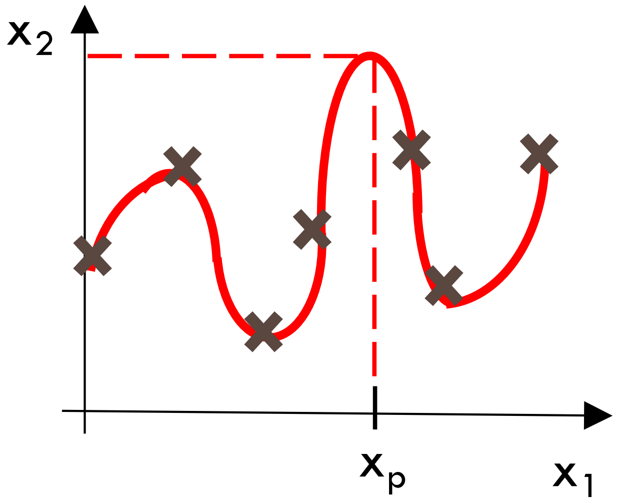


- Most of the classification methods look to establish **decision boundaries** to separate the different classes

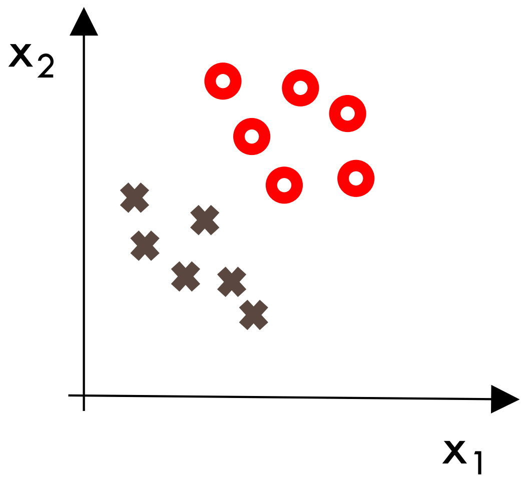
- Given some data points, the goal is to learn a **mathematical model** capable to fit data with a **curve,** so that the curve passes as close as possible to all of the data points
- This approach allows to **predict** the output for points that were not part of the original data



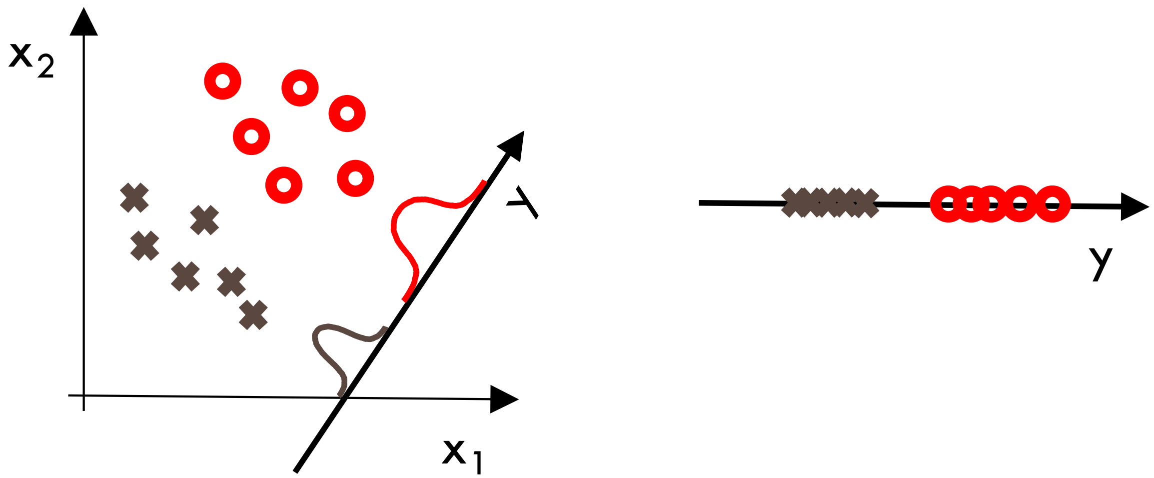
- Given some data points, the goal is to learn a **mathematical model** capable to fit data with a **curve**, so that the curve passes as close as possible to all of the data points
- This approach allows to **predict** the output for points that were not part of the original data



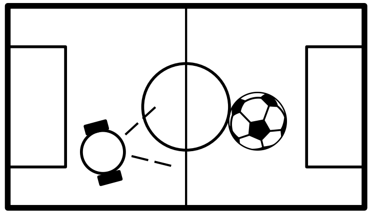
- Given a set of data points in a high number of dimensions (features or attributes), the goal is **to reduce the number of dimensions under consideration**
- Benefits: lower data volume, less computing required, shorter analysis time…



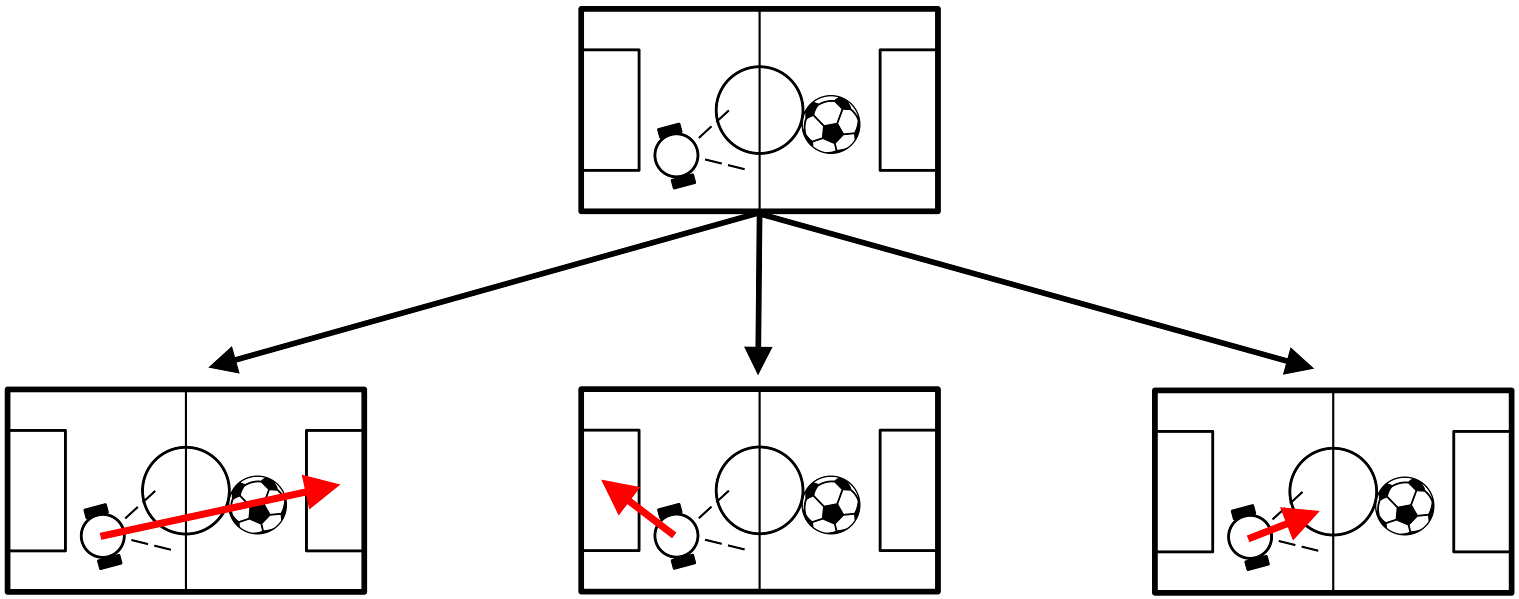
- Given a set of data points in a high number of dimensions (features or attributes), the goal is **to reduce the number of dimensions under consideration**
- Benefits: lower data volume, less computing required, shorter analysis time…



- Given a set of input data/information or even a environment, the goal is to learn models, concepts or policies that can **make agents able to make decisions**



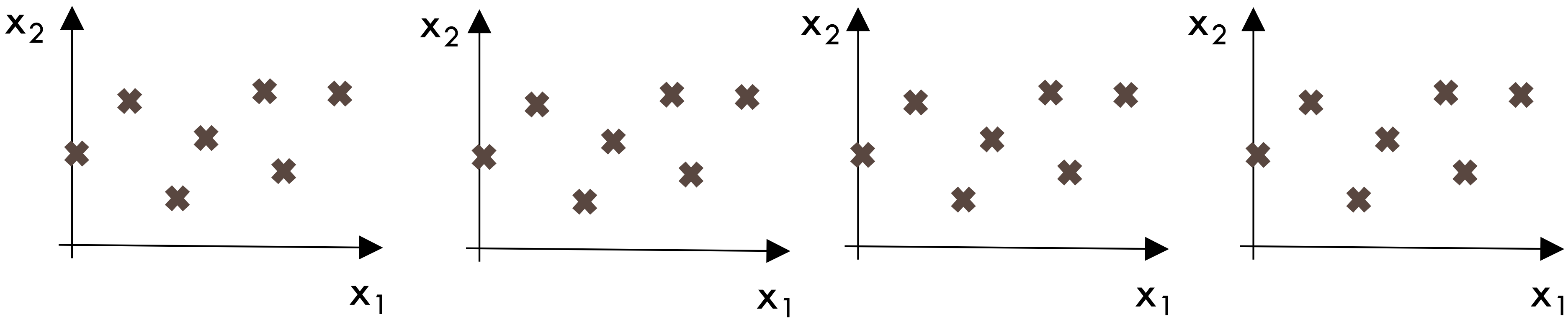
- Given a set of input data/information or even a environment, the goal is to learn models, concepts or policies that can **make agents able to make decisions**



Fundamental concepts

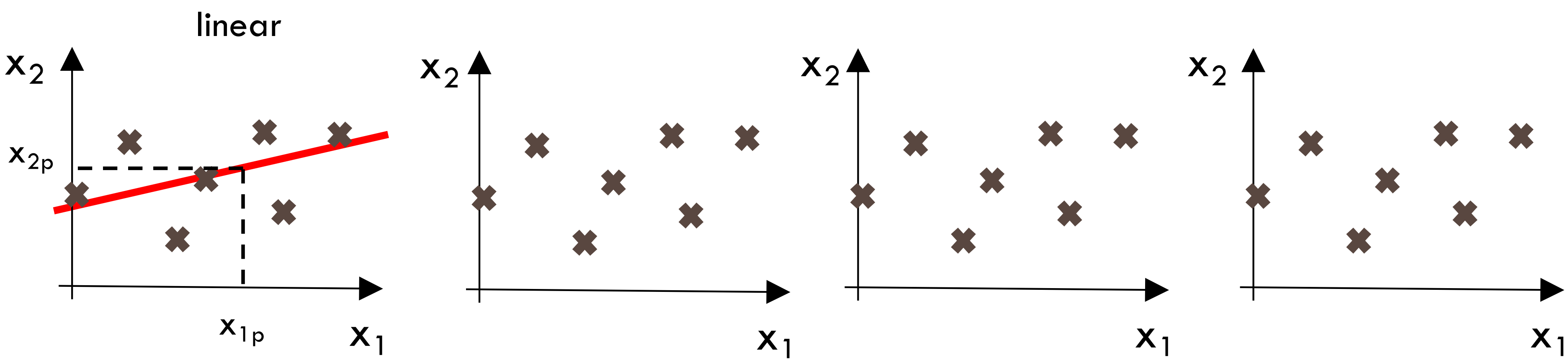
Generalization

- Which function best describes the data?



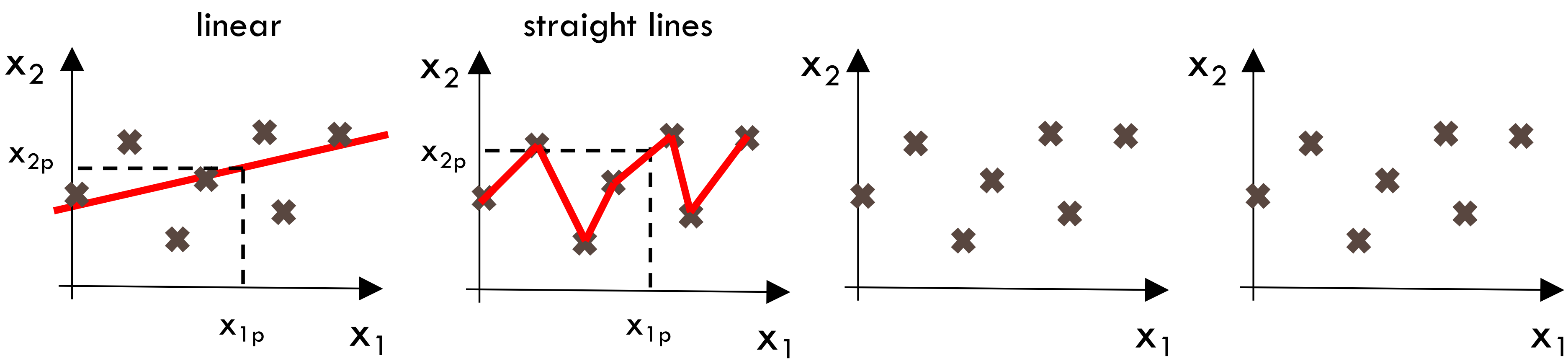
- In learning process, we are concerned with:
  1. Learning functions/models/concepts that will give us the correct outputs for the data present in the training set
  2. Learning functions/concepts/models that can **generalize** to instances similar, but not identical, to those in the training set
- **Generalization:** capacity to correctly predict values for unseen data

- Which function best describes the data?



- In learning process, we are concerned with:
  1. Learning functions/models/concepts that will give us the correct outputs for the data present in the training set
  2. Learning functions/concepts/models that can **generalize** to instances similar, but not identical, to those in the training set
- **Generalization**: capacity to correctly predict values for unseen data

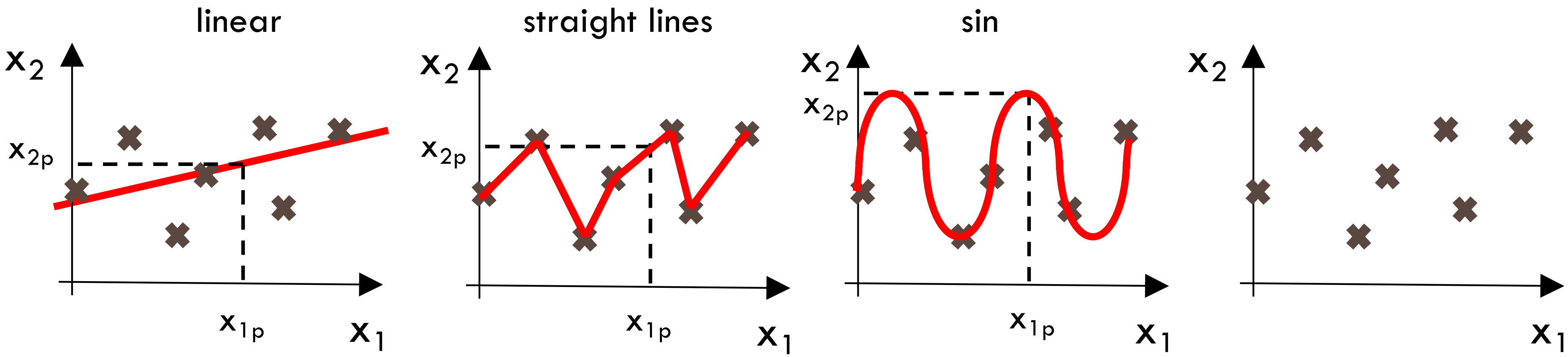
- Which function best describes the data?



- In learning process, we are concerned with:
  1. Learning functions/models/concepts that will give us the correct outputs for the data present in the training set
  2. Learning functions/concepts/models that can **generalize** to instances similar, but not identical, to those in the training set
- **Generalization**: capacity to correctly predict values for unseen data

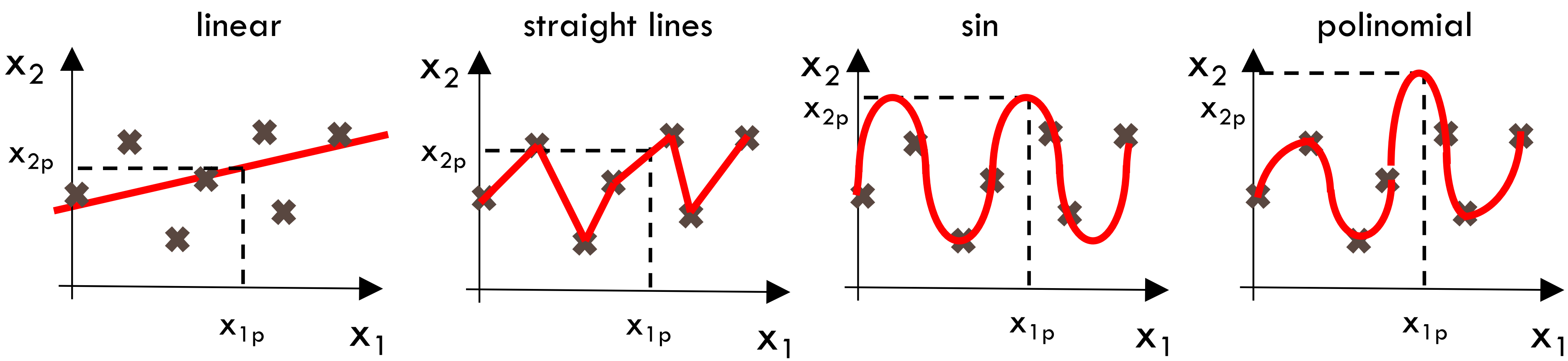
Generalization

- Which function best describes the data?



- In learning process, we are concerned with:
  1. Learning functions/models/concepts that will give us the correct outputs for the data present in the training set
  2. Learning functions/concepts/models that can **generalize** to instances similar, but not identical, to those in the training set
- **Generalization**: capacity to correctly predict values for unseen data

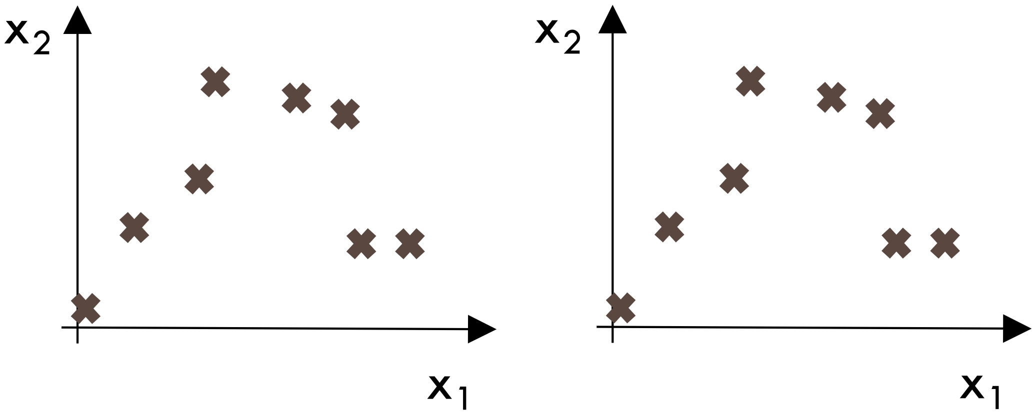
- Which function best describes the data?



- In learning process, we are concerned with:
  1. Learning functions/models/concepts that will give us the correct outputs for the data present in the training set
  2. Learning functions/concepts/models that can **generalize** to instances similar, but not identical, to those in the training set
- **Generalization:** capacity to correctly predict values for unseen data

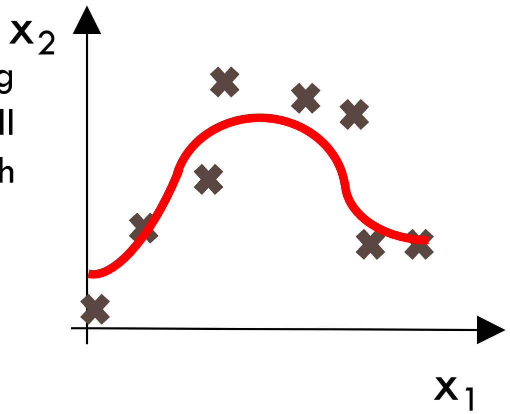
Underfitting

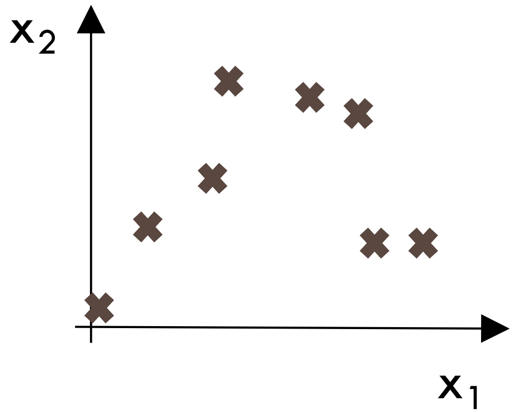
- Occurs when a model is **not capable** of adequately represent the training data ($\uparrow e_{training}$) nor generalize to new data
- It typically occurs due to:
  1. The system was **not trained enough**
  2. The model is **not complex enough** to adequately represent the data, even with more training



- Occurs when a model is **not capable** of adequately represent the training data (↑$e_{training}$) nor generalize to new data
- It typically occurs due to:
    1. The system was **not trained enough**
    2. The model is **not complex enough** to adequately represent the data, even with more training

A curve during a learning process ($e_{training}$ will typically decrease with more training)

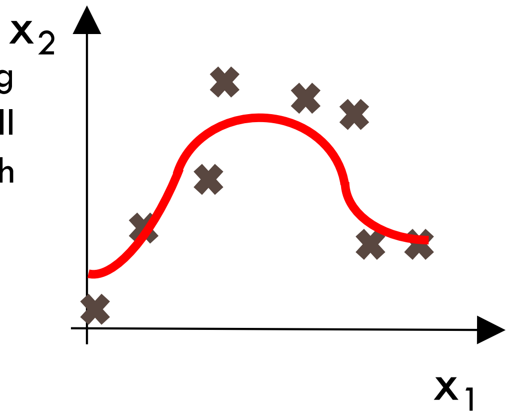


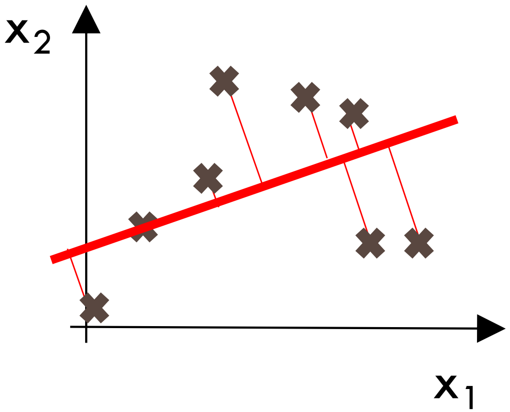


Underfitting

- Occurs when a model is **not capable** of adequately represent the training data ($\uparrow e_{training}$) nor generalize to new data
- It typically occurs due to:
  1. The system was **not trained enough**
  2. The model is **not complex enough** to adequately represent the data, even with more training

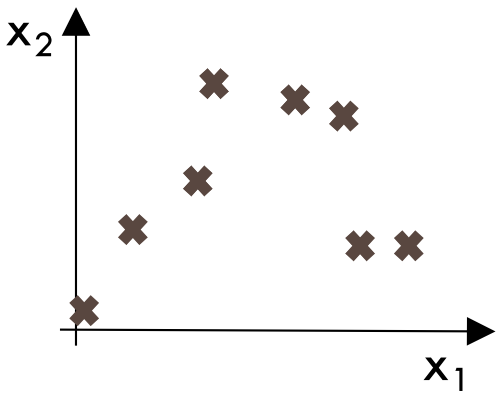
A curve during a learning process ($e_{training}$ will typically decrease with more training)

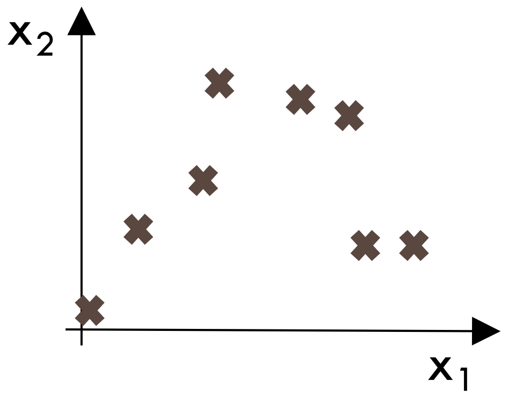




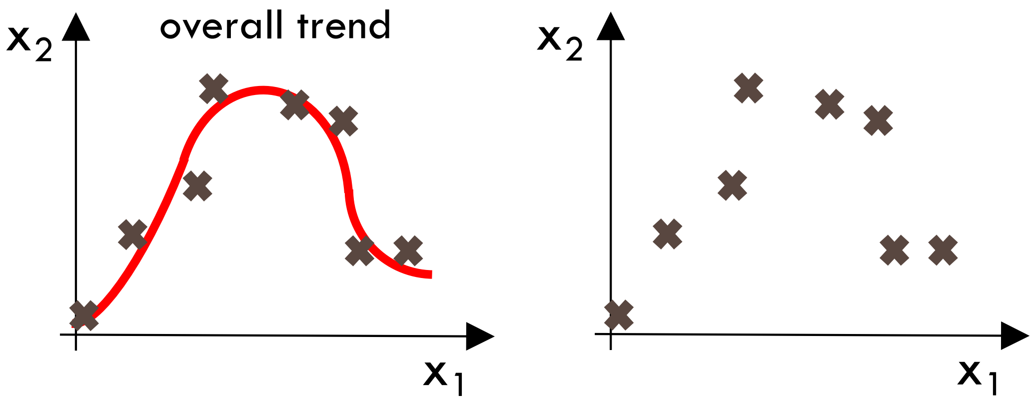
A first order model that may not be suitable for representing the data ($e_{training}$ typically will not decrease with more training)

- In well trained systems, typically curves will tend to fit **the overall trend** of the data ($e_{training}$>0)
- If the system is **trained for too long,** it tends to learn the noise and inaccuracies of the data. Models that are more complex than necessary will be learned, generating lower training errors (eventually $e_{training}$=0), but reducing the performance of the system with new data (↑$e_{generalization}$). This situation is called **overfitting.**

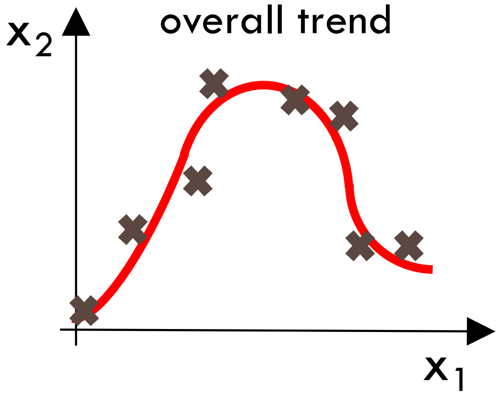


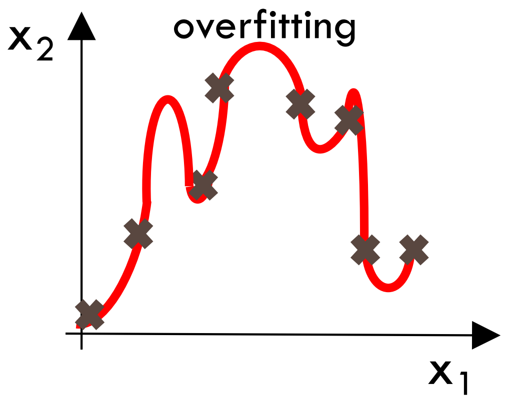


- In well trained systems, typically curves will tend to fit **the overall trend** of the data ($e_{training}$>0)
- If the system is **trained for too long,** it tends to learn the noise and inaccuracies of the data. Models that are more complex than necessary will be learned, generating lower training errors (eventually $e_{training}$=0), but reducing the performance of the system with new data (↑$e_{generalization}$). This situation is called **overfitting.**



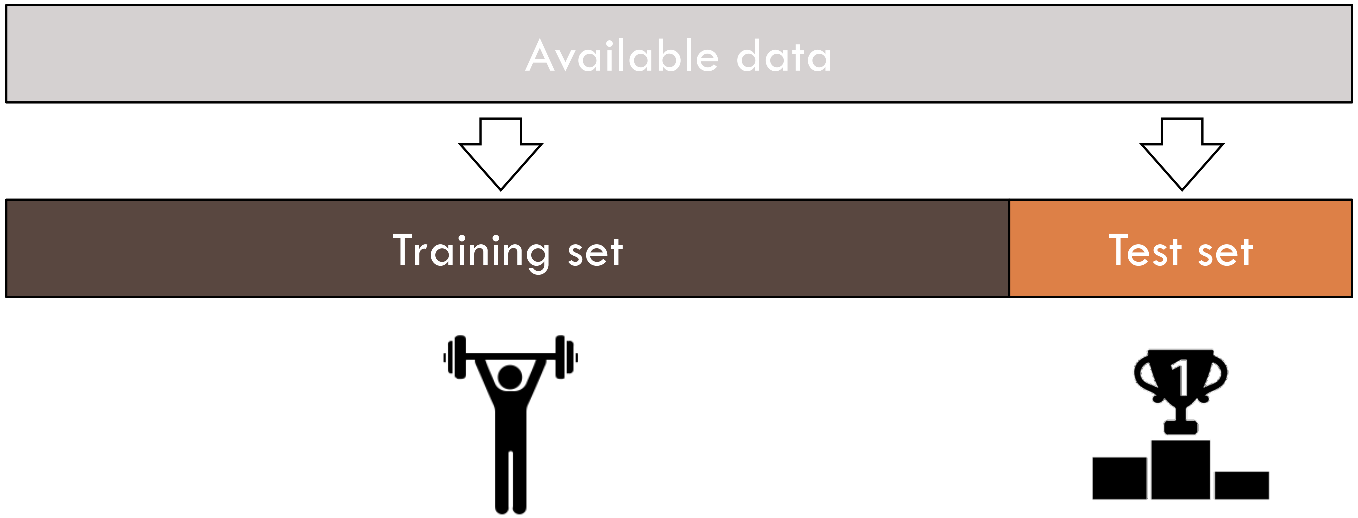
- In well trained systems, typically curves will tend to fit **the overall trend** of the data ($e_{training}>0$)
- If the system is **trained for too long,** it tends to learn the noise and inaccuracies of the data. Models that are more complex than necessary will be learned, generating lower training errors (eventually $e_{training}=0$), but reducing the performance of the system with new data ($\uparrow e_{generalization}$). This situation is called **overfitting.**





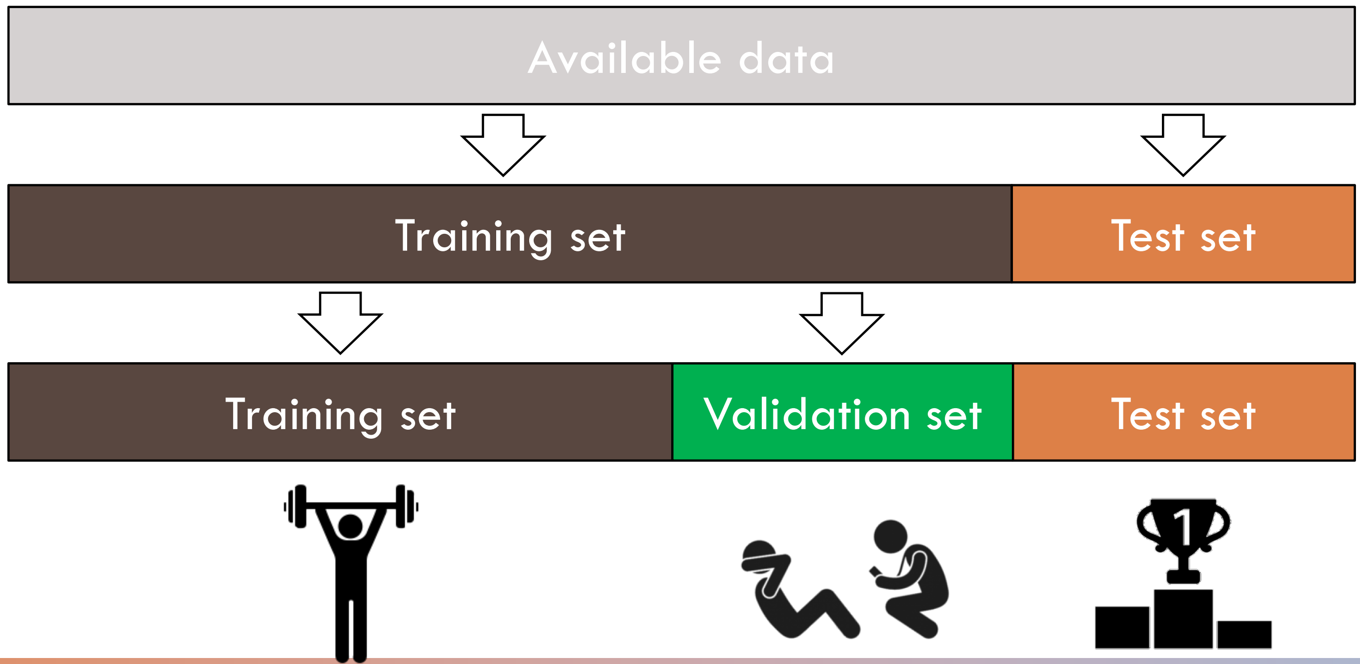
Data splitting

- A common strategy in learning systems is to split the data set in two subsets:
  - **Training set:** data used to train system (adjust the free parameters of the model)
  - **Test set:** data used to evaluate system (learned model). This data should never be used until the end of training!

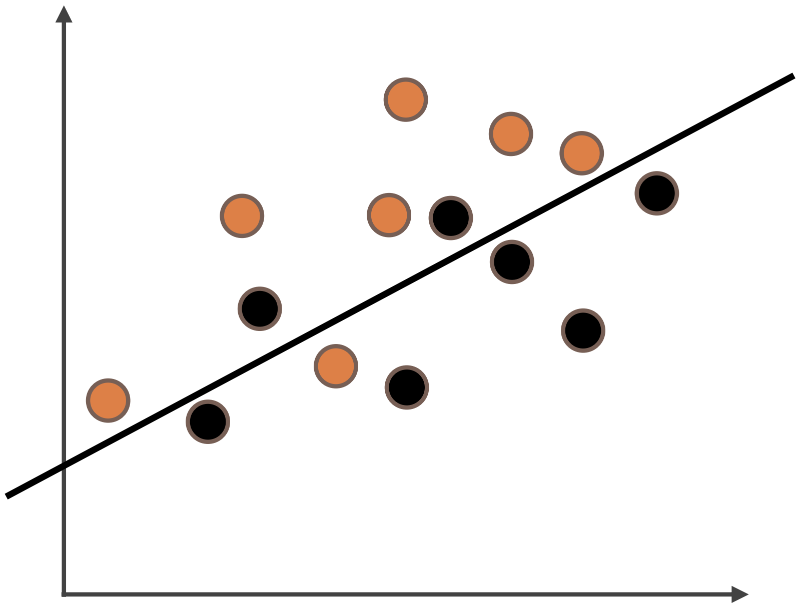


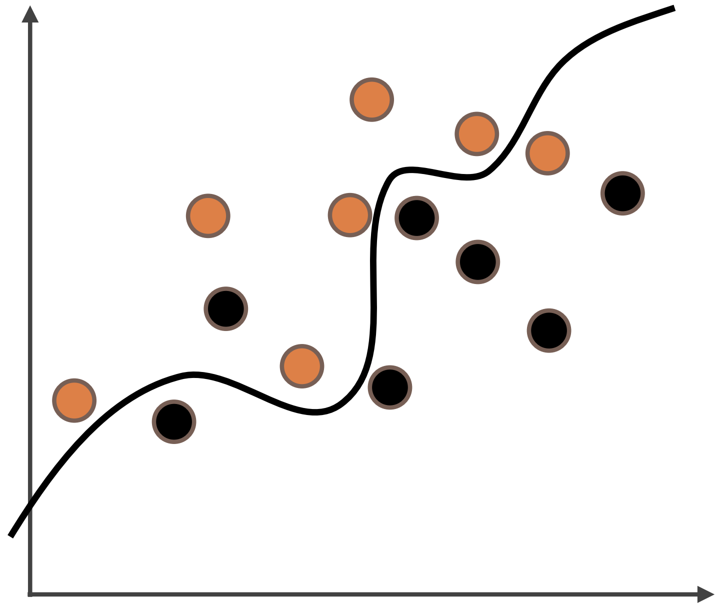
Cross-validation

- Strategy to estimate **how good is the model generalization** in the training phase and eventually to **stop training**
- **Validation set:** data set (not used in training phase) that will be eventually used to evaluate training
- A suitable **proportion** between sets must be defined. Usual practical values are: 50:25:25 or 60:20:20 (if you have plenty of data) until 90:5:5 (few data)

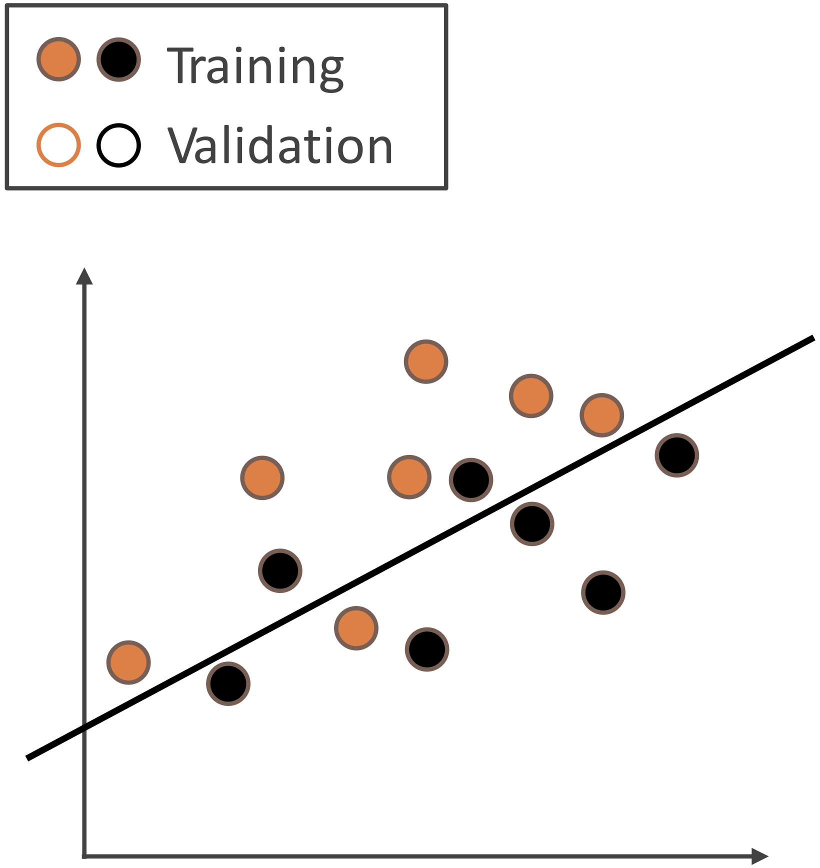


Which model is better?

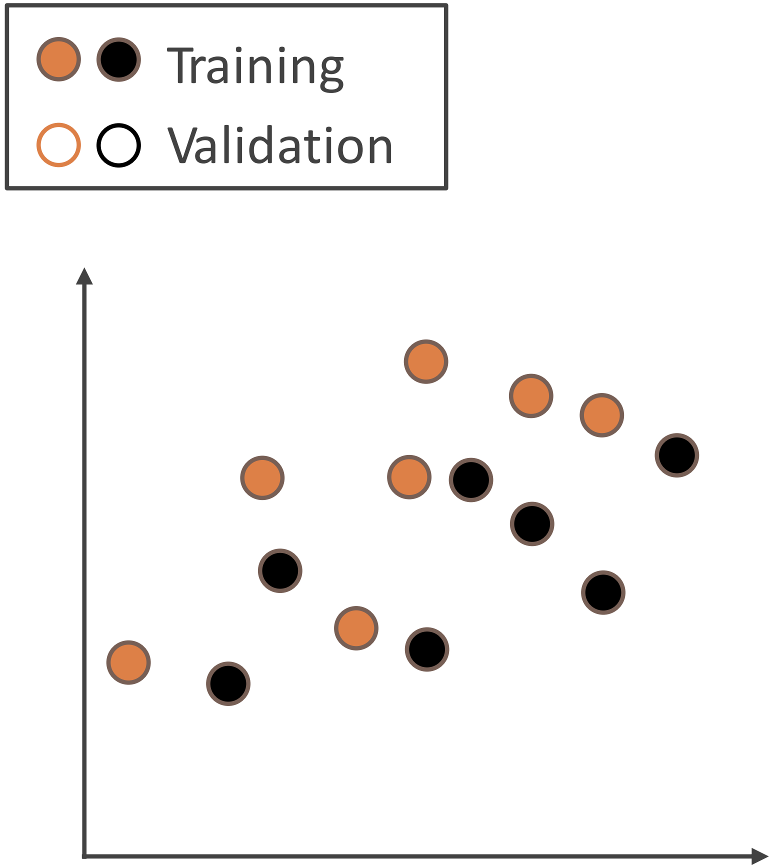




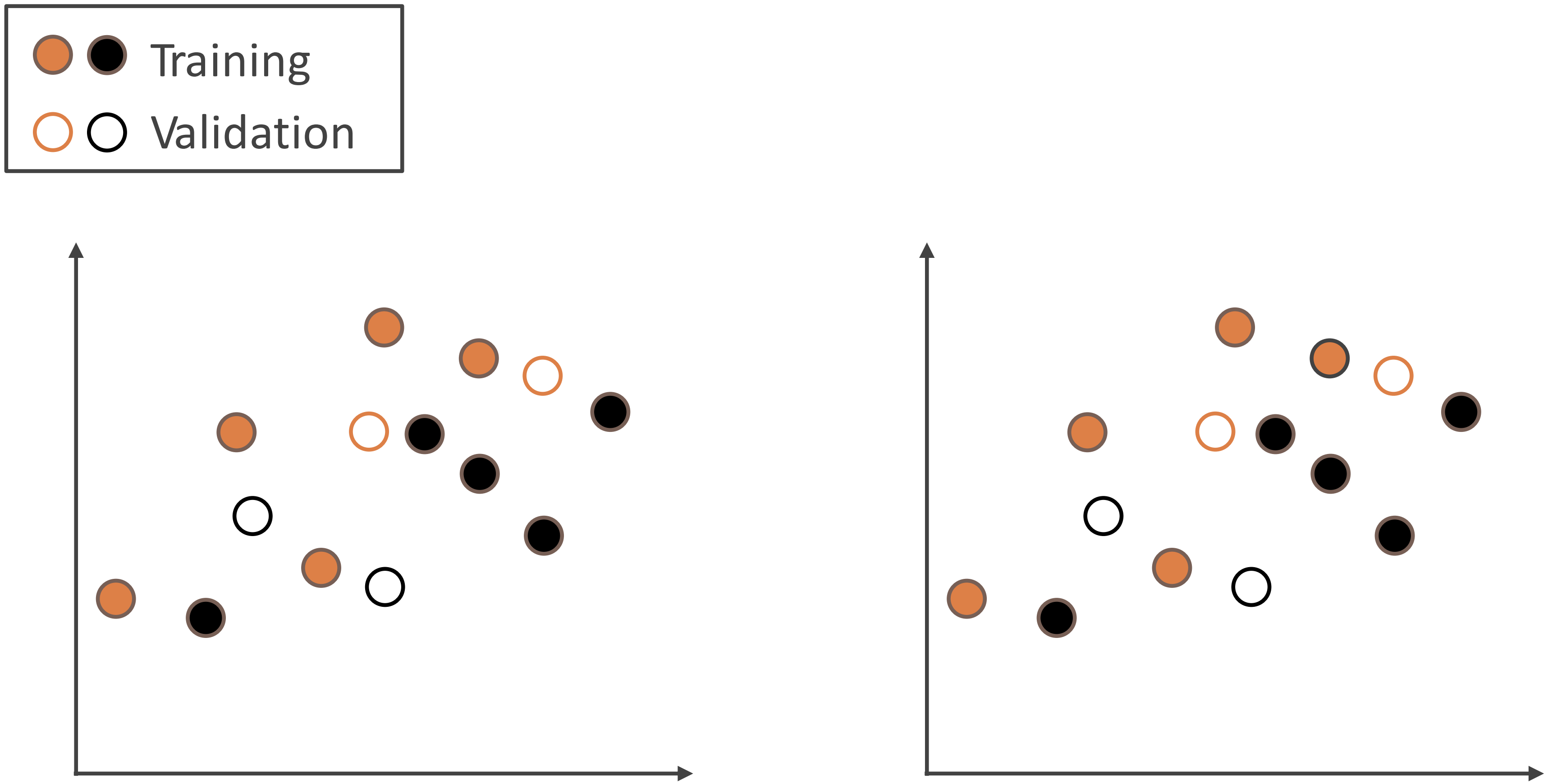
Why validating?



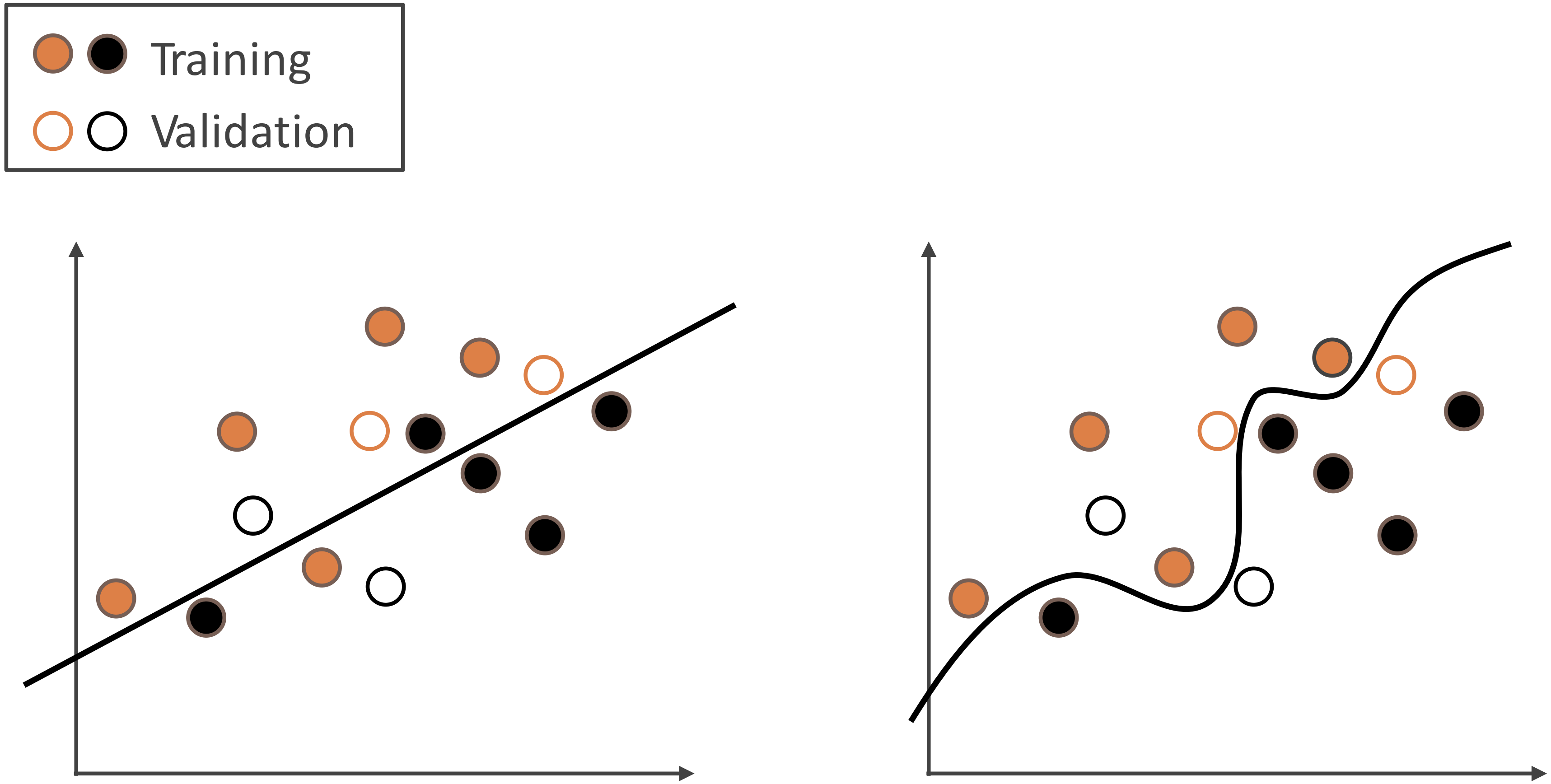
Why validating?



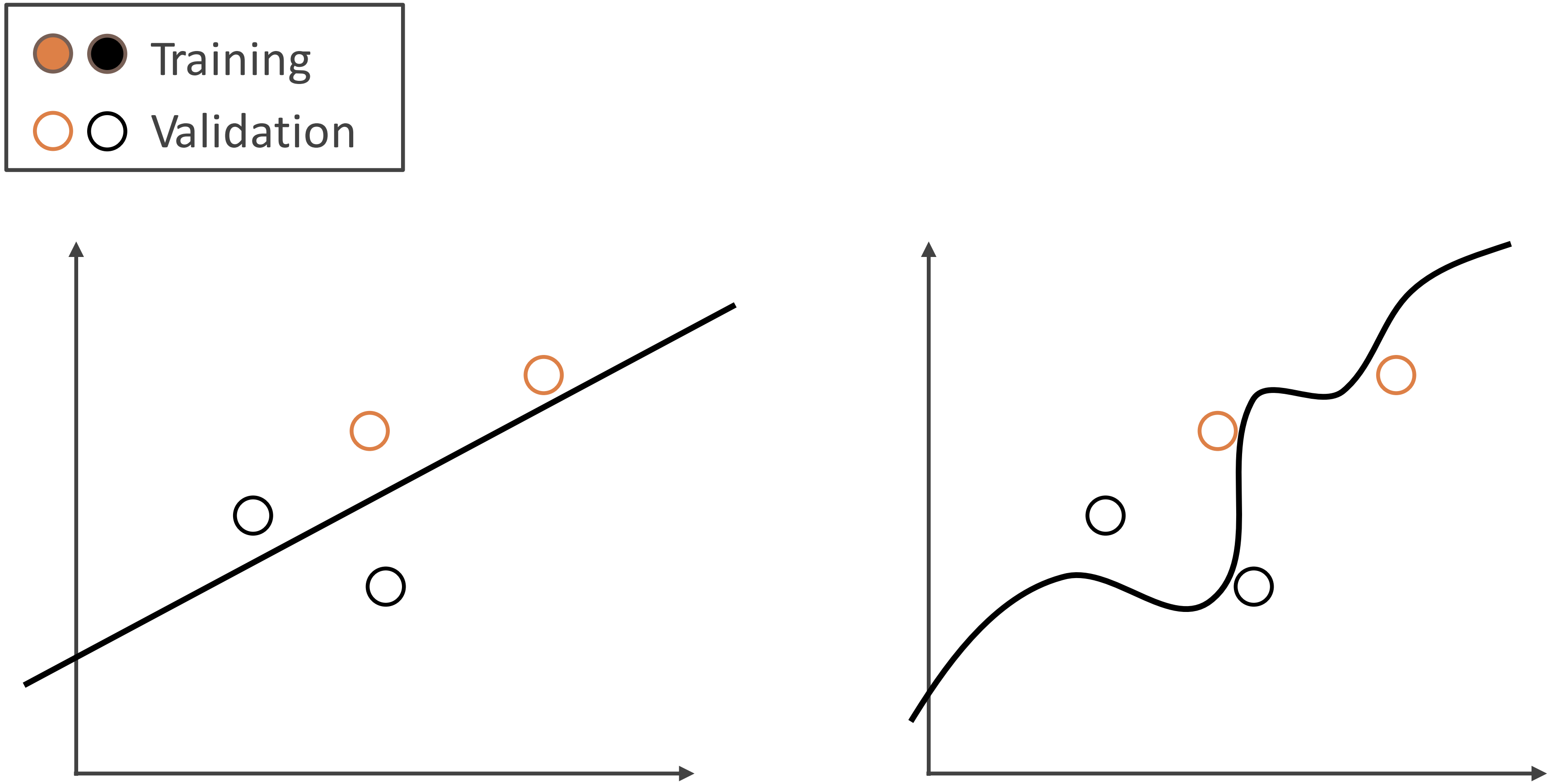
Why validating?



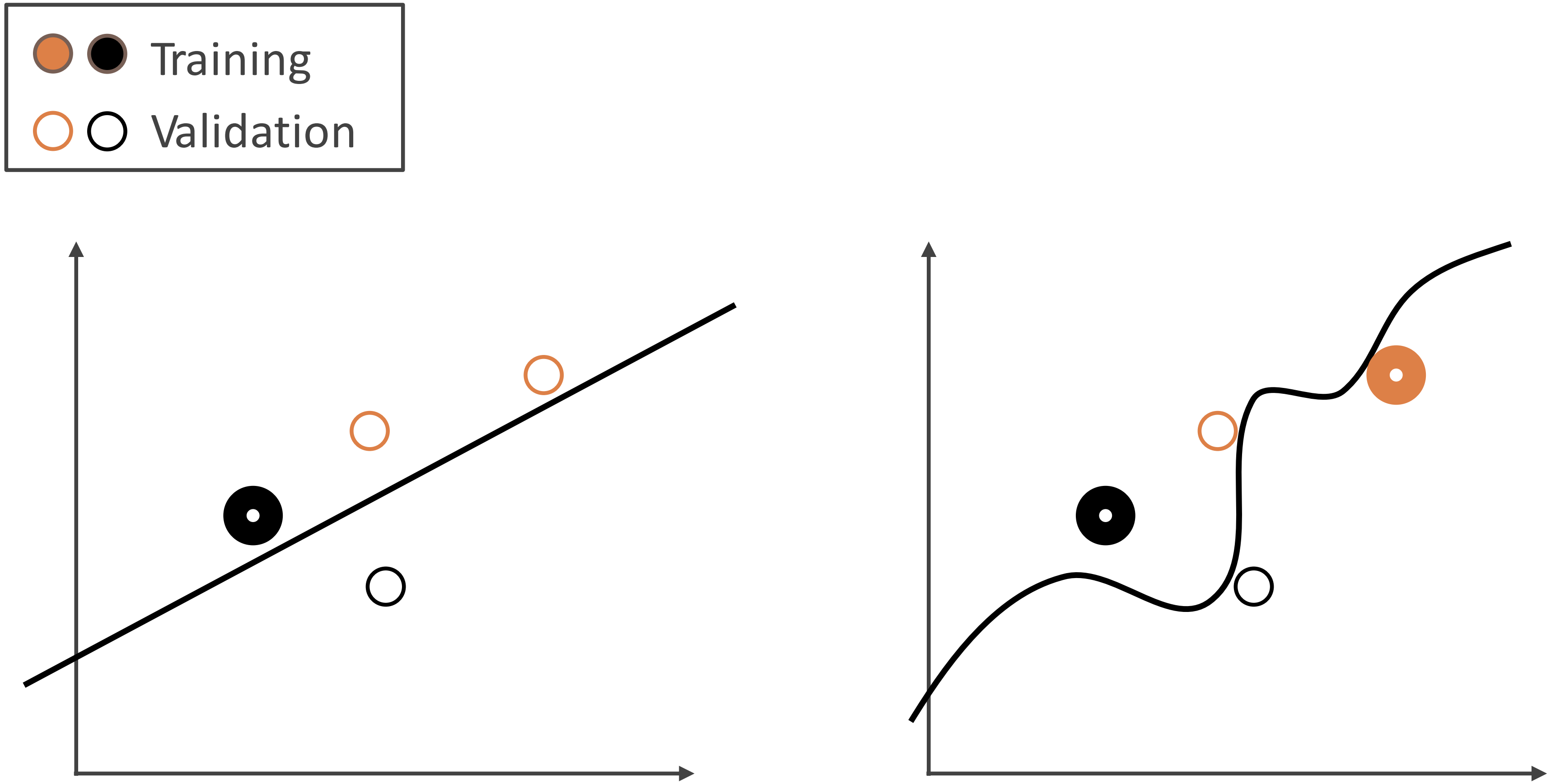
Why validating?



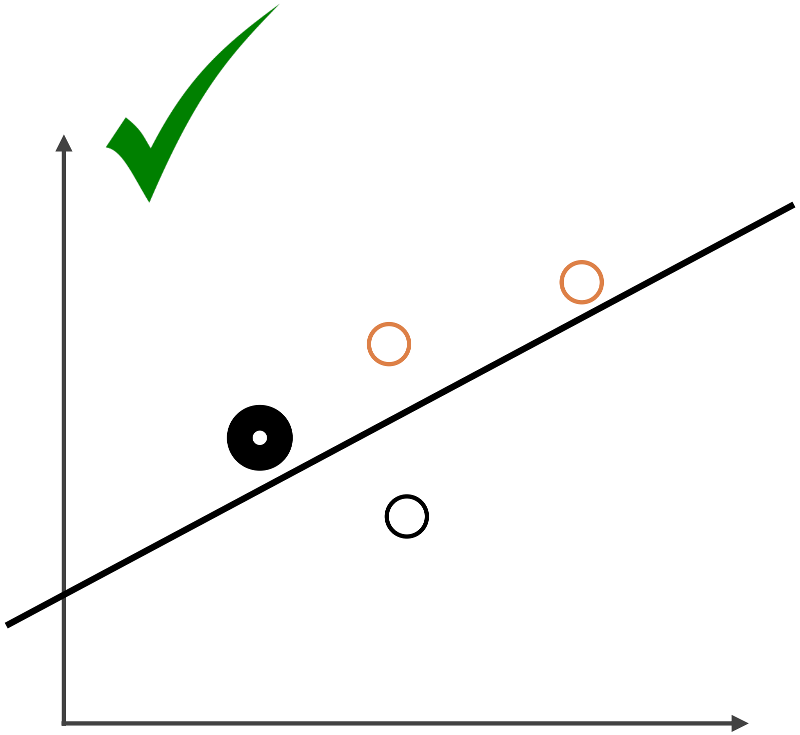
Why validating?

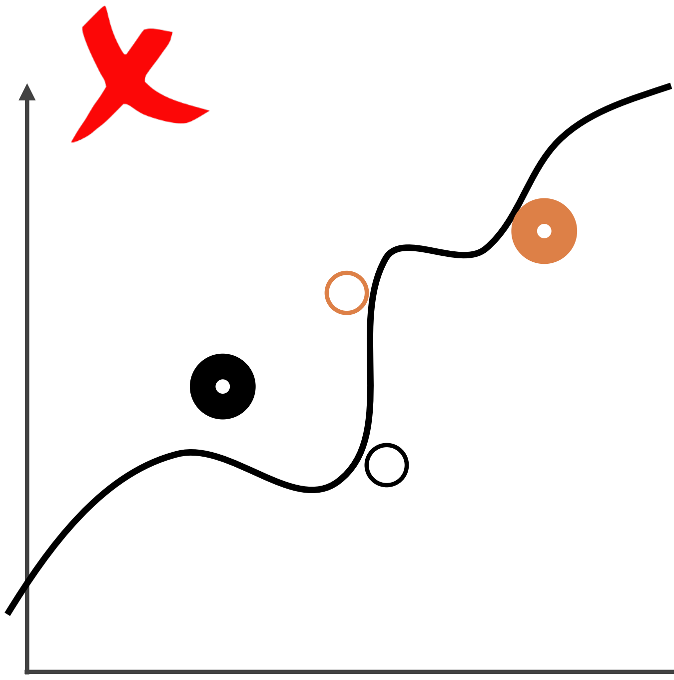


Why validating?



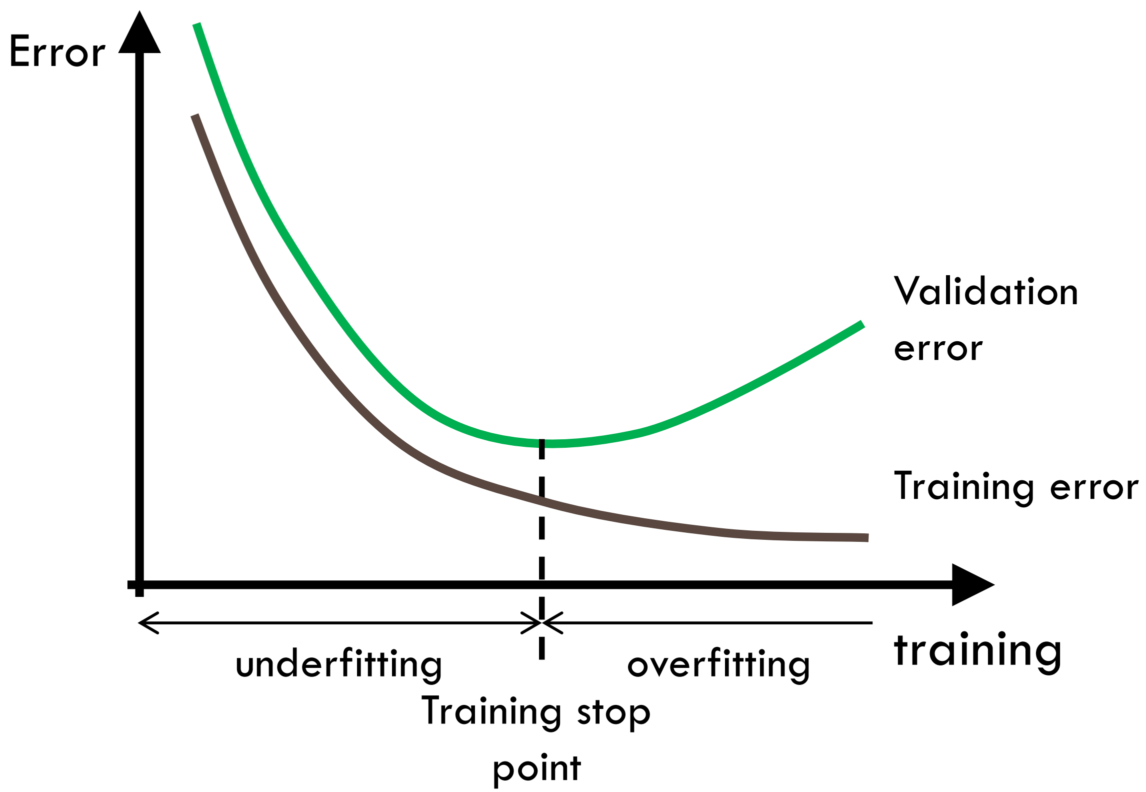
Why validating?





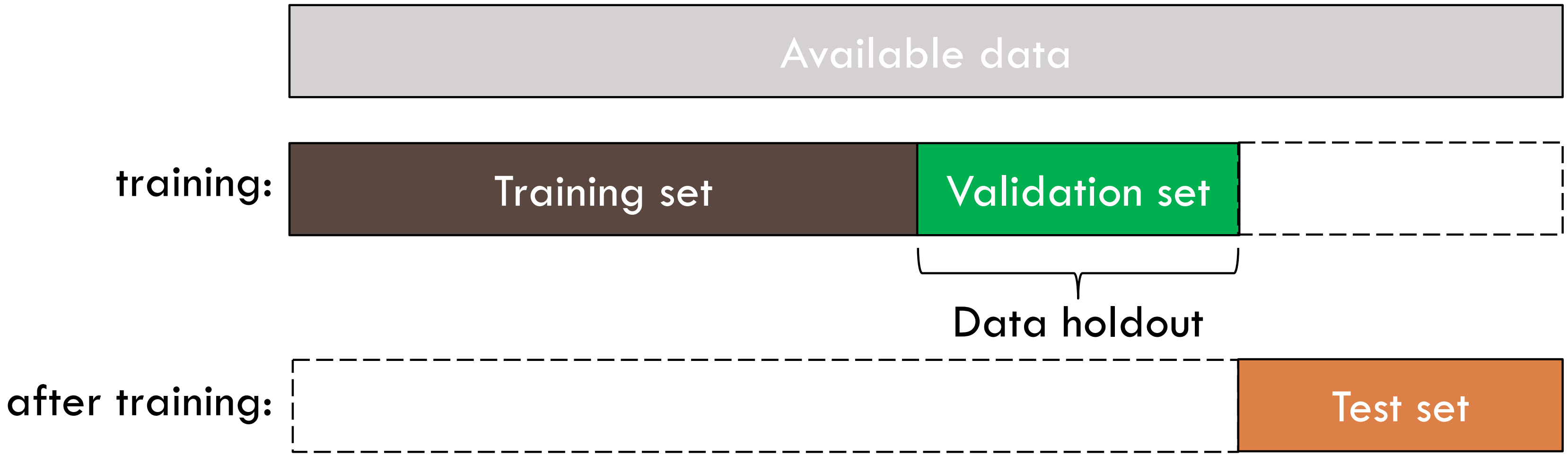
Training early-stopping

- In training, the early-stopping must occur to avoid *underfitting* and *overfitting*



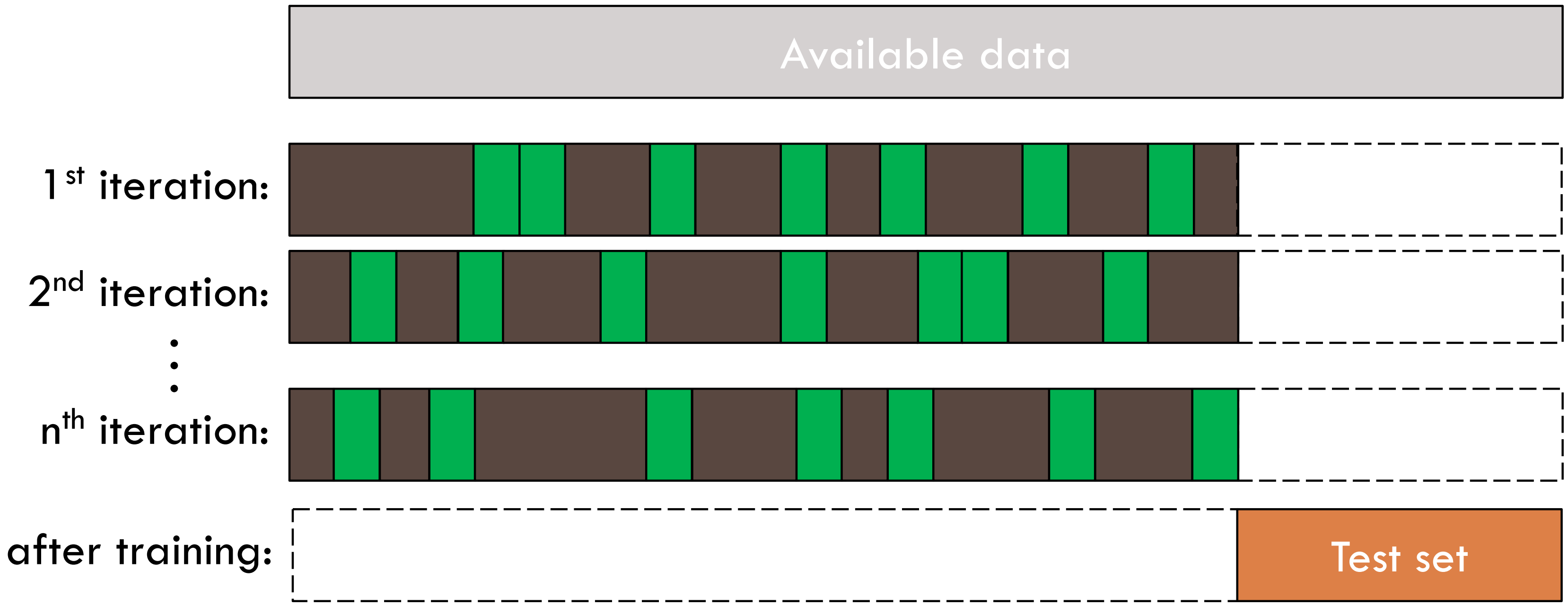
Data splitting strategies

- Holdout method:



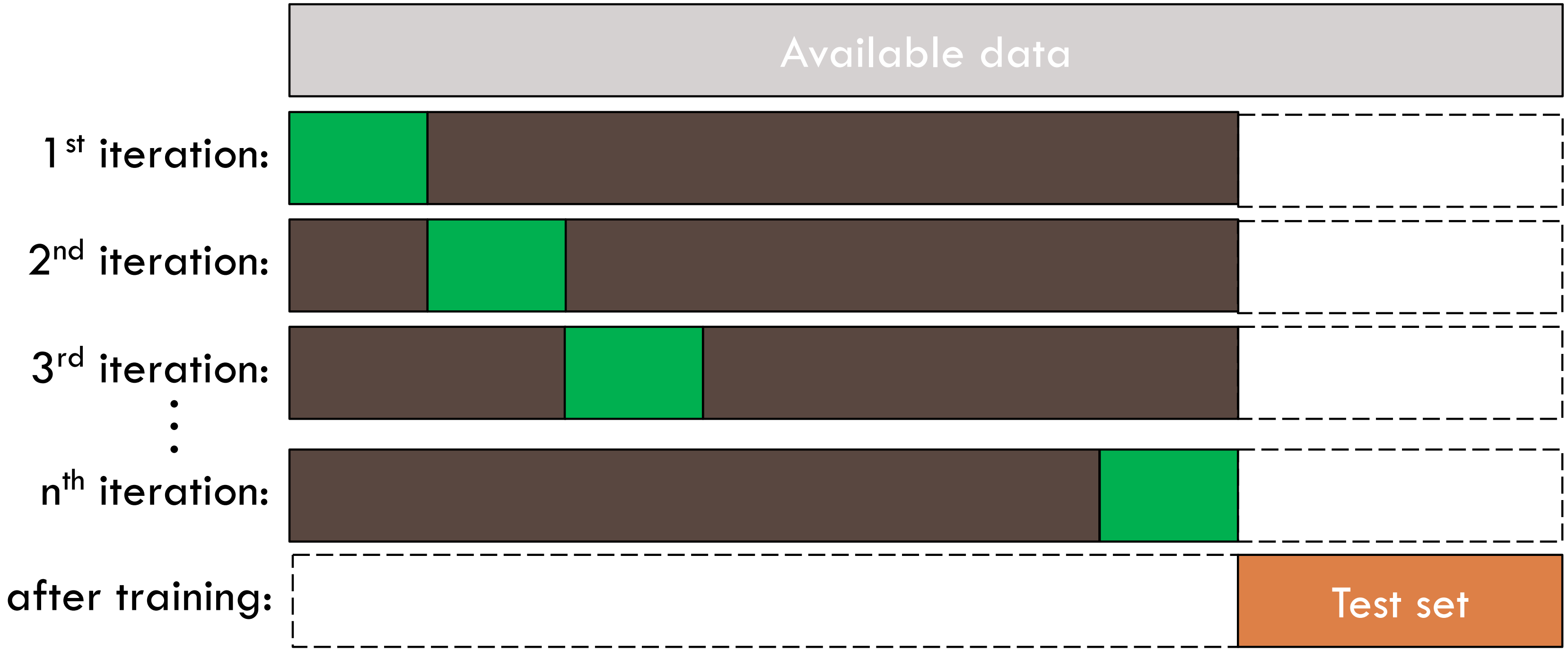
- Remarks:
  - Less data available to training
  - If training and validation sets have different data distribution, results can be distorted

- Repeated holdout method:



- Remark:
  - If some examples are selected more (or less) than others for the training or validation sets, results can be distorted

- K-Fold:

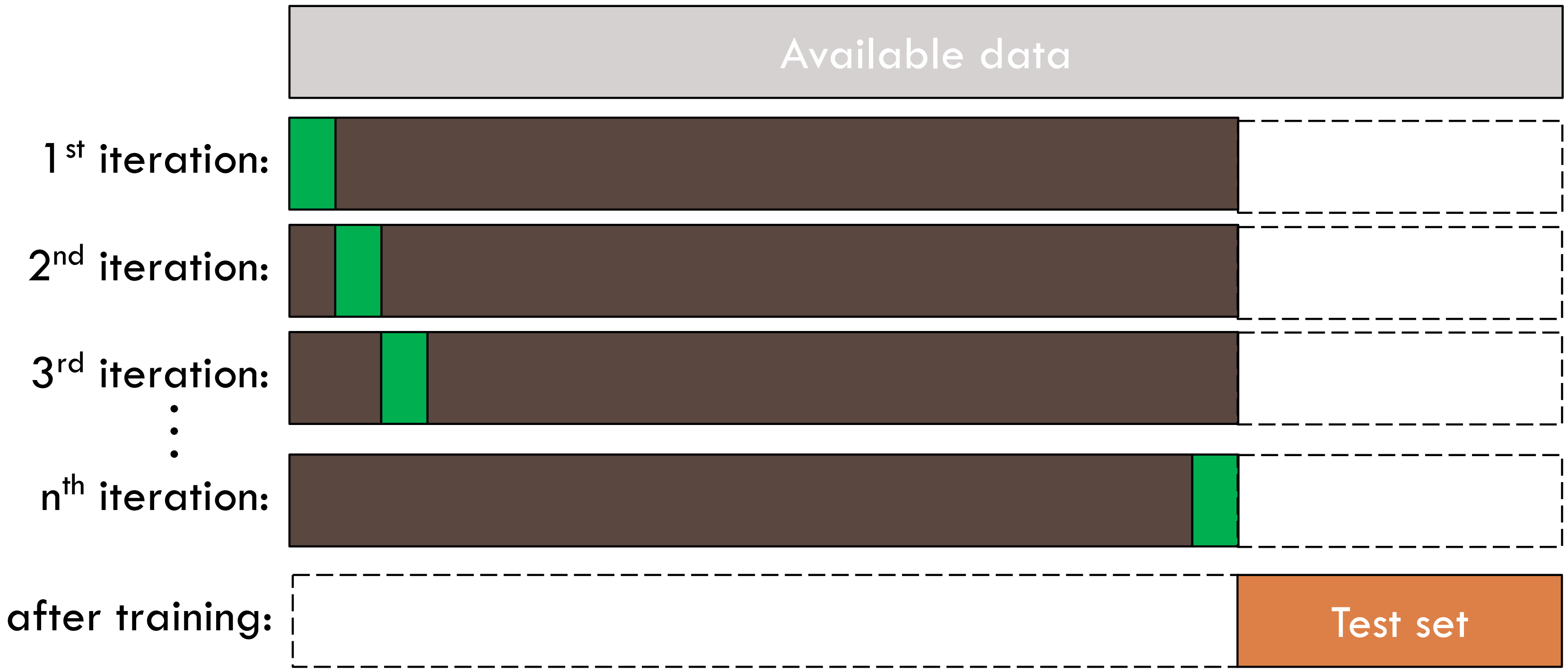


- Remarks:
  - This strategy has a more structured way to divide the available data between training and validation sets
  - Very common approach in practice

$$CV_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} \text{Error}_k$$

Data splitting strategies

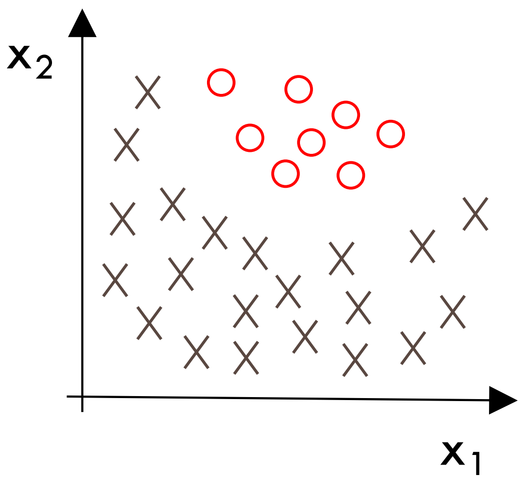
- Leave-one-out:

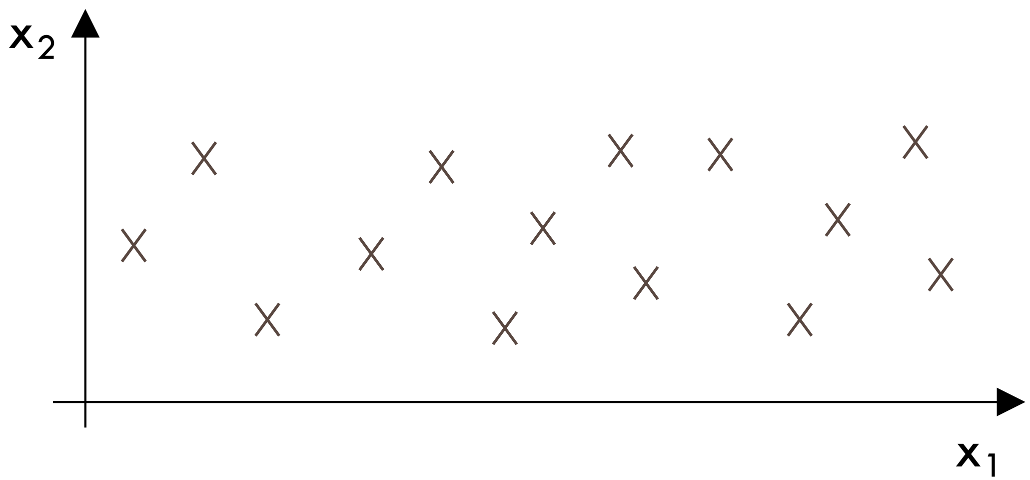


- Remarks:
  - Particular case of K-Fold with just one sample at the validation set

Input data distribution

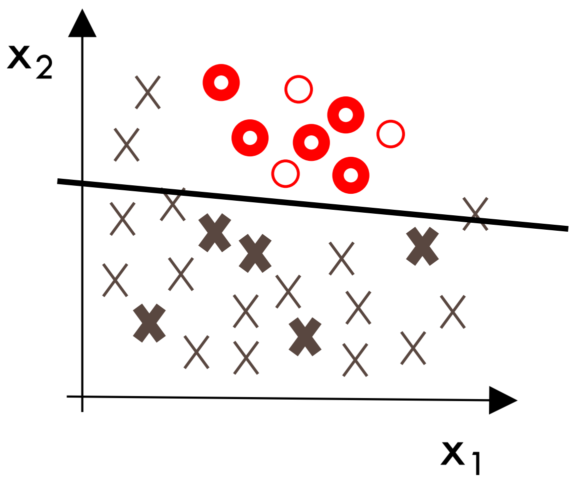
- Ideally, sampling in all datasets (training, validation and test) must have the **same data distribution than the total available data.** Particularly:
  1. Sets must preserve the same relation of the number of samples from each class present in original data
  2. Sets must contain samples spread over all regions of the domain

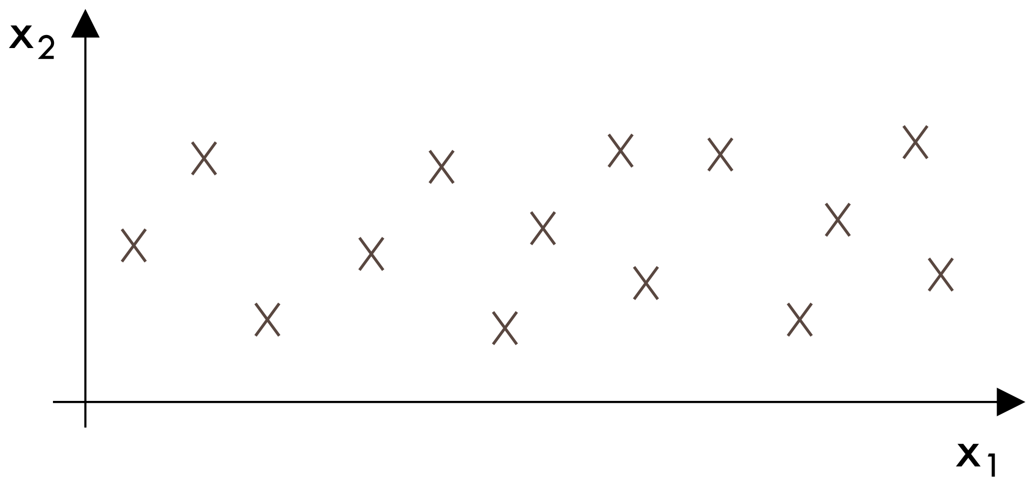




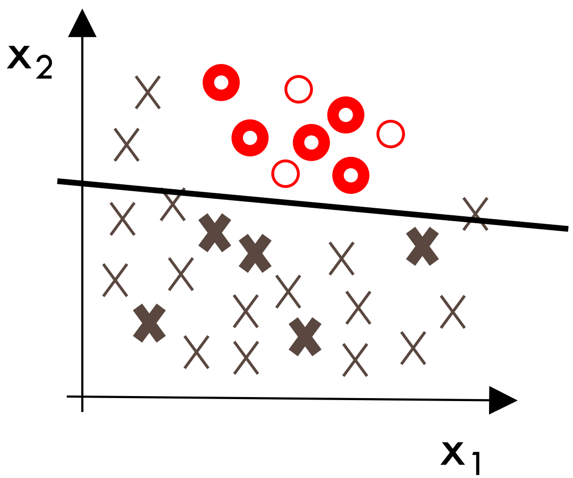
Input data distribution

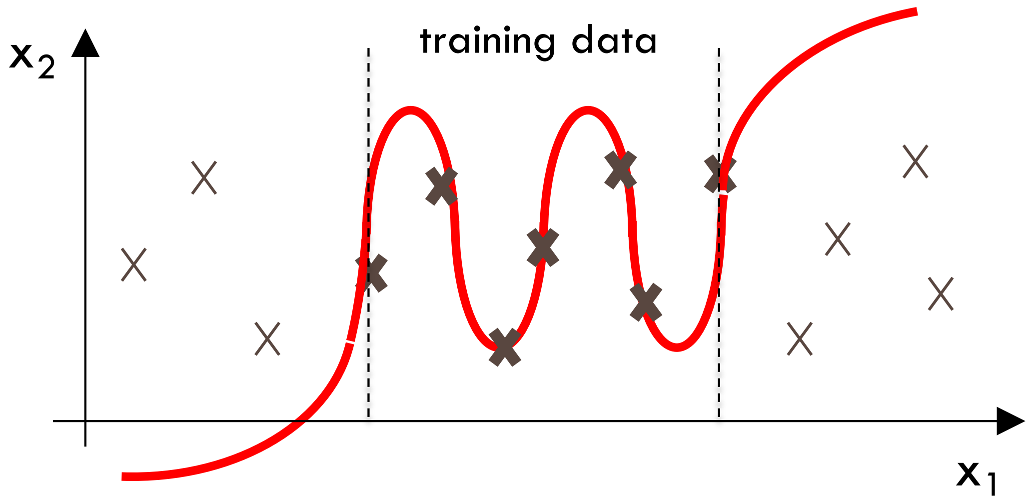
- Ideally, sampling in all datasets (training, validation and test) must have the **same data distribution than the total available data.** Particularly:
  1. Sets must preserve the same relation of the number of samples from each class present in original data
  2. Sets must contain samples spread over all regions of the domain





- Ideally, sampling in all datasets (training, validation and test) must have the **same data distribution than the available data.** Particularly:
  1. Sets must preserve the same relation of the number of samples from each class present in original data
  2. Sets must contain samples spread over all regions of the domain

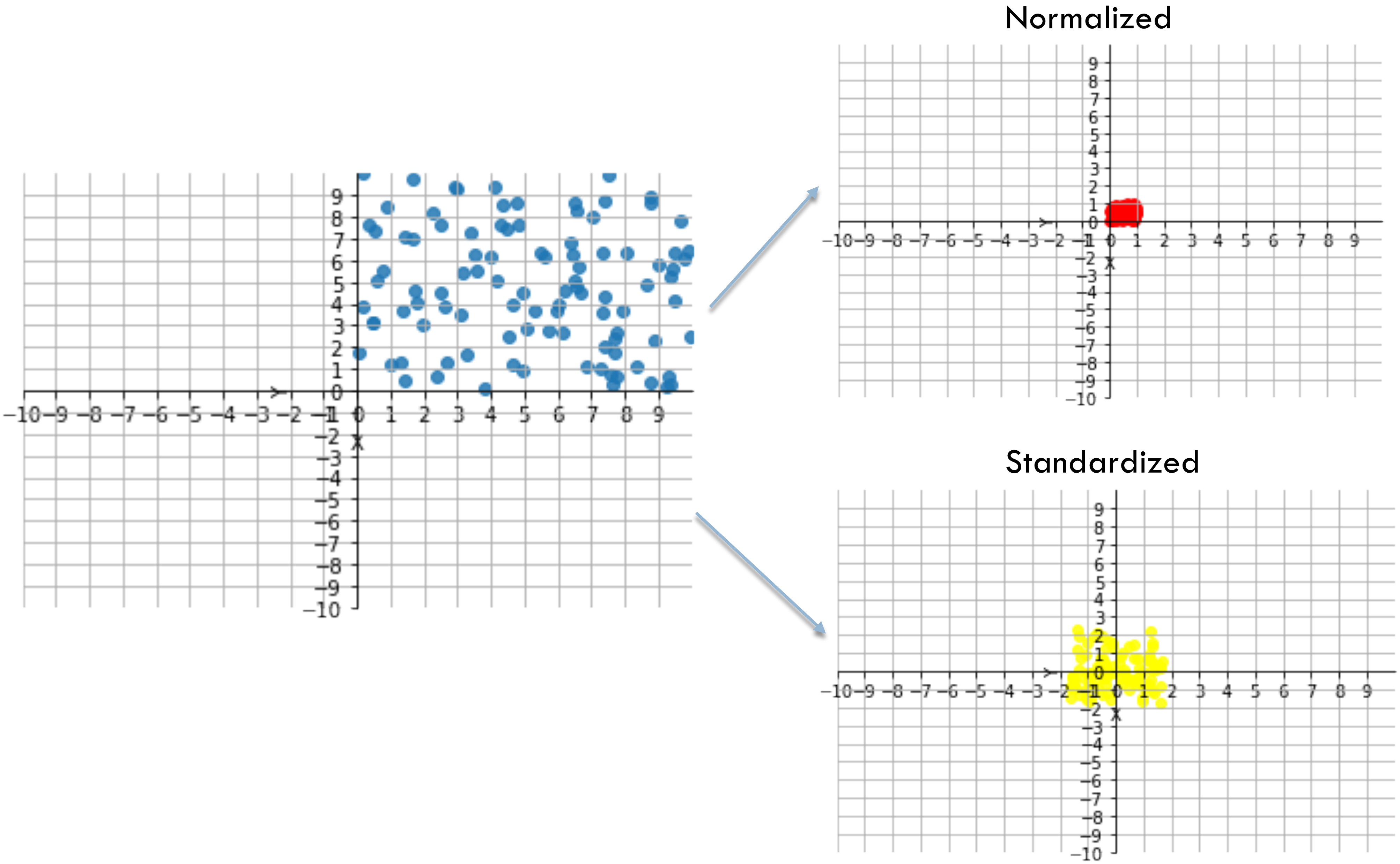




Feature scaling

- **Feature scaling** is a method used to normalize the range of independent variables or features of data
  - It is also known as data **normalization** and is generally performed during the data preprocessing step
  - In **normalization**, the range of your data changes. Normalization means adjusting values measured on different scales to a common scale
    - typically, between [0,1] or [-1,1]
  - In **standardization** the shape of the distribution of your data changes
    - typically means rescaling data to have a mean of 0 and a standard deviation of 1 (unit variance)
- Why use these pre-processing steps?
  - the range of values of raw data varies widely
  - some machine learning algorithms will not work properly without normalization
  - For example, methods that calculate the distance between two points using the Euclidean distance
    - If one of the features has a broad range of values, the distance will be governed by this particular feature

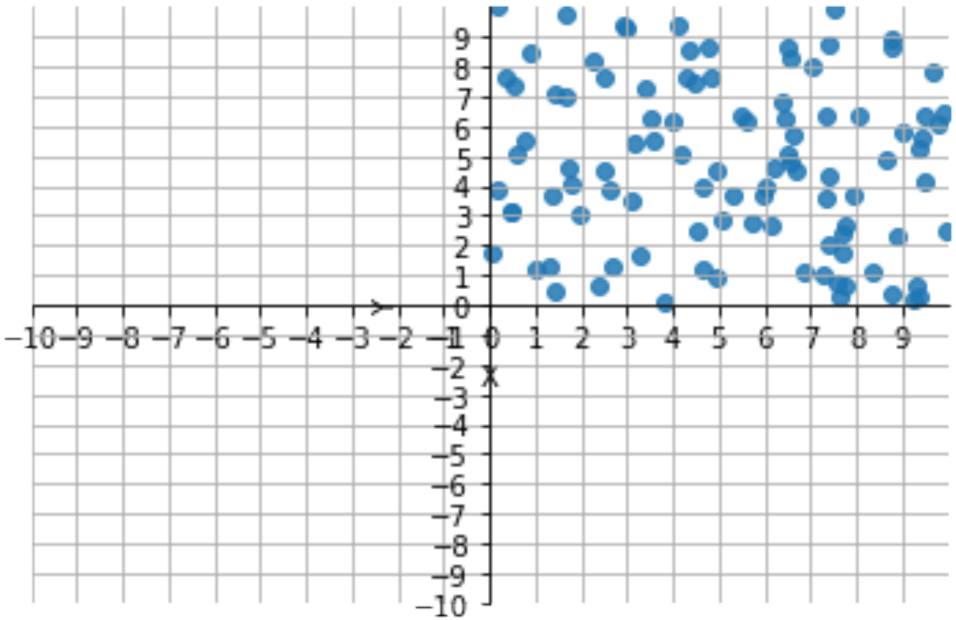
Feature scaling



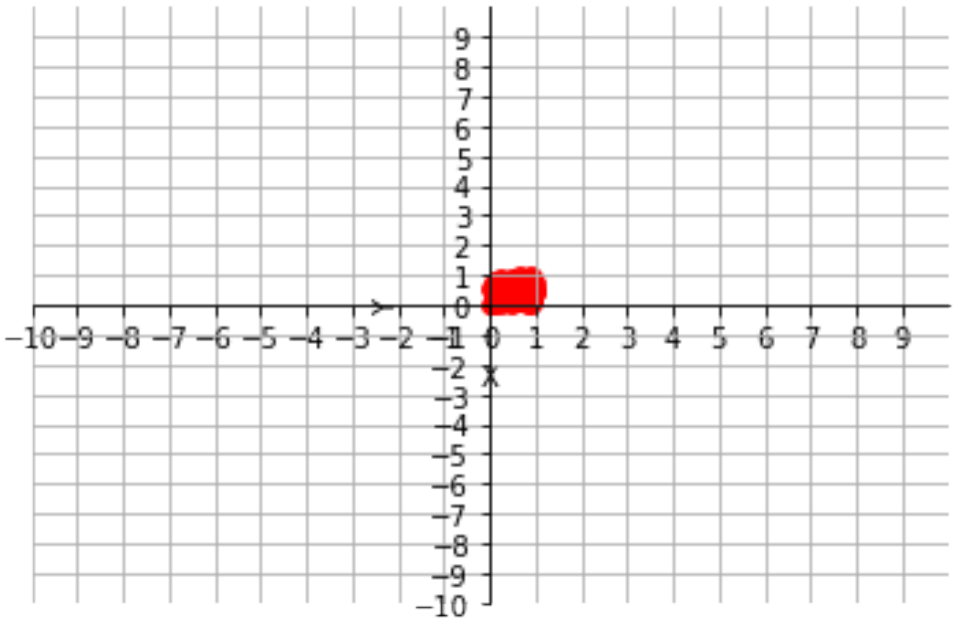
Feature scaling

Types

- Min-Max Scaler (MinMaxScaler)
  - Transform features by scaling each feature to a given range



mean: [4.64091733 4.71477069]
std: [2.92323848 2.93324495]
min: [0.02917744 0.07193049]
max: [9.97337705 9.99855761]

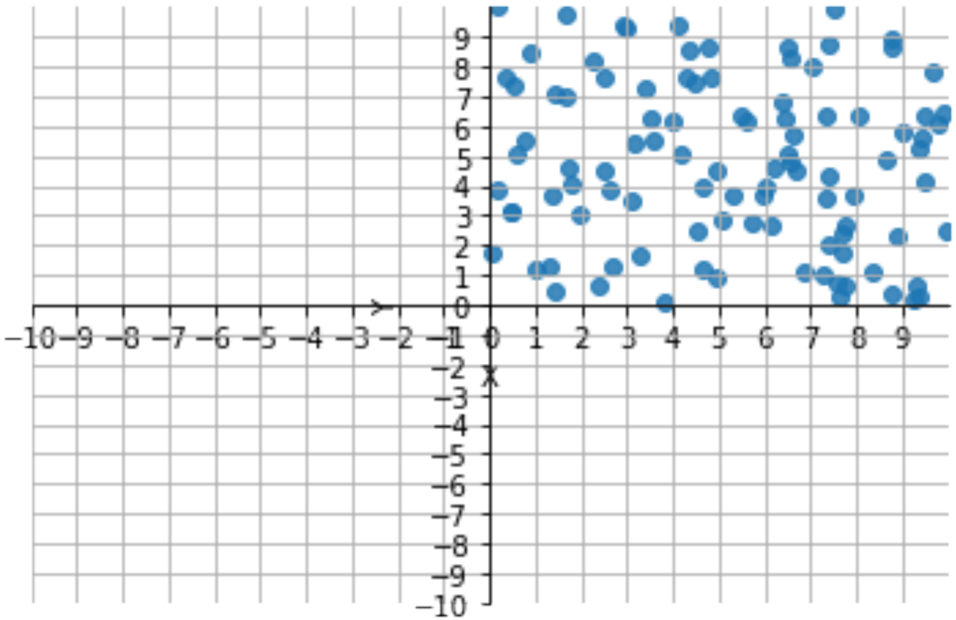


mean: [0.52228216 0.50308238]
std: [0.30594303 0.26681954]
min: [0. 0.]
max: [1. 1.]

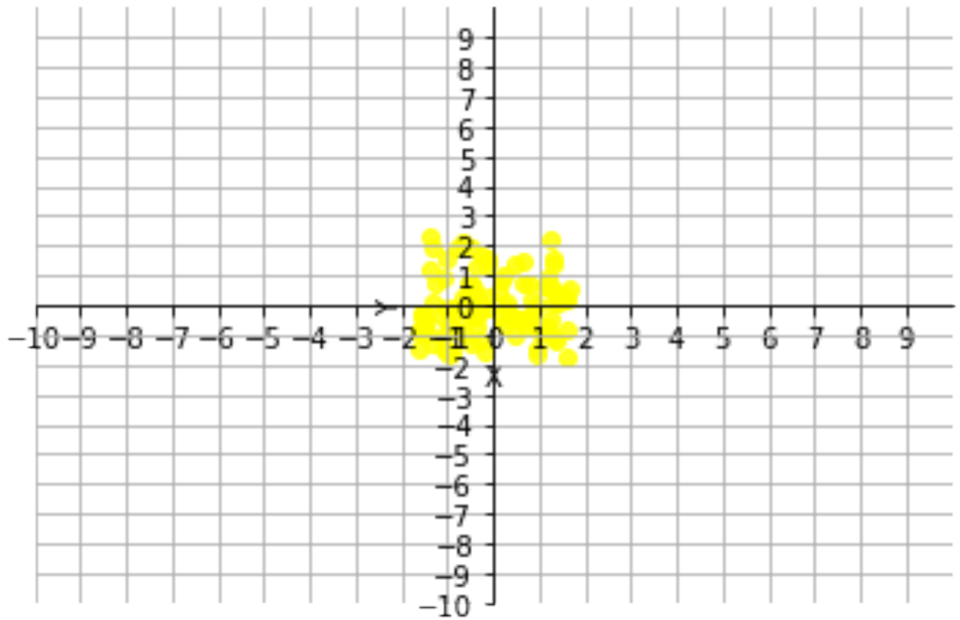
Feature scaling

Types

- Standard Scaler (StandardScaler)
  - Standardize features by removing the mean and scaling to unit variance

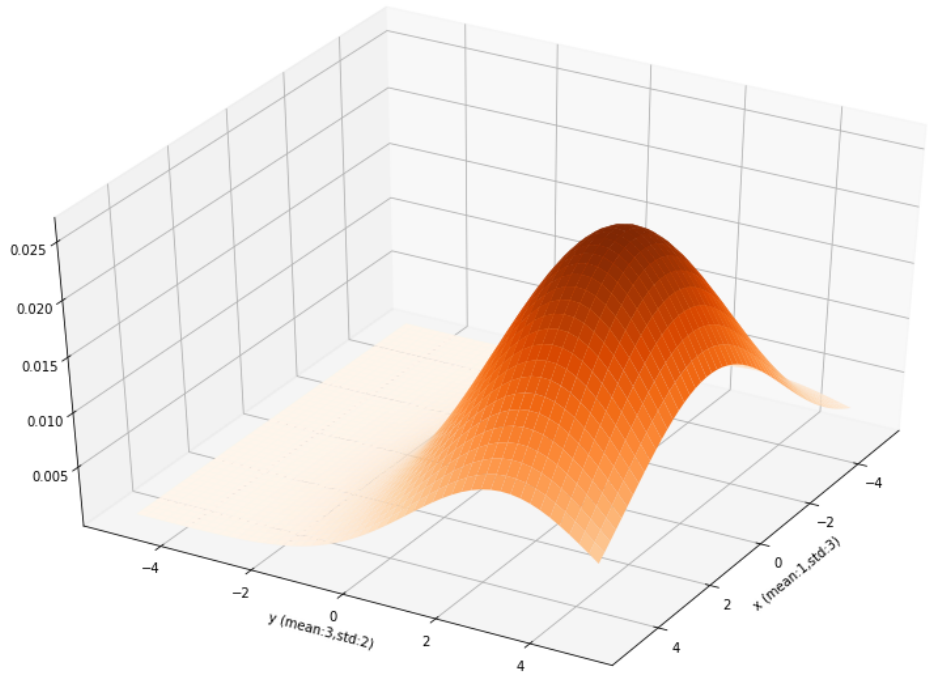


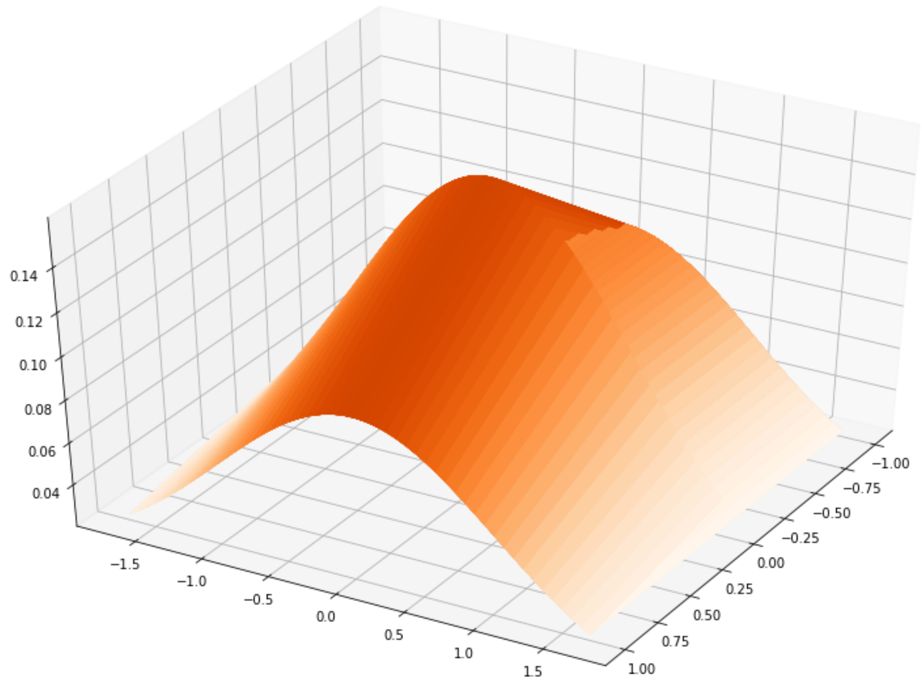
mean: [4.64091733 4.71477069]
std: [2.92323848 2.93324495]
min: [0.02917744 0.07193049]
max: [9.97337705 9.99855761]



mean: [0.52228216 0.50308238]
std: [0.30594303 0.26681954]
min: [0. 0.]
max: [1. 1.]

- Standard Scaler (StandardScaler) – 3d
  - Standardize features by removing the mean and scaling to unit variance

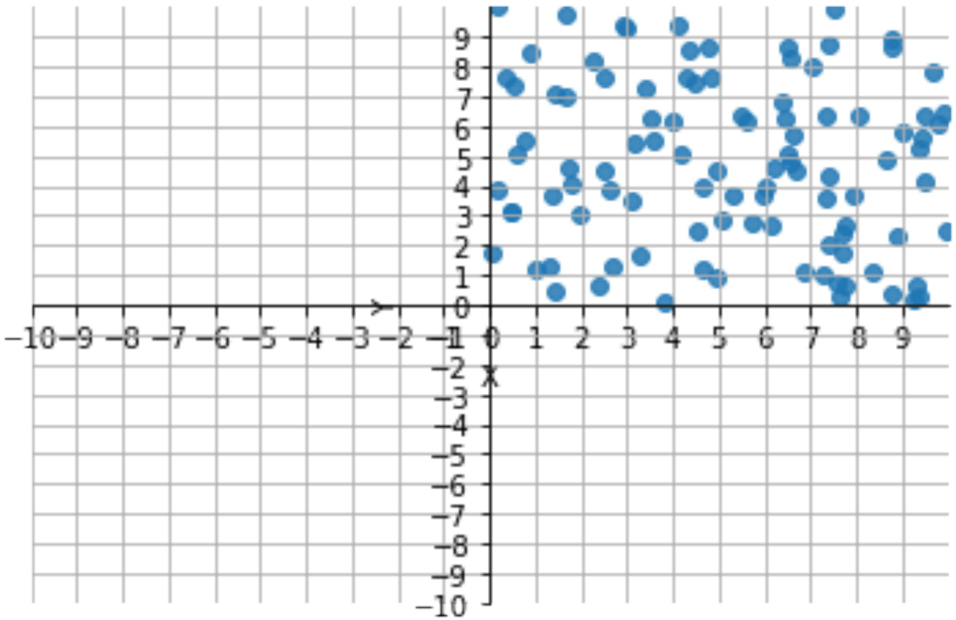




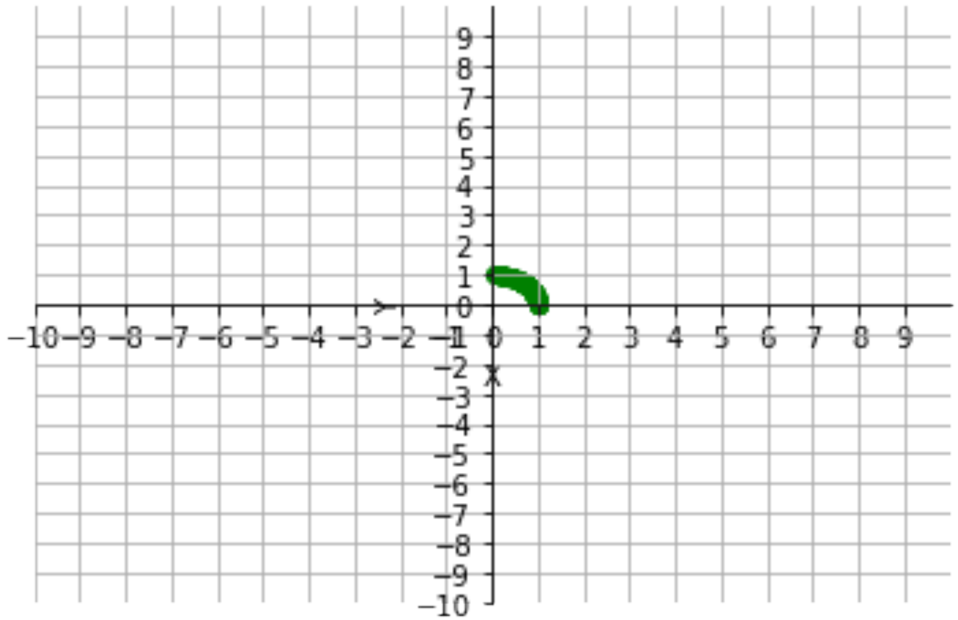
Feature scaling

Types

- Unit Vector Scaler (Normalizer)
  - Normalize samples individually to unit norm.



mean: [5.24874418 4.65066746]
std: [3.03272127 3.02536061]
min: [0.01605648 0.07731266]
max: [9.93963622 9.76344994]



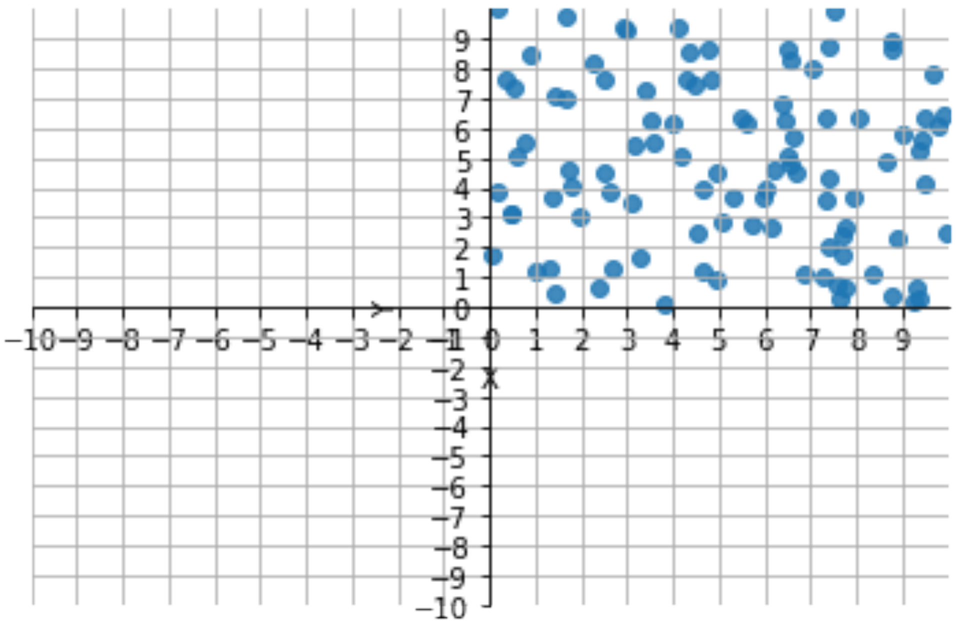
mean: [0.67938241 0.59354895]
std: [0.30042037 0.3096559]
min: [0.00180982 0.05084794]
max: [0.99870641 0.99999836]

Feature scaling

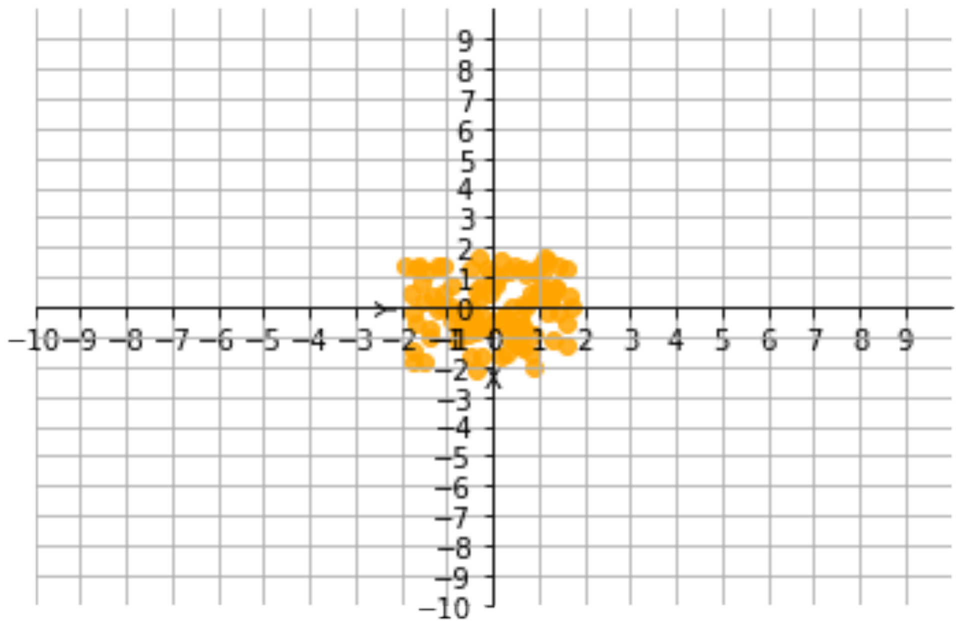
Types

Power Transformer Scaler (PowerTransformer)

- Apply a power transform featurewise to make data more Gaussian-like.



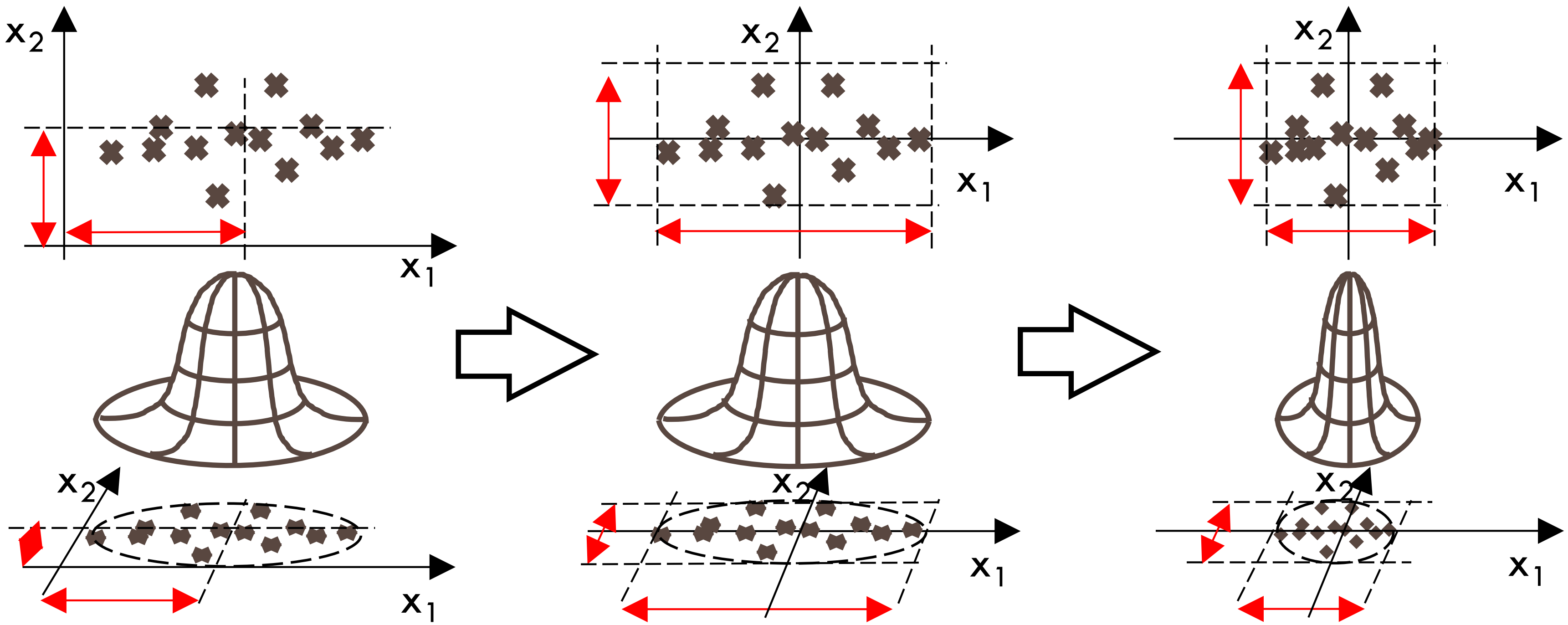
mean: [5.00643671 5.14637406]
std: [2.7285058 2.73460667]
min: [0.07036322 0.0115435]
max: [9.99080893 9.87448843]



mean: [-2.44249065e-17 -7.73617281e-16] std: [1. 1.]
min: [-1.92383512 -2.07521871]
max: [1.74245717 1.63357549]

Input data normalization

- Adjust of measured values (probability distribution):



Zero mean adjustment:

$$\mu_d = \frac{1}{n}\sum_{i=1}^{n} x_{id}$$

$$x_{id} := x_{id} - \mu_d$$

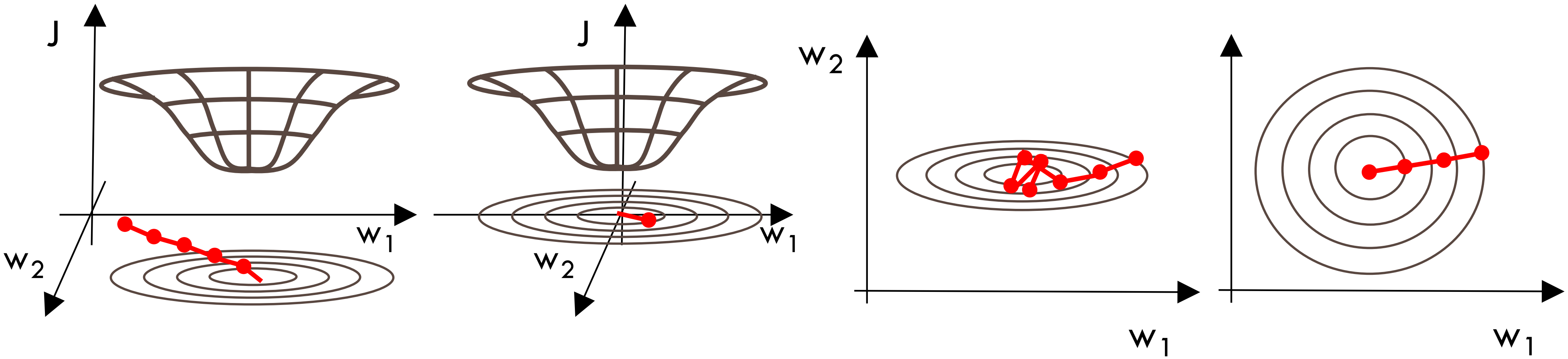
Variance normalization:

$$\sigma_d{}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_{id} - \mu_d)^2$$

$$x_{id} := \frac{x_{id}}{\sigma_d{}^2}$$

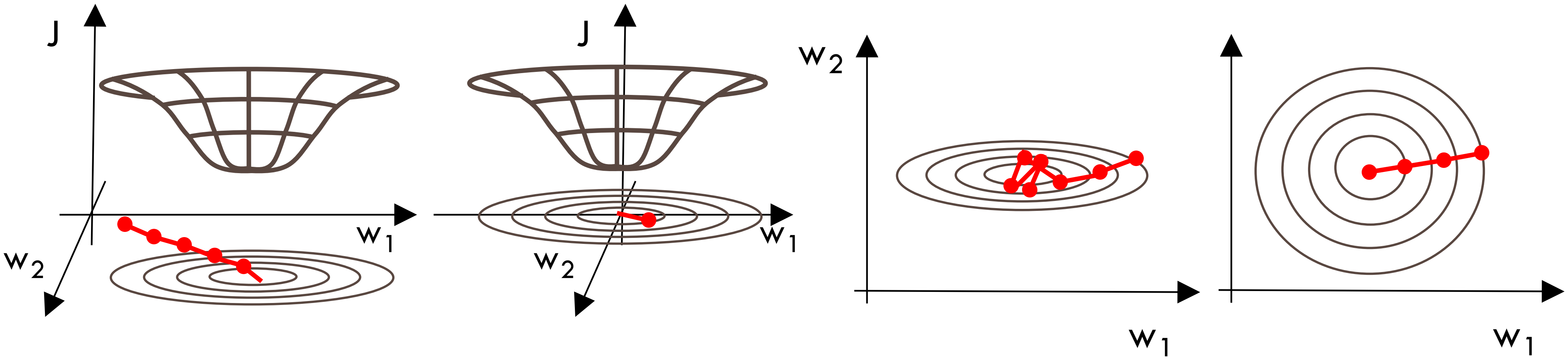
Why normalize input data?

- Input data distribution has implications in the **shape of the cost function** used in some learning processes. Although most of the learning algorithms are able to learn over unnormalized data:
  - If data distribution has **zero mean,** it will probably require less learning steps to tune variables in models
  - In a **symmetrical cost function,** the step size (learning rate) tends to be more effective to find the center of the function



Why normalize input data?

- Input data distribution has implications in the **shape of the cost function** used in some learning processes. Although most of the learning algorithms are able to learn over unnormalized data:
  - If data distribution has **zero mean,** it will probably require less learning steps to tune variables in models
  - In a **symmetrical cost function,** the step size (learning rate) tends to be more effective to find the center of the function



Extra-class activities

Lecture 2

- Activities
  - Quis 2
    - Fundamental of ML
  - Reading:
    - Python for ML

References

Lecture 2

- MARSLAND, S. Machine Learning: an algorithm perspective. CRC Press, 2nd edition, 2015.
- ALPAYDIN, E. Introduction to Machine Learning. MIT Press, 3rd edition, 2014.
- SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. Redes Neurais Artificiais para engenharia e ciências aplicadas. Curso prático. Artliber Editora, 2010.
- SUTTON, R. S.; BARTO, A. G. Reinforcement learning: an introduction. MIT Press, Cambridge, MA, 2nd edition in progress, 2017.

This material is part of the Machine Learning Course
By Esther Colombini and Alexandre Simões

UNICAMP

unesp