Machine Learning

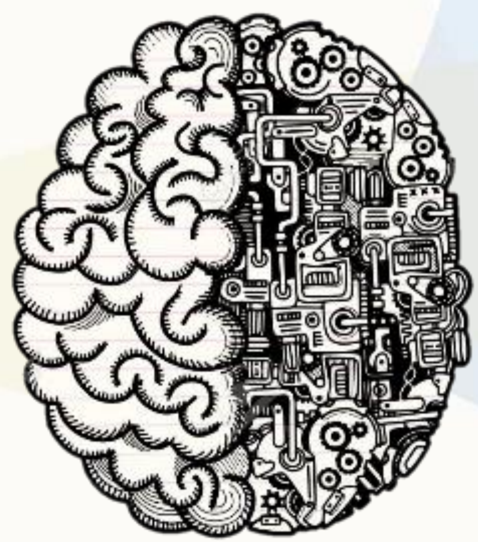
An Introduction

neuro
Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital
McGill
Centre universitaire de santé McGill
McGill University Health Centre

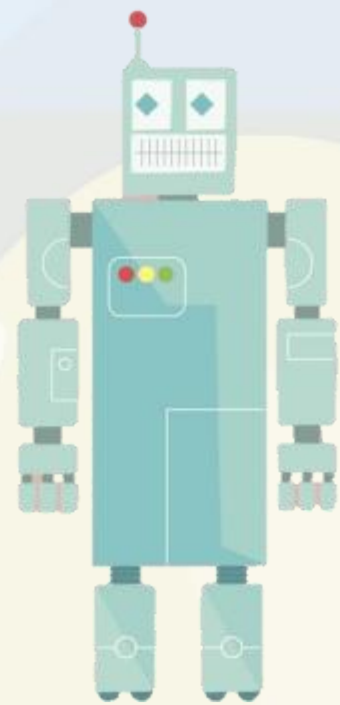
Machine Learning (ML)

Learn from experience

DATA
Learn from ~~experience~~



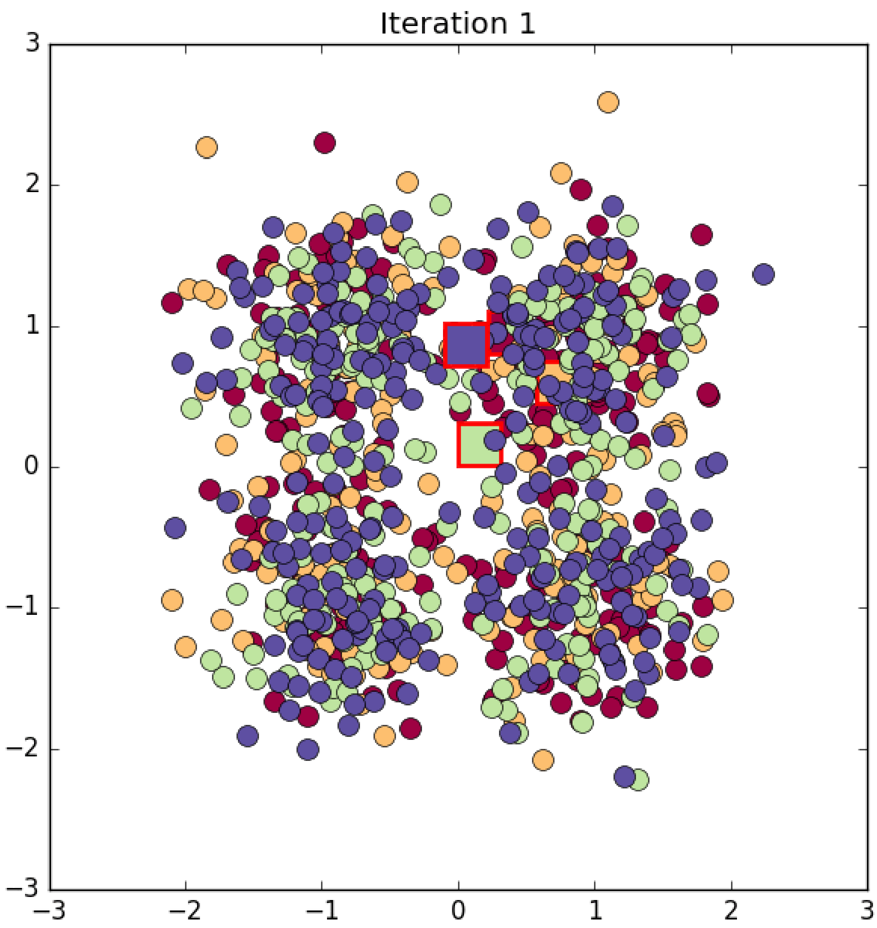
Follow instructions



neuro

McGill

Machine Learning (ML)



ML Algorithms

Automatically find patterns in data - prediction

Supervised Learning

$$x, y$$

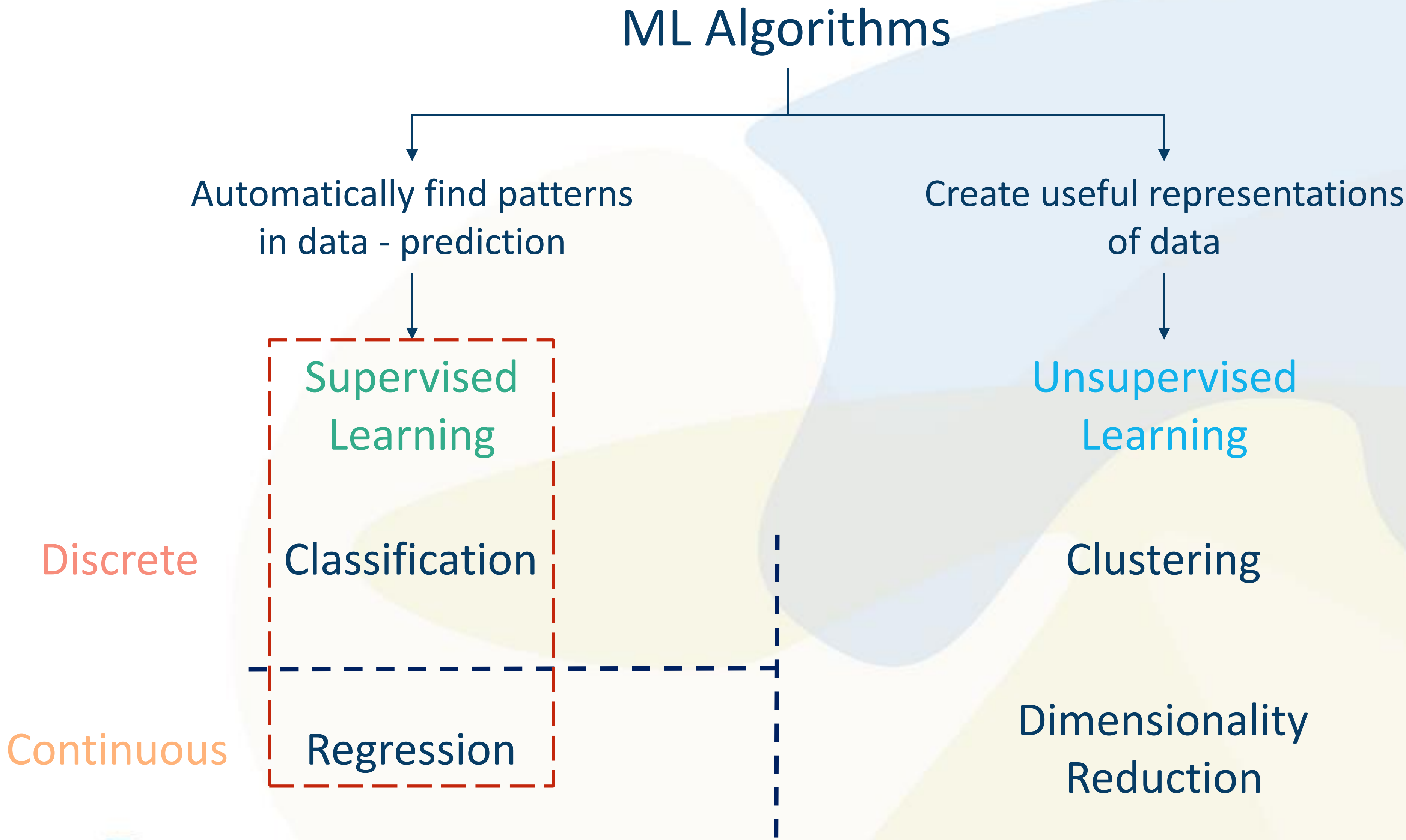
$$f: x \rightarrow y$$

$$y = f(x)$$

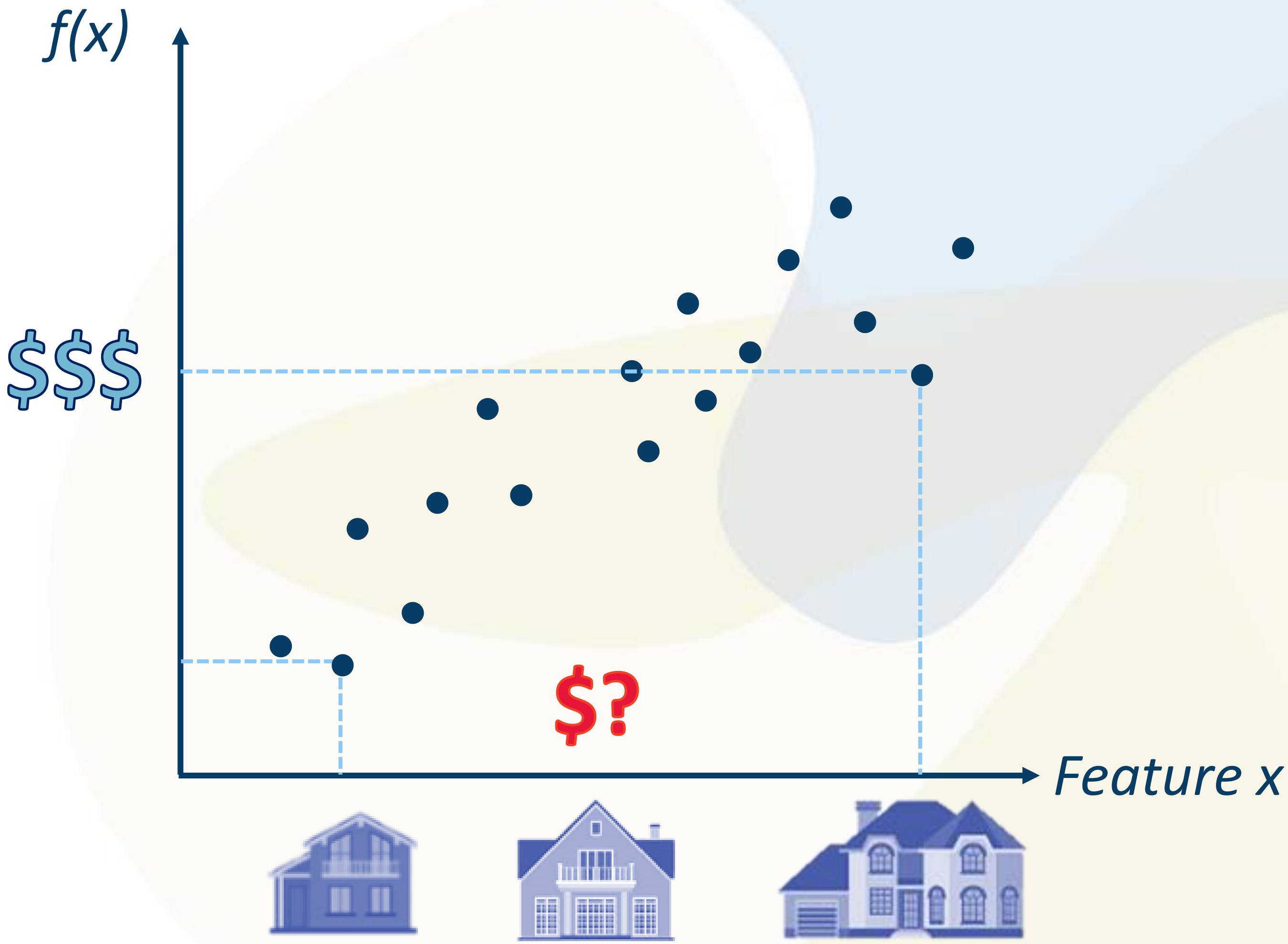
Create useful representations of data

Unsupervised Learning

x, ~~y~~

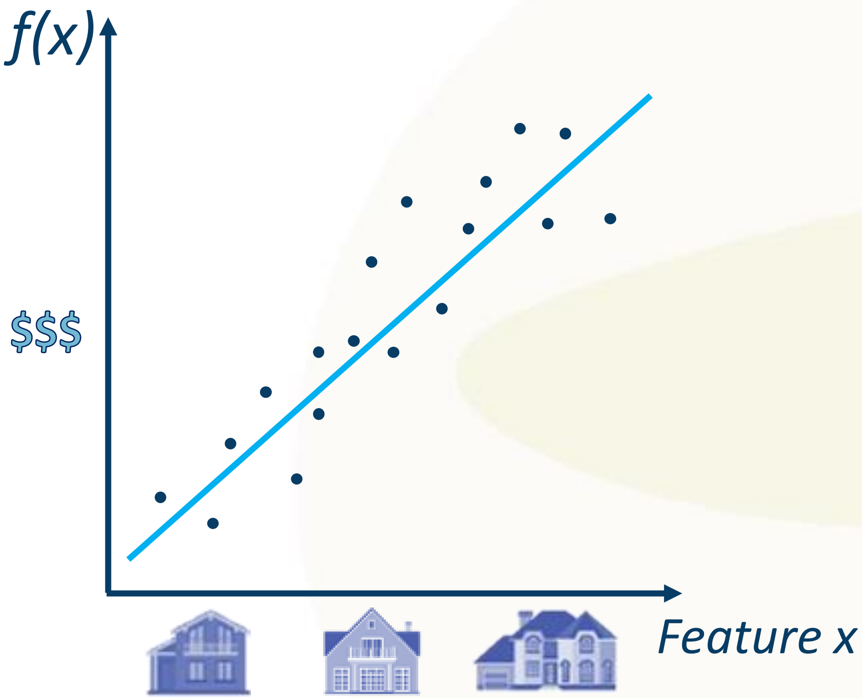

ML Algorithms
Automatically find patterns in data - prediction
Create useful representations of data
Supervised Learning
Unsupervised Learning
Discrete
Classification
Clustering
Continuous
Regression
Dimensionality Reduction

Regression

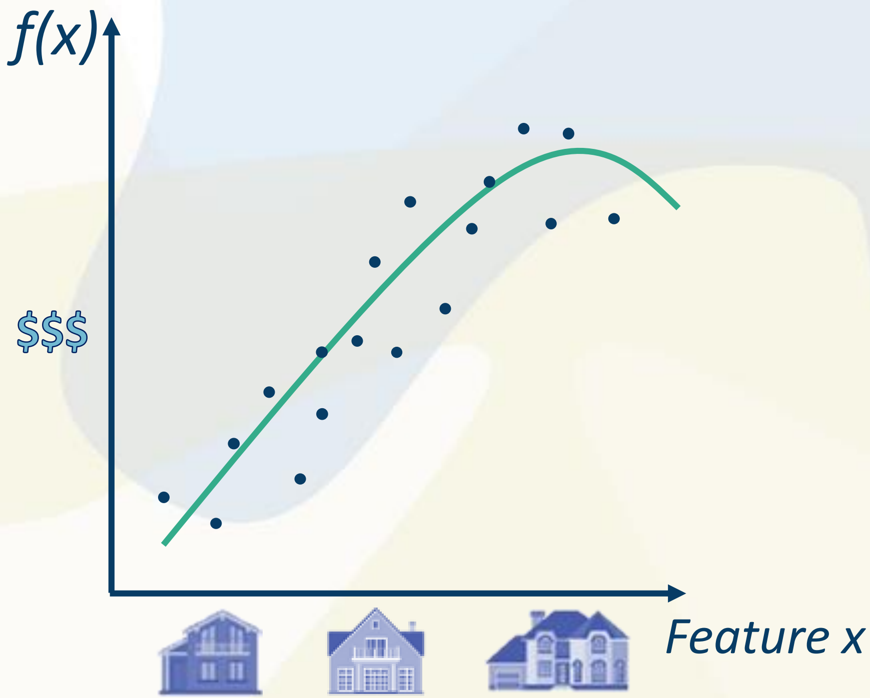


Regression

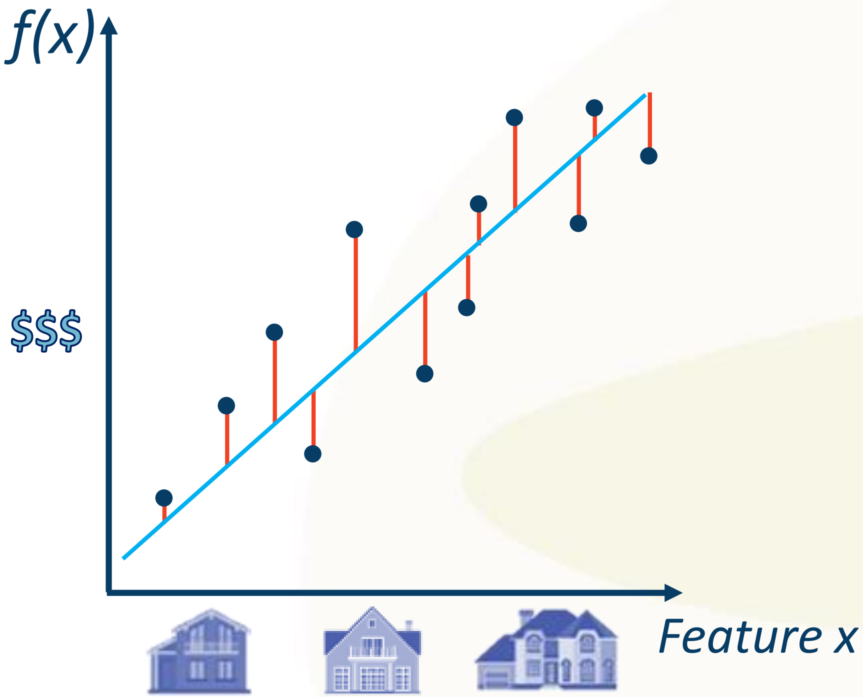
$y = ax + b$



$y = ax^2 + bx + c$



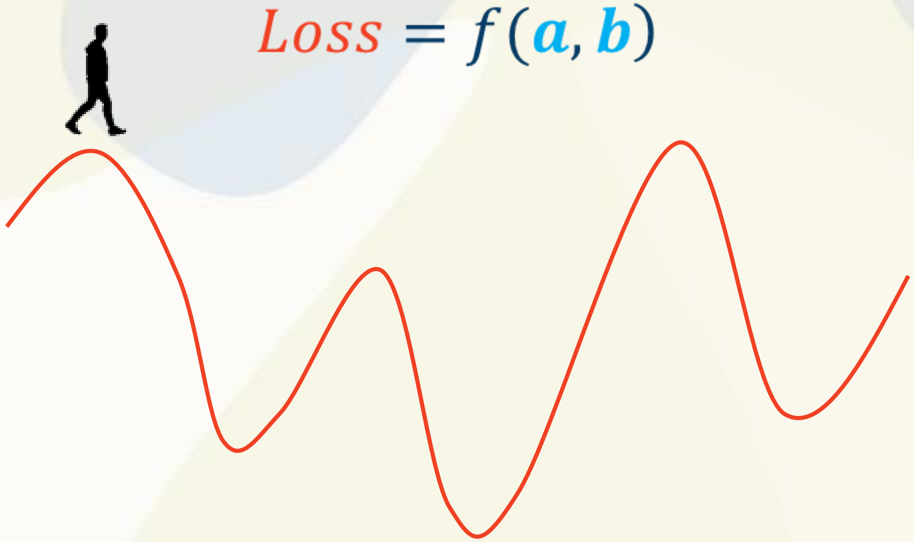
Regression



$$\hat{y} = \boldsymbol{a}x + \boldsymbol{b}$$

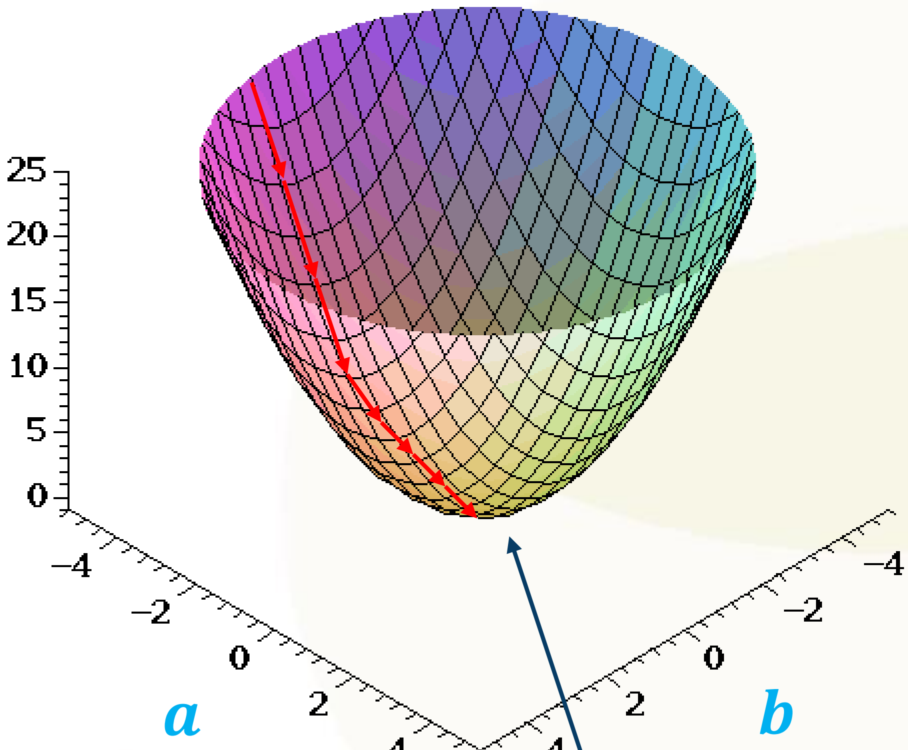
$$MSE = \sum (y - \hat{y})^2$$

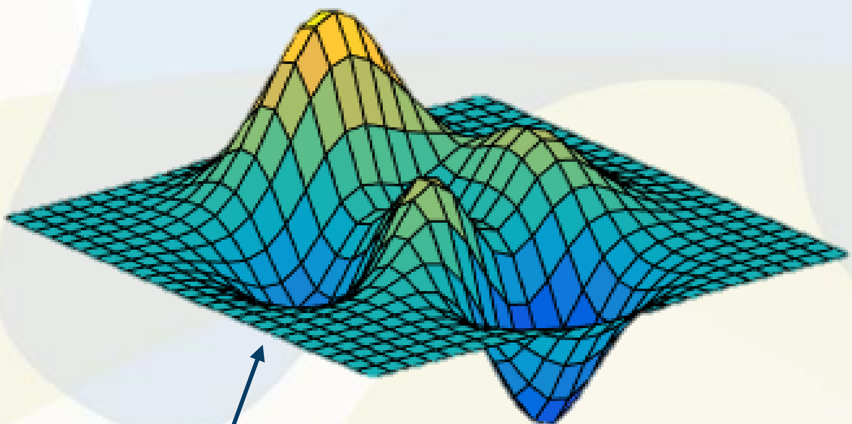
$$Loss = f(\boldsymbol{a}, \boldsymbol{b})$$



Gradient Descent

$$\mathit{Loss} = f(a, b)$$

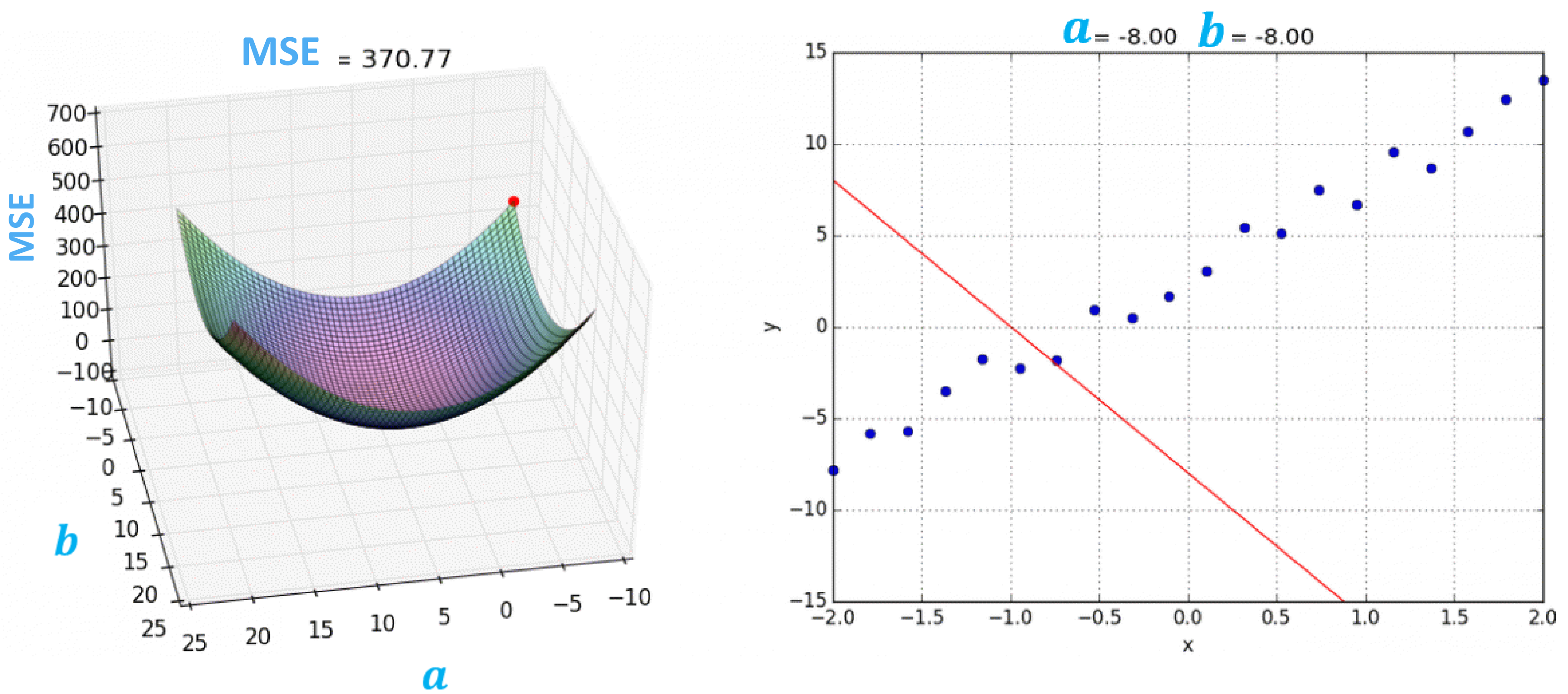




Gradient Descent

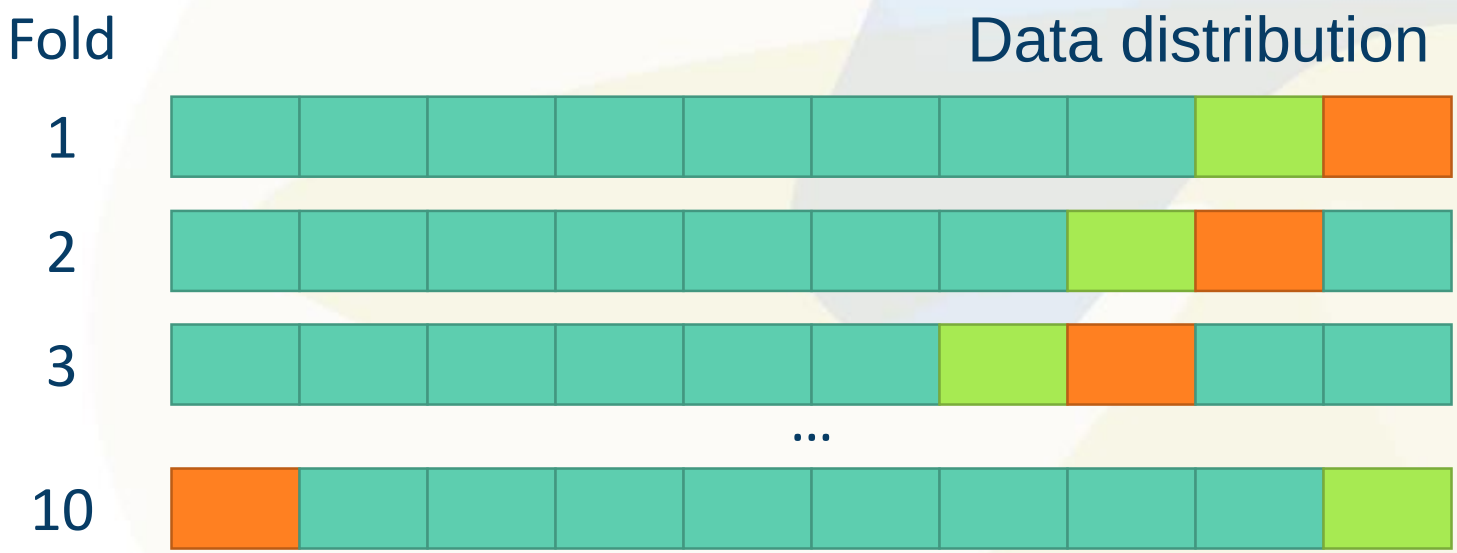
Initialize model parameters (a, b) randomly
Iterate between:

1) Compute *loss* (mean square error – MSE)
2) Update model parameters (a, b) in direction of gradient

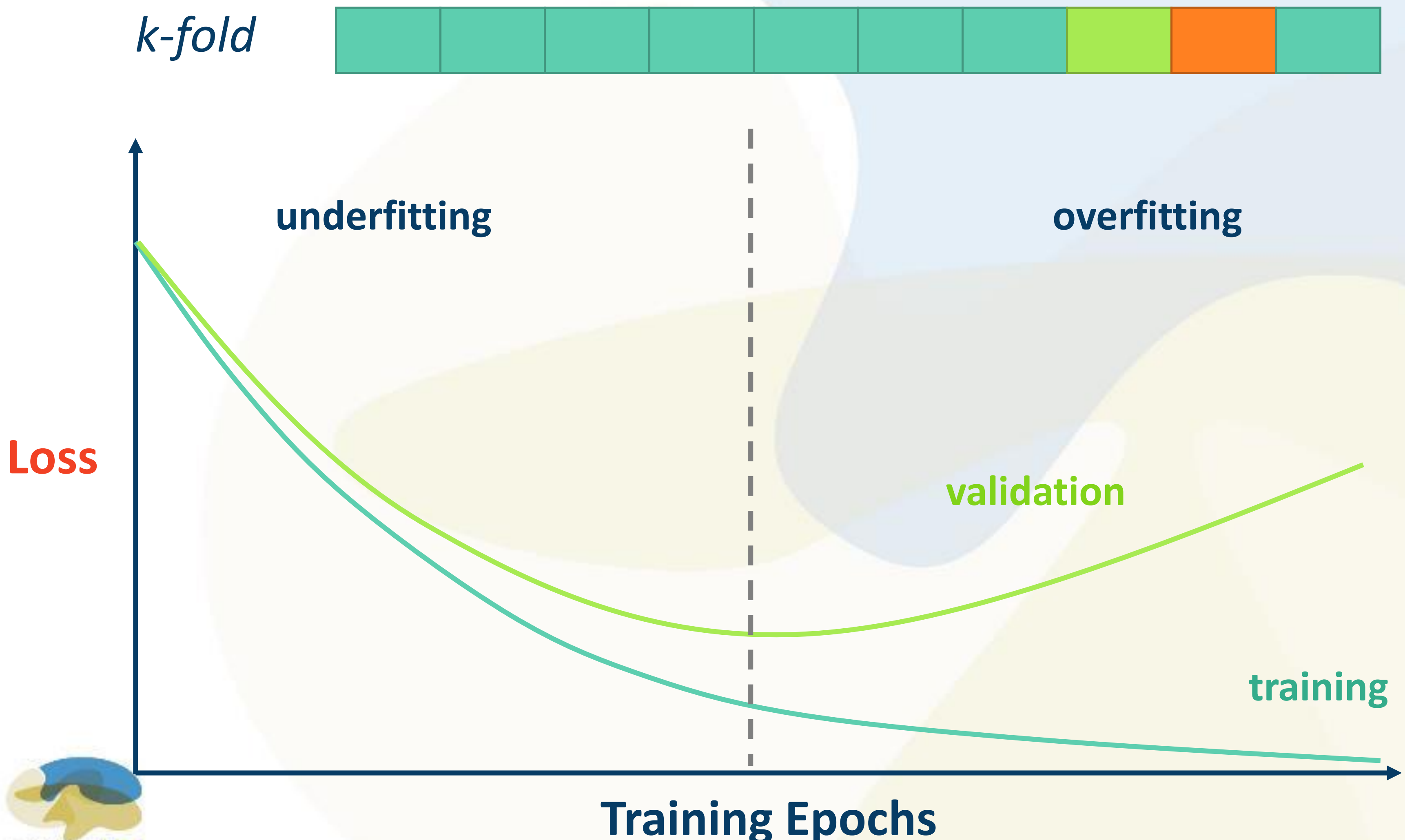


Cross-Validation

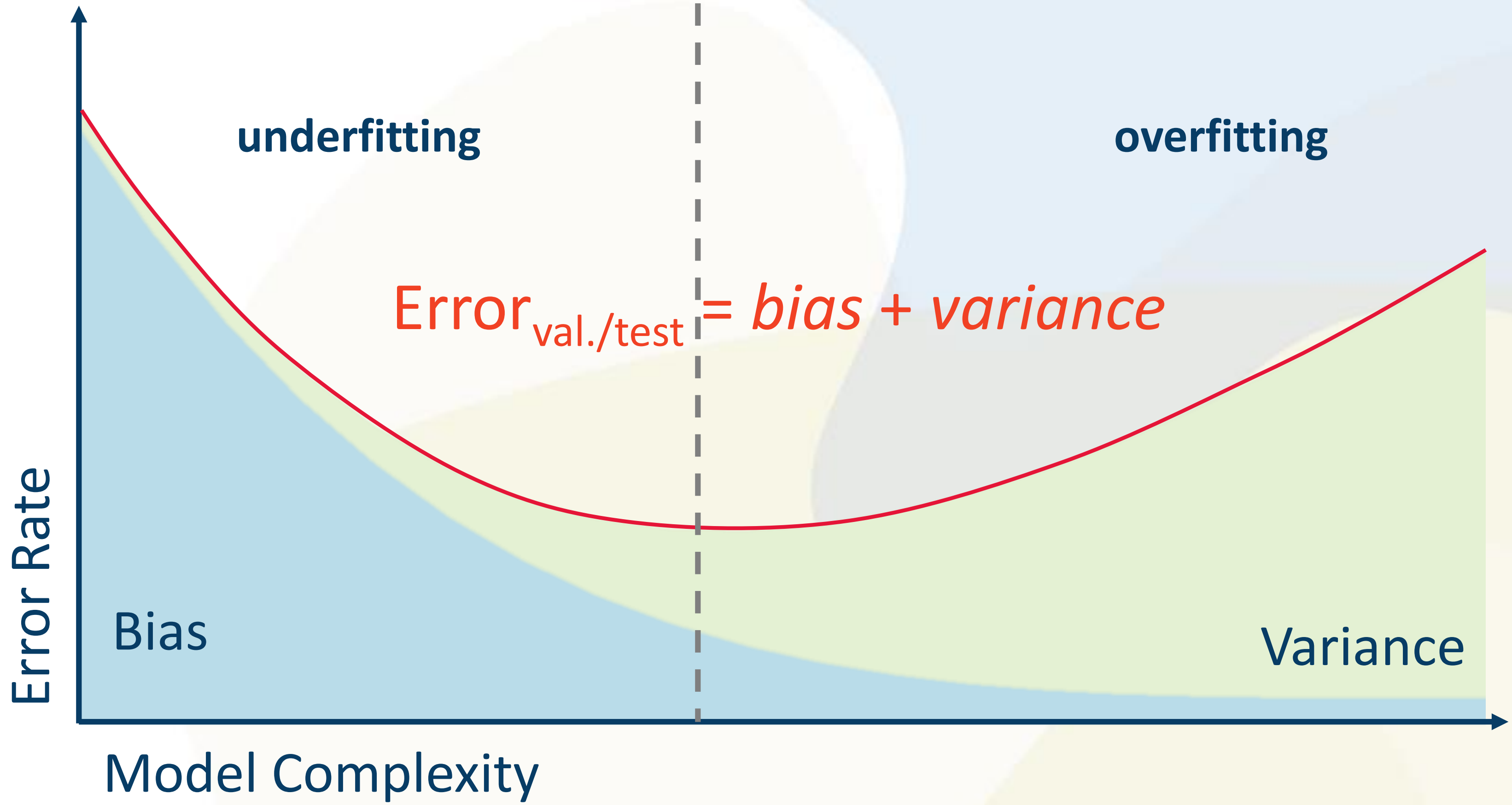
1) Train model parameters on **training** set
2) Evaluate training with the **validation** set
3) Report error on **test** set



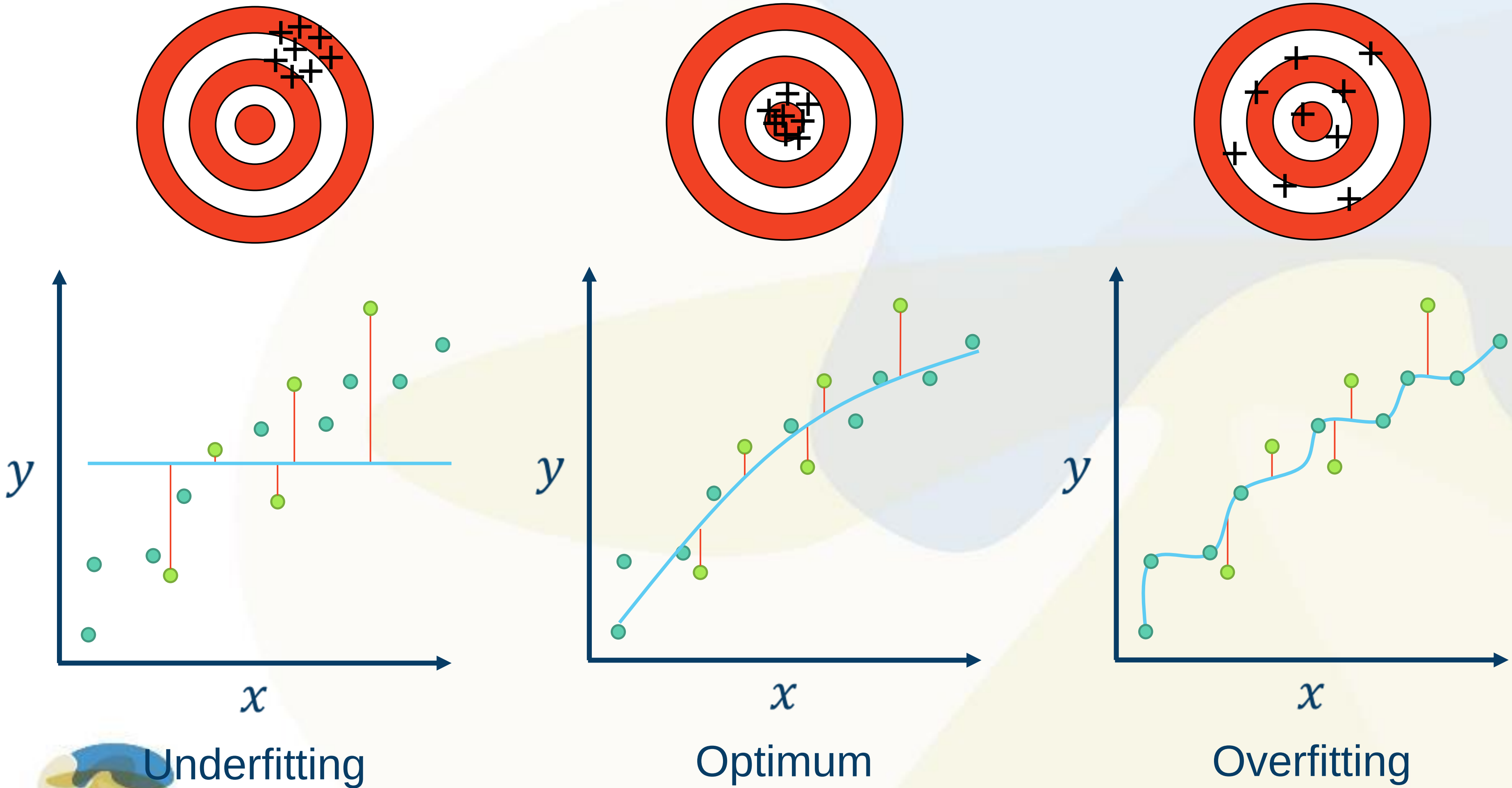
Overfitting



Bias-Variance Tradeoff



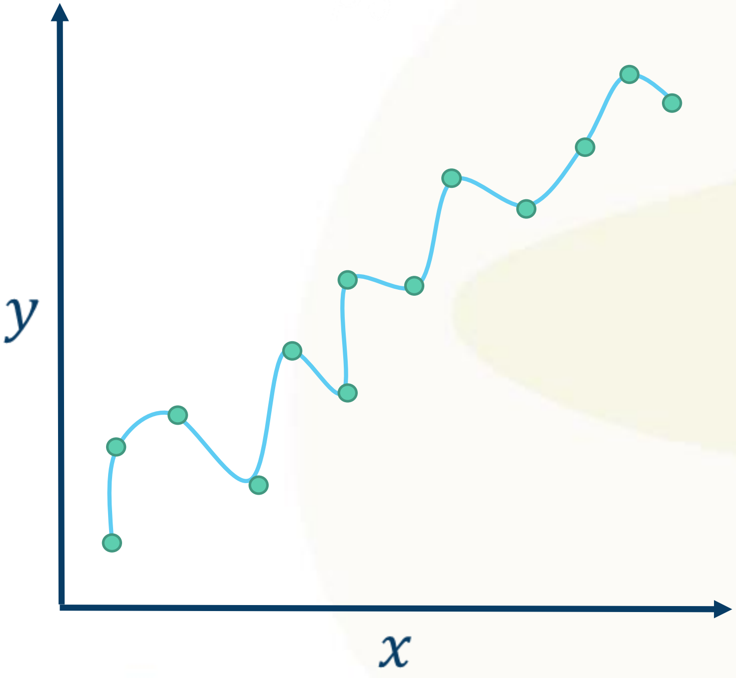
Bias-Variance Tradeoff



Training set   Test set

Regularization

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p$$



Penalties on the *LOSS* function to prevent overfitting!

Regularization

$$y = \cancel{\beta_0} + \beta_1 x + \cancel{\beta_2} x^2 + \cdots + \cancel{\beta_p} x^p$$

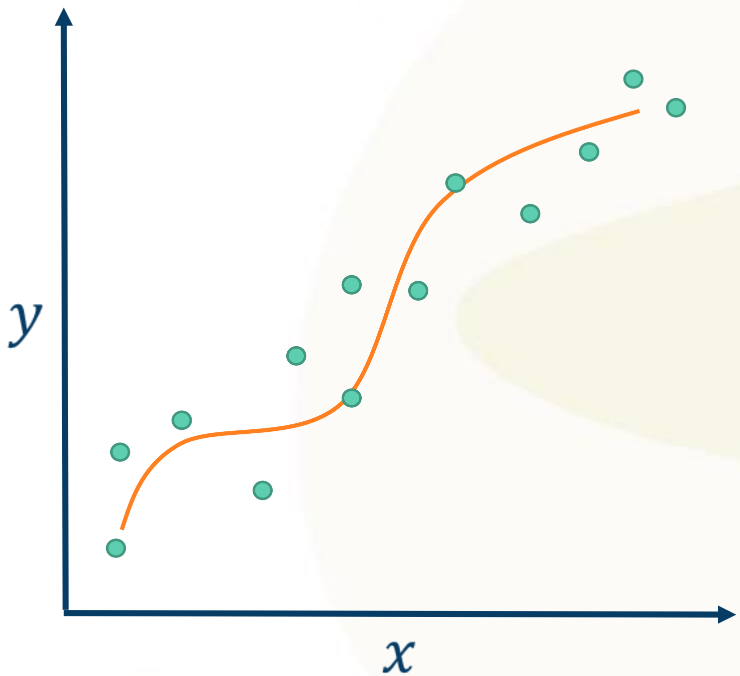
Penalties on the *LOSS* function to prevent overfitting!

1) L1/Lasso: constrains parameters to be **sparse**

$$MSE = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \boldsymbol{\lambda} \sum_{j=1}^{p} |\boldsymbol{\beta_j}|$$

Regularization

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p$$



Penalties on the *LOSS* function to prevent overfitting!

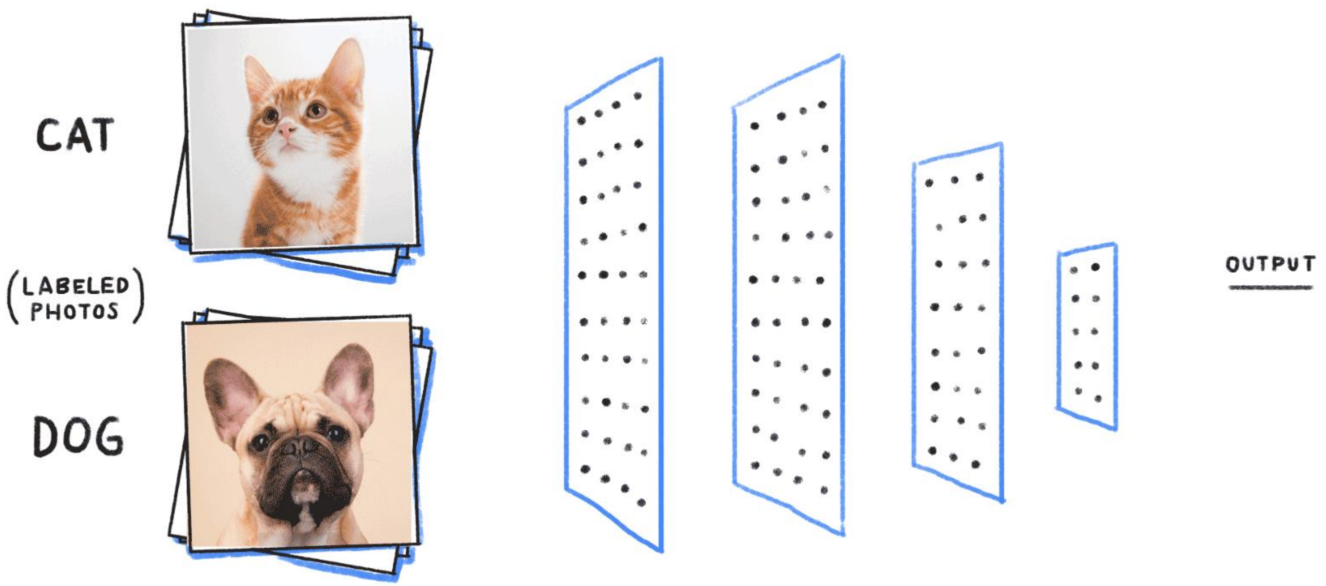
1) L1/Lasso: constrains parameters to be **sparse**

$$MSE = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

2) L2/Ridge: constrains parameters to be **small**

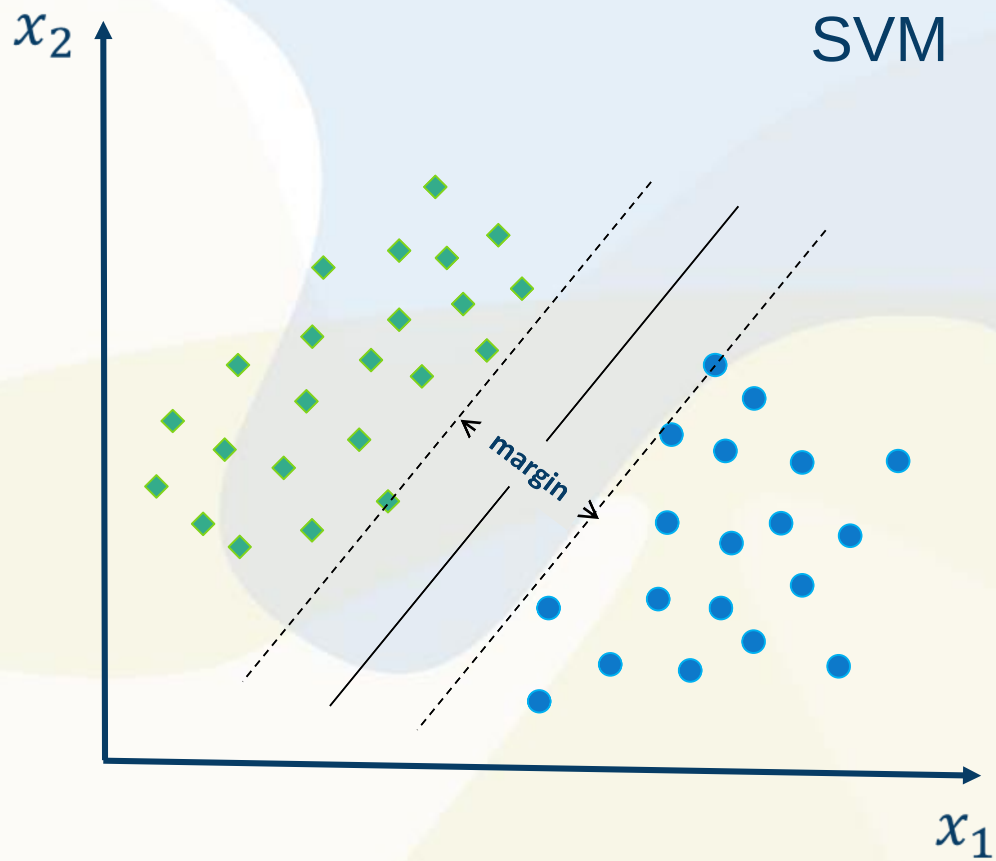
$$MSE = \sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2$$

Classification

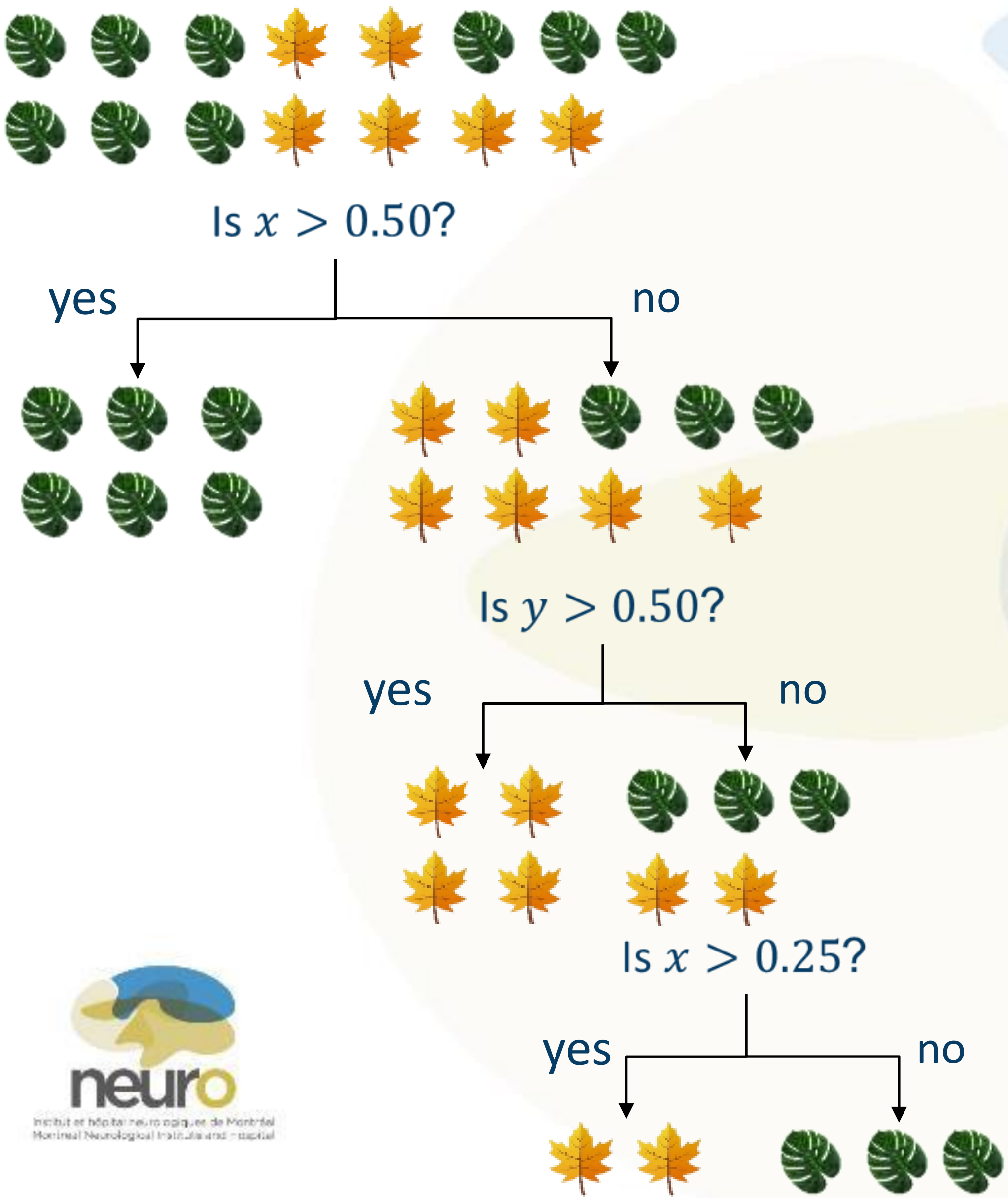


Classification

- Logistic Regression
- Support Vector Machine (SVM)



Classification



- Decision Trees
- Random Forests
- Neural Networks

ML Algorithms

Automatically find patterns in data → **Supervised Learning**

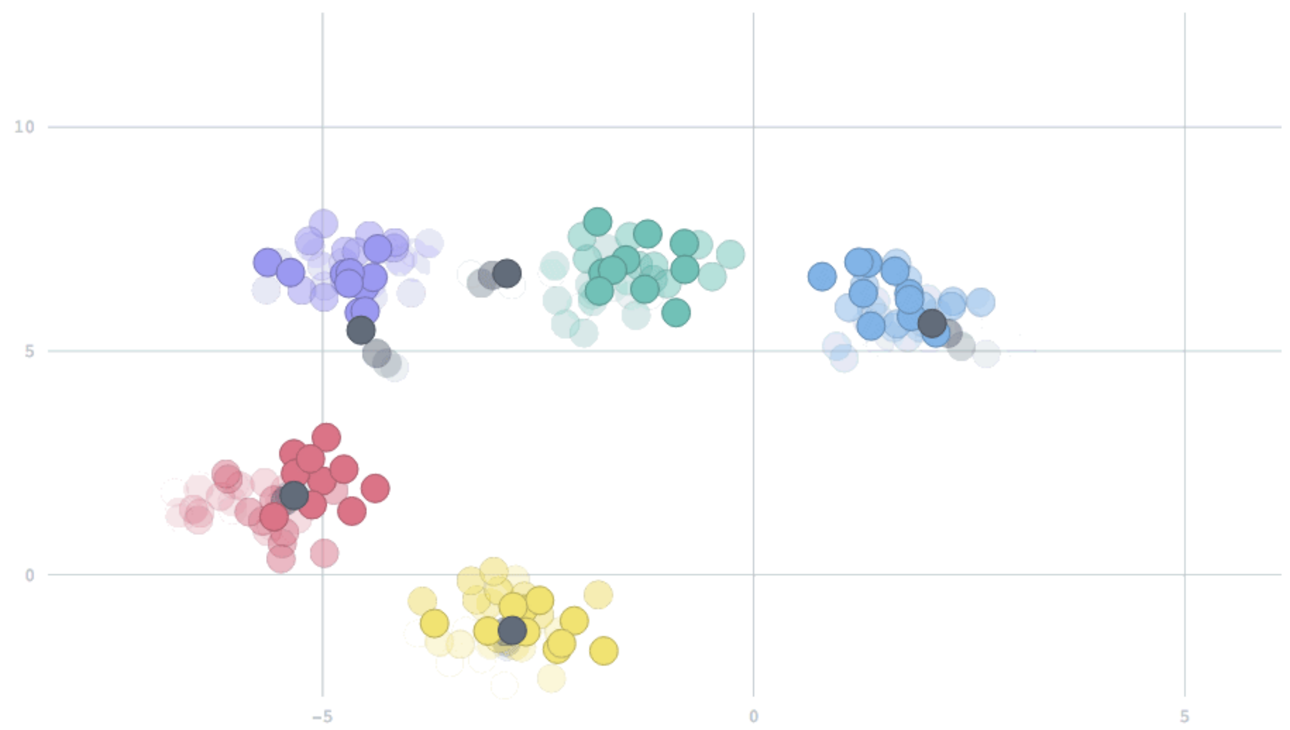
- Discrete: Classification
- Continuous: Regression

Create useful representations of data → **Unsupervised Learning**

- Clustering
- Dimensionality Reduction

Clustering

- *K*-means
- Hierarchical clustering
- Mixture of Gaussians



Dimensionality Reduction

- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- t-SNE