zh
aw
Life Sciences and
Facility Management

Institute of
Computational Life Sciences

Summary

Machine Learning

Norman Juchler

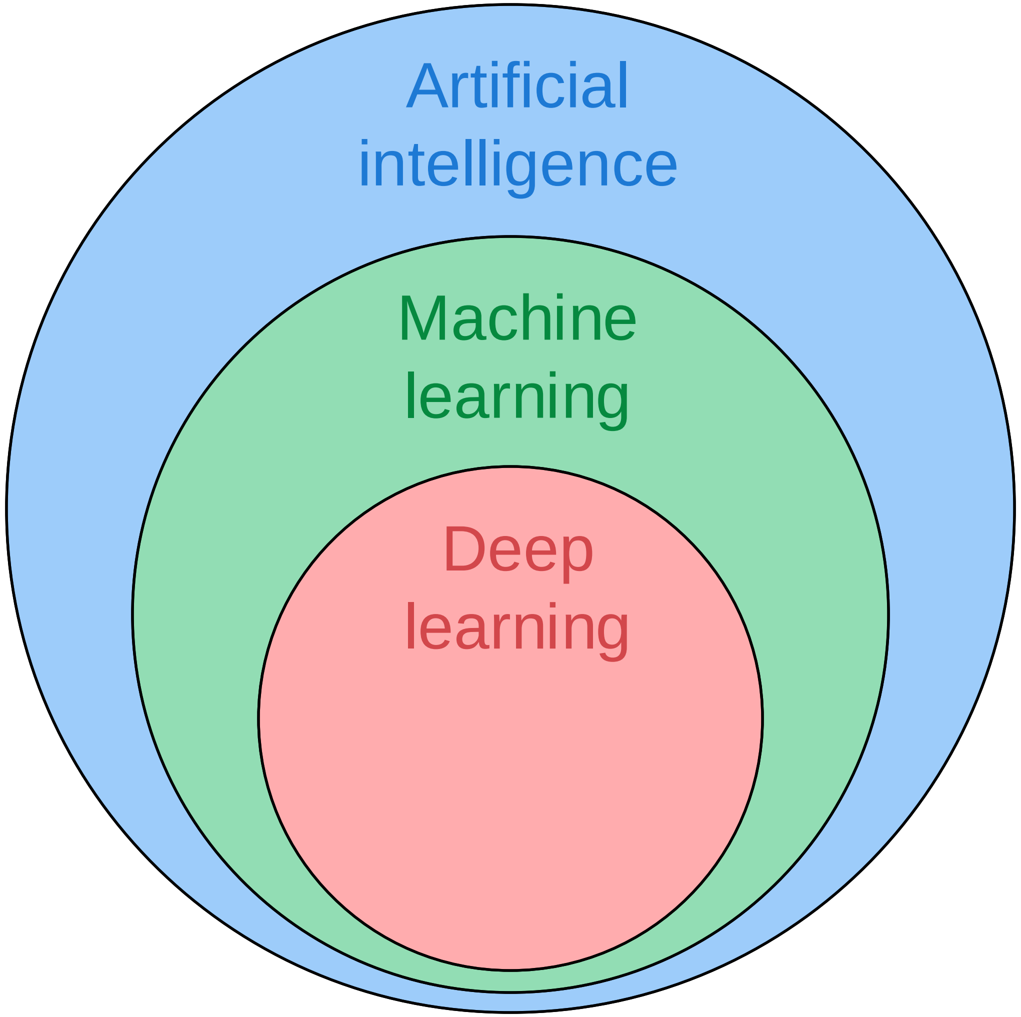
INSTRUCTIONS: Highspeed recap of the semester

Notes:

- The following slides summarize the topics covered in this module
- Although some slides have been modified, they are generally the same as the ones you already know from the corresponding lecture.
- It is probably advisable not to use this batch of slides for studying, as the slides are torn out of their context...

Fundamental concepts

What is artificial intelligence?



Distinctions:

- Statistics ≠ machine learning
  - ML: the focus is on prediction
  - Statistics: focus is on inference
  - ML theory and algorithms rely on statistics and probability theory
  - ML often makes fewer formal assumptions about data
- Machine learning ≠ statistical learning
  - The terms are often used interchangeably, but have slightly different meanings
  - SL: includes the formulation of stochastic models for the data generating processes
  - SL: besides the use for prediction, the focus is also on formally understanding the data

Formal introduction of a machine learning model

- In ML, a **model** is a mathematical representation, function or algorithm that defines the relationship between input data and the desired output.

Model (e.g., classifier or regression model)

Model parameters

$$y = f(x|\theta)$$

$$x \in \mathbb{R}^d$$

Problem dimensionality
d = 1: univariate
d > 1: multivariate

$$\theta \in \mathbb{R}^p$$

$$y \in \mathbb{R}$$

Defines problem type:
Binary classification

Response or labels
The information we want to predict. Often difficult or expensive to measure.

Predictors or features
The information we know about the subjects of interest. Usually easier to observe and access.

How a machine learning model operates

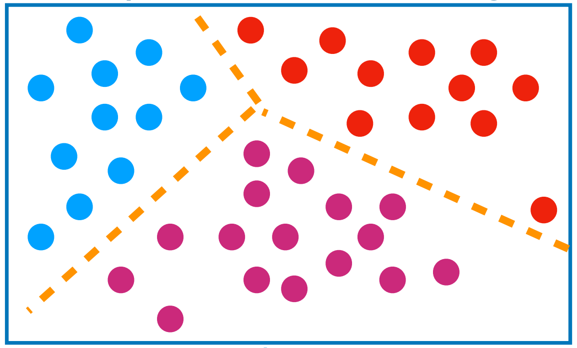
x = X-ray image

f () =

- COVID ✓
- Healthy X

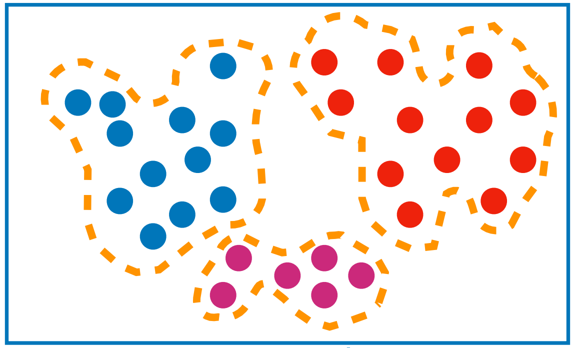
Machine learning paradigms

Supervised Learning



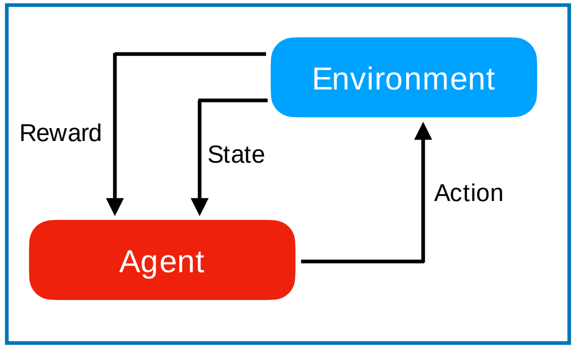
$\{x_i \in R^d, y_i \in R\}_{i=1}^{n}$

Unsupervised Learning



$\{x_i \in R^d\}_{i=1}^{n}$

Reinforcement Learning



Supervised Learning

x_i : 

y_i : panda lion dog

Unsupervised Learning

x_i : 

Reinforcement Learning

Agent: Kid

Environment: Neighbourhood street

Bias and variance in machine learning

- The **bias** is a property of the model that causes it to miss relevant relations between the features and the target output, e.g. because of wrong assumptions in the model.
- The **variance** measures how susceptible the model is to small changes in the training data. A model with high variance will change a lot if the training data is changed slightly.

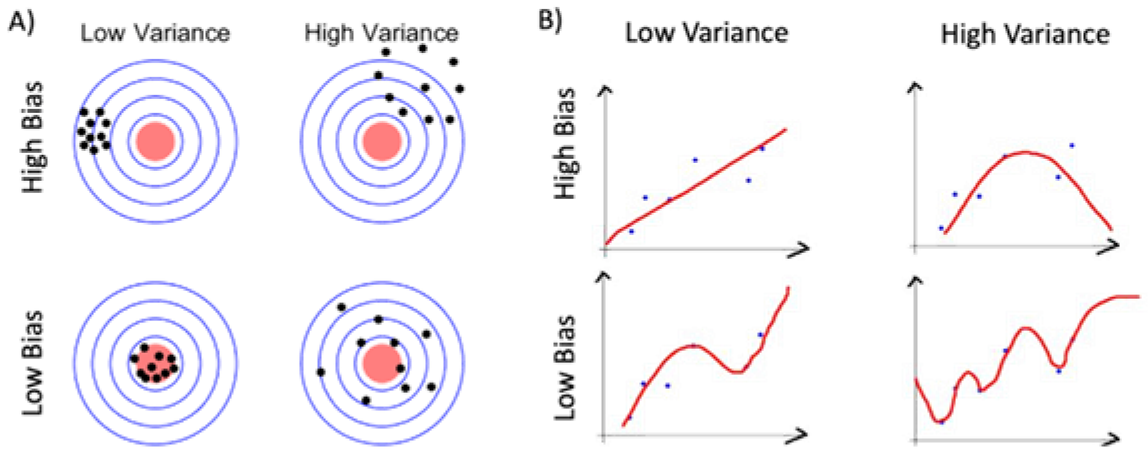
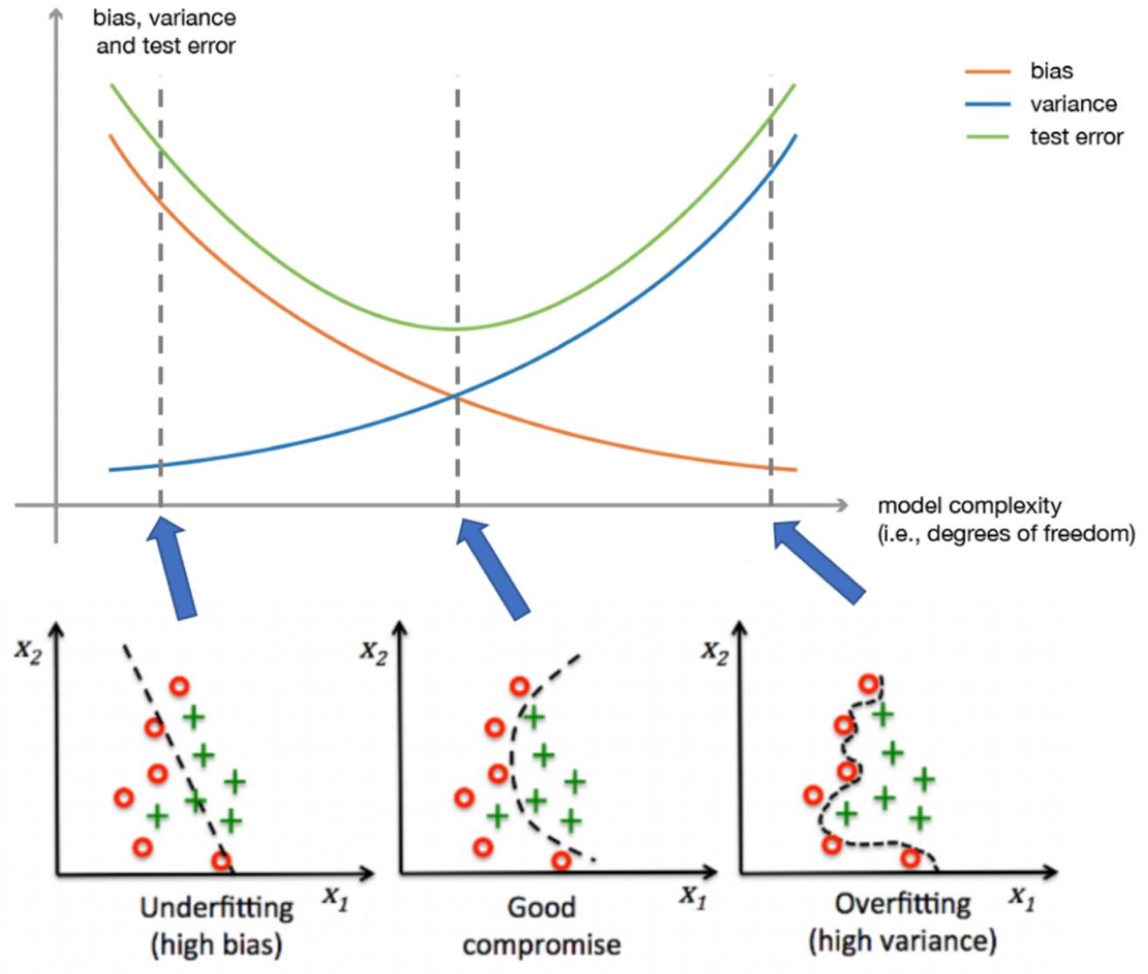


Figure 4: A: Statistical properties of bias and variance. B: Example of regression models for different combinations of bias and variance (assuming that the red curve in the lower left is the true data generating process).

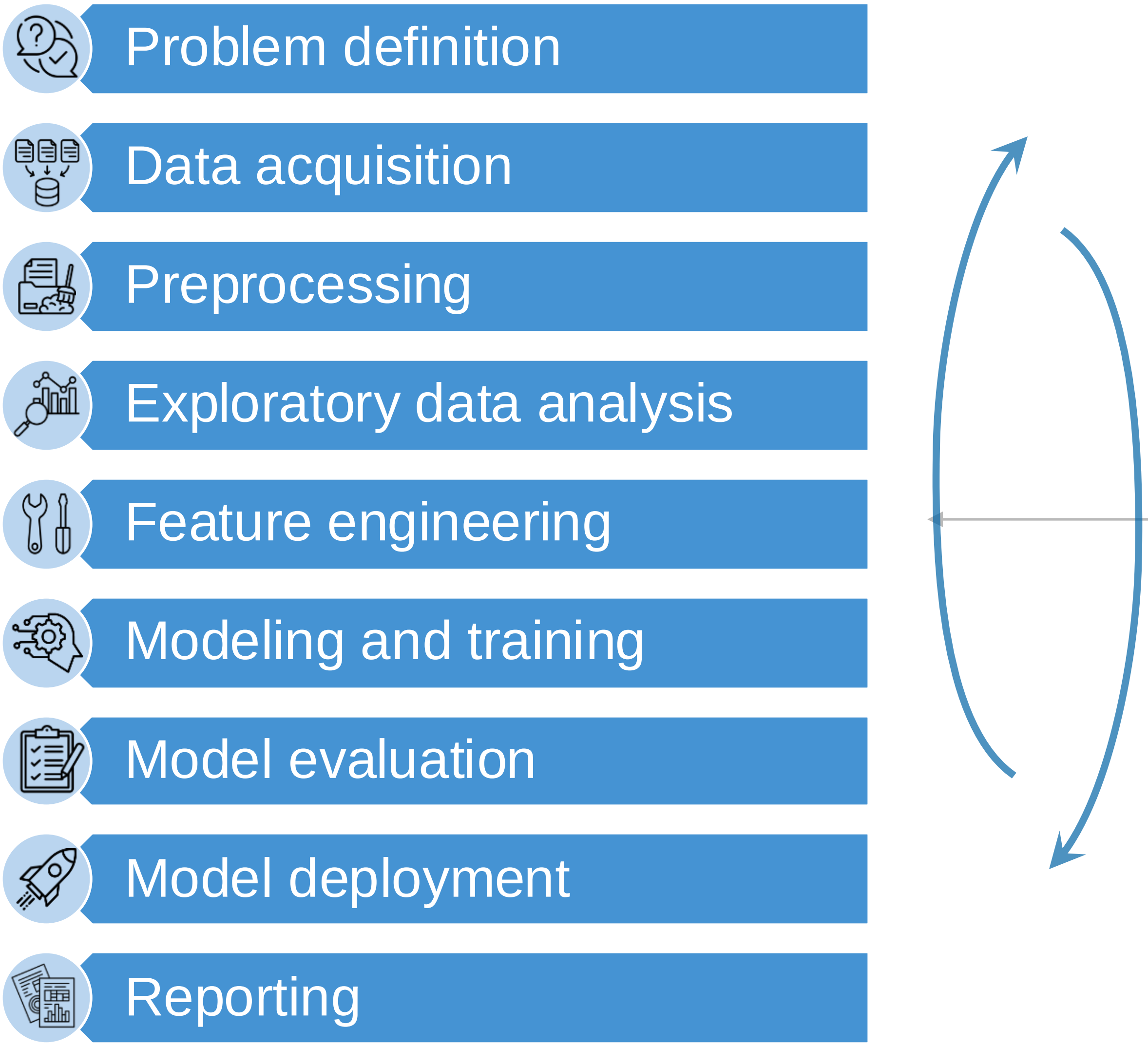
Model complexity vs. bias-variance tradeoff

We need to find the sweet spot in the middle!



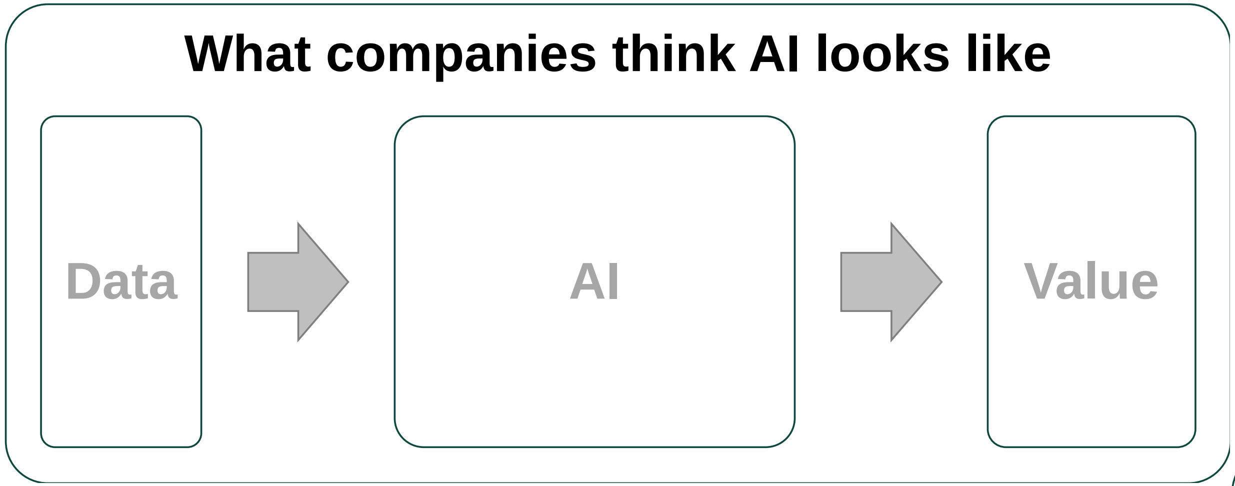
Data science workflow

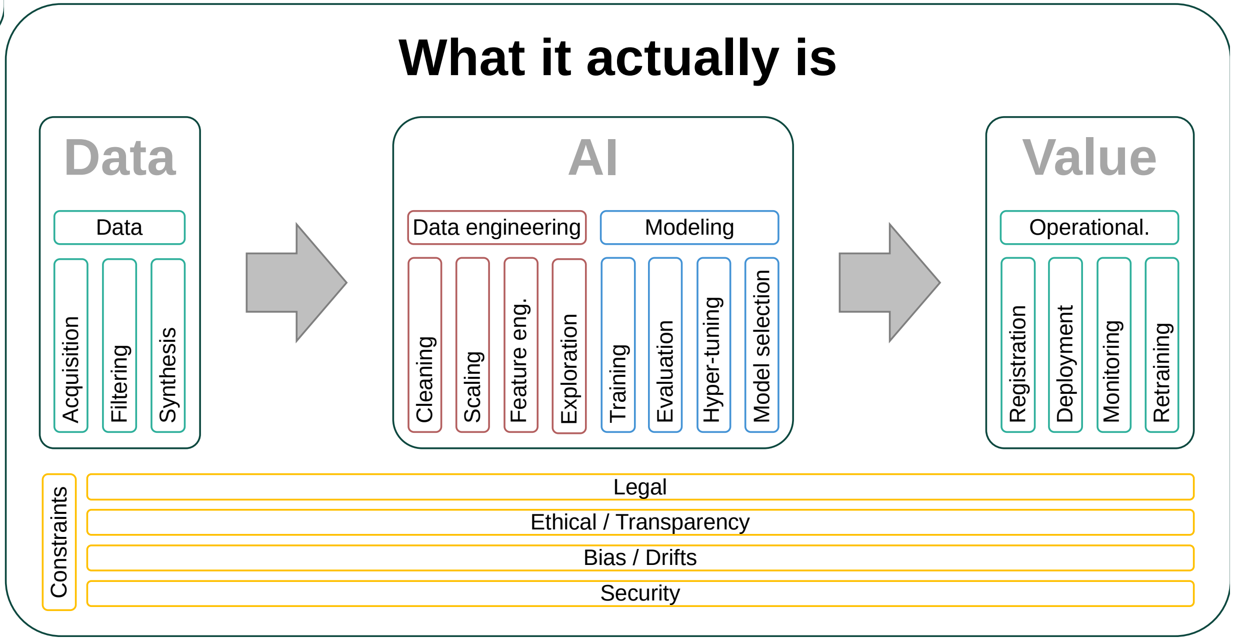
The data science workflow



- Implementing the workflow is an **iterative process**
- Repeat or revisit stages of the workflow based on new findings or feedback.

Another perspective...





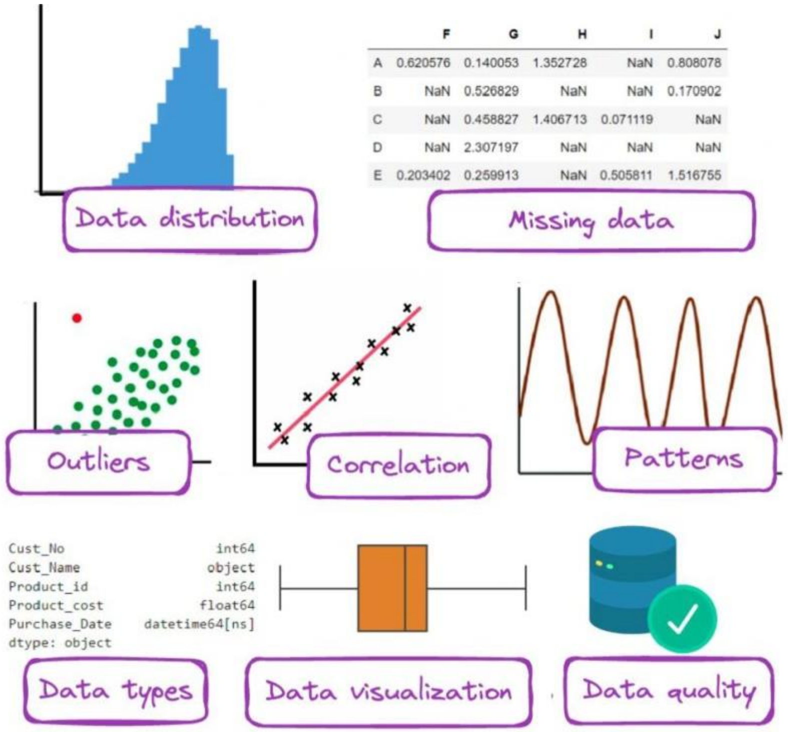
Source: Andy Scherpenberg, link, modified

Purpose of exploratory data analysis (EDA)

- **Understand the data's structure**: Identify patterns, relationships, trends, and anomalies in the dataset.
- **Detect missing or incorrect data**: Uncover issues such as missing values, outliers, or inconsistencies that need to be addressed.
- **Guide feature selection and engineering**: Inform decisions about which features are important or need transformation.
- **Generate hypotheses**: Formulate potential hypotheses and insights for further analysis or modeling.
- **Validate assumptions**: Check if the data fits certain assumptions required by statistical or machine learning models.

Means of EDA

- Summary statistics
- Data visualization
- Correlation analysis (between features)
- Univariate analysis (between features and target)



Training and validation

Different data splits

Train-test split

Train-validation-test split

Cross-validation (for evaluation)

Split 1: à Model 1

Split 2: à Model 2

Split 3: à Model 3

Split 4: à Model 4

Nested cross-validation

Outer fold 1:

Outer fold 2:

Outer fold 3:

Outer fold 4:

Outer fold 5:

Evaluation metrics: for regression

- **Mean squared error (MSE)**:
  - $\hat{y}_i$: predicted values
  - y_i: true values (from the training data)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Root mean squared error (RMSE)**:
  - Has the same unit as y

$$RMSE = \sqrt{MSE}$$

- **Mean absolute error (MAE)**:
  - Penalizes bigger errors less

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Evaluation metrics: for classification

- **Accuracy**:
$$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}+\mathrm{TN}} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{P}+\mathrm{N}}$$
Fraction of correctly classified events
- **Precision**:
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$$
Fraction of relevant instances among the retrieved instances
- **Sensitivity or recall** (true positive rate, TPR)
$$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = \frac{\mathrm{TP}}{\mathrm{P}}$$
Fraction of relevant instances that were correctly retrieved
- **Specificity** (true negative rate, TNR)
$$\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} = \frac{\mathrm{TN}}{\mathrm{N}}$$
Fraction of negative instances that were correctly retrieved
- The sensitivity quantifies the avoidance of false negatives, the specificity the same for false positives.

		True condition ( )	
	Total population	**P** Condition positive	**N** Condition negative
Prediction (^)	Predicted condition positive	**TP** True positive	**FP** False positive Type I error
	Predicted condition negative	**FN** False negative Type II error	**TN** True negative

Contingency table
(aka confusion matrix)

Specific methods

Overview of methods

Machine Learning

- StatQuest: gentle intro to machine learning (13 min)
- Arten des maschinellen Lernens (6 min)

Unsupervised Learning

Clustering

- StatQuest: K-means clustering (8 min)
- StatQuest: hierarchical clustering (11 min)
- (StatQuest: clustering with DBSCAN (9 min))

Principle Component Analysis

- StatQuest: PCA main ideas in 5 minutes
- StatQuest: Principal component analysis (22 min)
- StatQuest: PCA - practical tips (8 min)
- Mathematische Grundlagen von D. Jung erklärt: Eigenraum einer Matrix, Lineare Algebra, Matrixalgebra, Unimathematik

Data Preprocessing

Explorative data analysis, e.g., Pandas, Seaborn Plots
Data cleaning, e.g., missing values
StatQuest: Pearson correlation (19 min)

Supervised Learning

Linear Regression

- StatQuest: main ideas of fitting a line to data (9 min)
- StatQuest: linear regression (27 min)
- StatQuest: multiple lineare regression (11 min)

Logistic Regression

- StatQuest: odds and log(odds) (12 min)
- StatQuest: logistic regression (8 min)
- StatQuest: maximum likelihood (6 min)
- StatQuest: logistic regression …: coefficients (16 min)
- StatQuest: logistic regression …: max. likelihood (10 min)
- StatQuest: confusion matrix (7 min)
- StatQuest: specificity and sensitivity (12 min)

Decision Trees

- StatQuest: decision and classification trees (18 min)
- StatQuest. decision trees part 2 (5 min)
- StatQuest: regression trees (22 min)
- StatQuest: random forests part 1 (9 min)
- StatQuest: gradient-boosted trees part 1 (15 min)
- (StatQuest: XGBoost part 1 (25 min), follow with parts 2-4)

Deep Learning

- MIT: introduction to deep learning (56 min)
- MIT: deep computer vision (CNNs) (55 min)

Model evaluation

- StatQuest: main ideas of fitting a line to data (least-squares) (9 min)
- StatQuest: R2 score (11 min)
- StatQuest: entropy (16 min)
- StatQuest: cross validation (5 min)
- StatQuest: confusion matrix (7 min)
- StatQuest: sensitivity and specificity (11 min)
- StatQuest: ROC and AUC (16 min)

Explainable AI (XAI)

- DeepFindr: SHAP – model agnostic (15 min)
- Grad-CAM – for deep learning

This overview has been assembled by Susanne Suter from FHNW

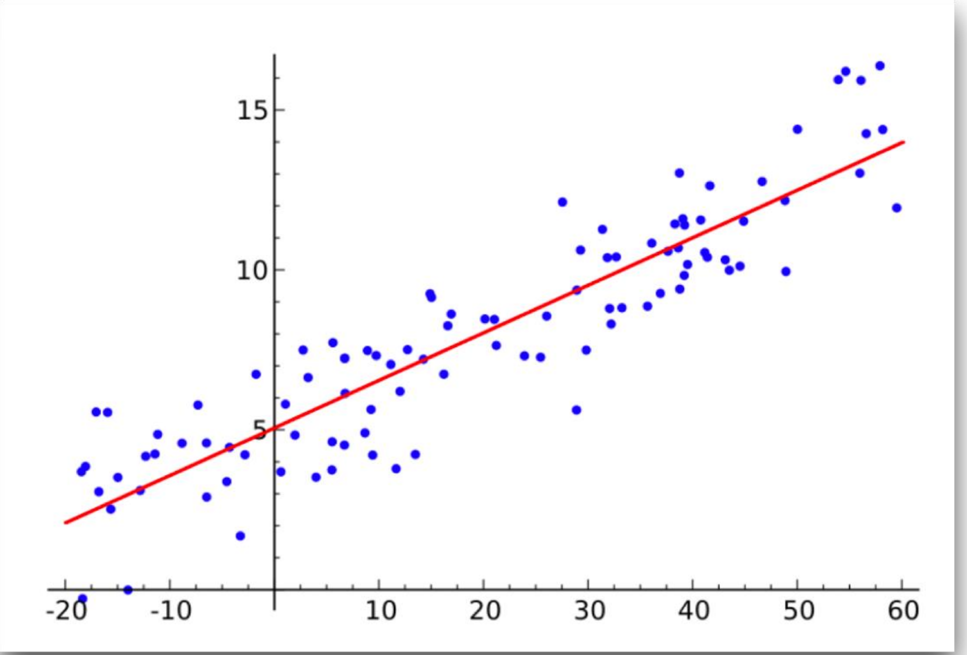
Linear models

- Assume we have a set of p features: $\boldsymbol{x}_i := (x_{i1}, x_{i2}, \ldots, x_{ip})$
- We want to use them to predict a target variable y
- The simple assumption we can make is (model ansatz):

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}$$

$$f_i(\boldsymbol{x}, \boldsymbol{\beta}) = \boldsymbol{\beta x}$$

- Where the β_i are unknown parameters that we want to determine from the data



Loss functions

- A loss function (a.k.a. cost function) in the context of machine learning usually measures how well the predicted values match the true target values.

$$\mathcal{L}(y - \hat{y})$$

- For regression, we used the residual sum of squares (RSS) as loss function.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- Sometimes, it is more meaningful to compute the **mean squared error** (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{n} RSS$$

- (Note: both yield the same optimal solution!)

Numerical methods for optimization: Gradient descent

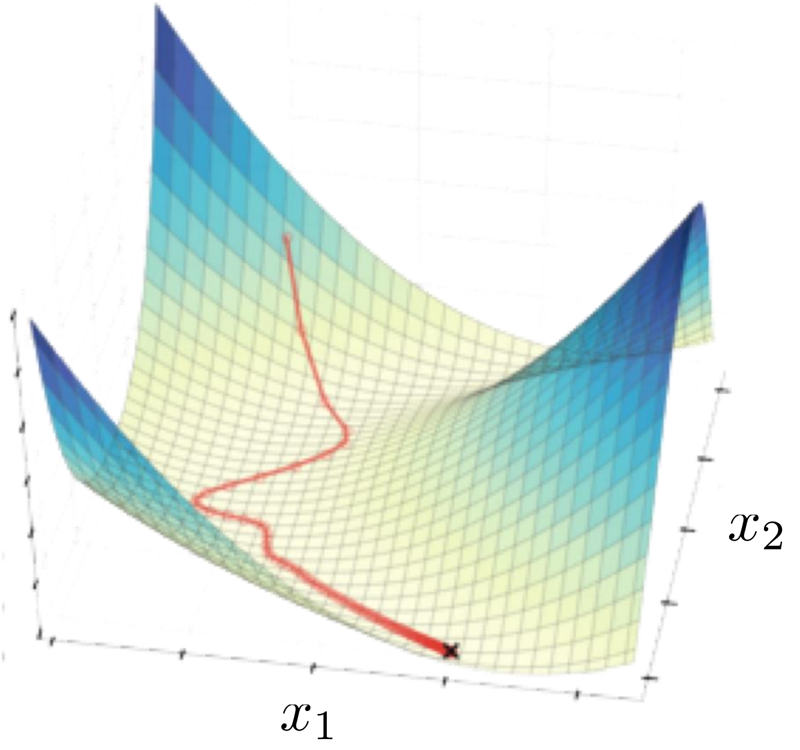
- Gradient descent is another iterative method to find solutions to $f'(\boldsymbol{x}) = 0$
- The method advances a small step in the direction of the gradient each time
- Gradient descent is applicable under more relaxed conditions

Gradient = direction of steepest assent

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \gamma_n \nabla f(\boldsymbol{x}_n)$$

Learning rate (adjustable)

$\mathcal{L}(\boldsymbol{x}) = f(\boldsymbol{x})$



Regularized regression

- Idea: Prevent overfitting by penalizing large coefficients, encouraging simpler models that generalize better.

- How? Modify the loss function!

Regularized loss = Original loss + Penalty term on coefficients

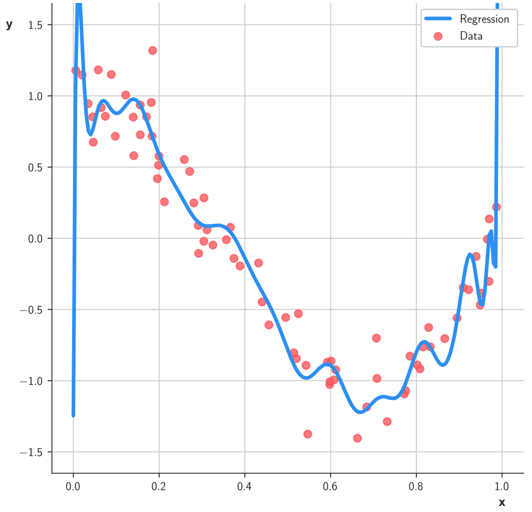
- Example: (Ridge regression)

$$\mathcal{L}(\beta|X,y) = MSE(\beta|X,y) + \lambda \sum_{j=1}^{p} \beta_j^2$$

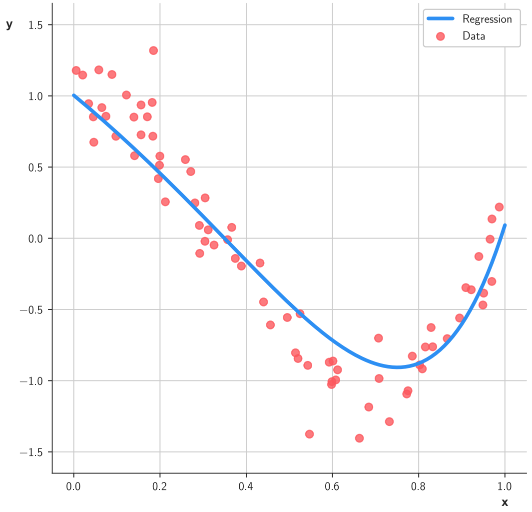
Recap:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i|\beta))^2$$

- This introduces a **hyperparameter λ** (or α in sklearn):
  - Controls the strength of regularization
  - Larger α increases penalty, leading to smaller coefficients



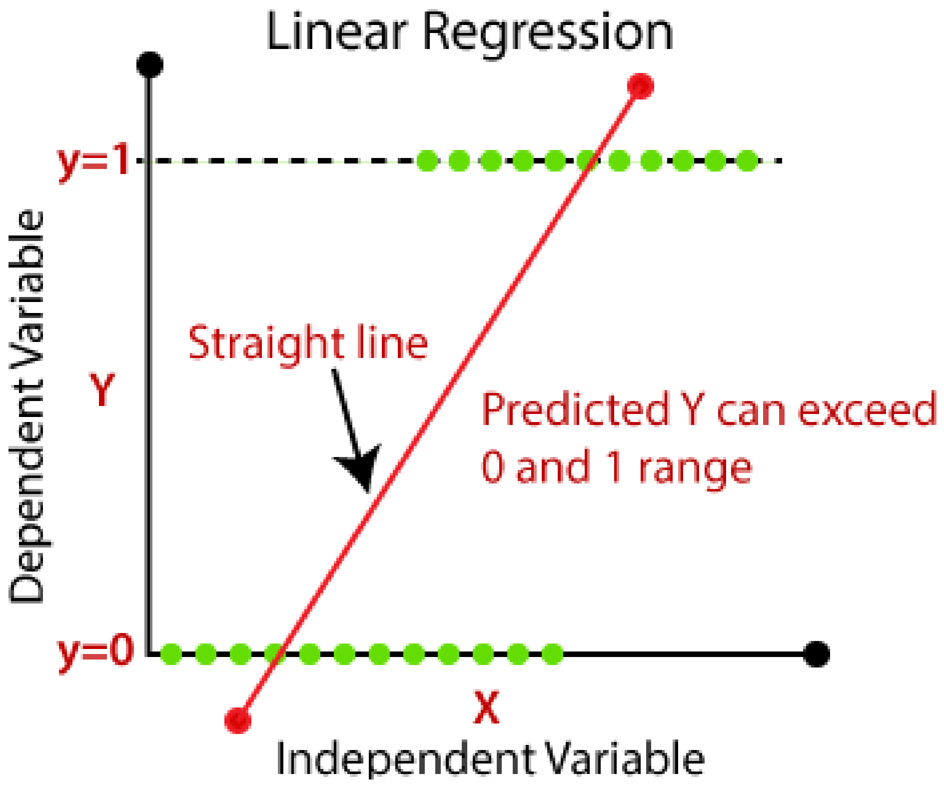
Polynomial regression without regularization showing overfitting

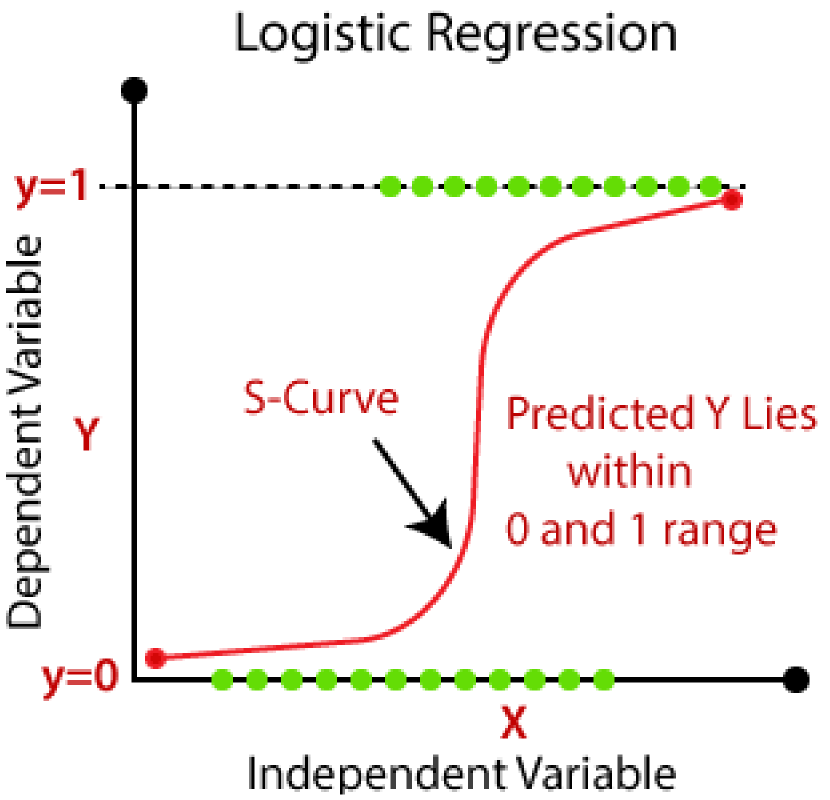


Polynomial regression with regu-larization (ridge), which in this case prevents overfitting.

Logistic regression for binary classification

- It makes little sense to compute a linear regression for the classification.
- Use a logistic regression instead!





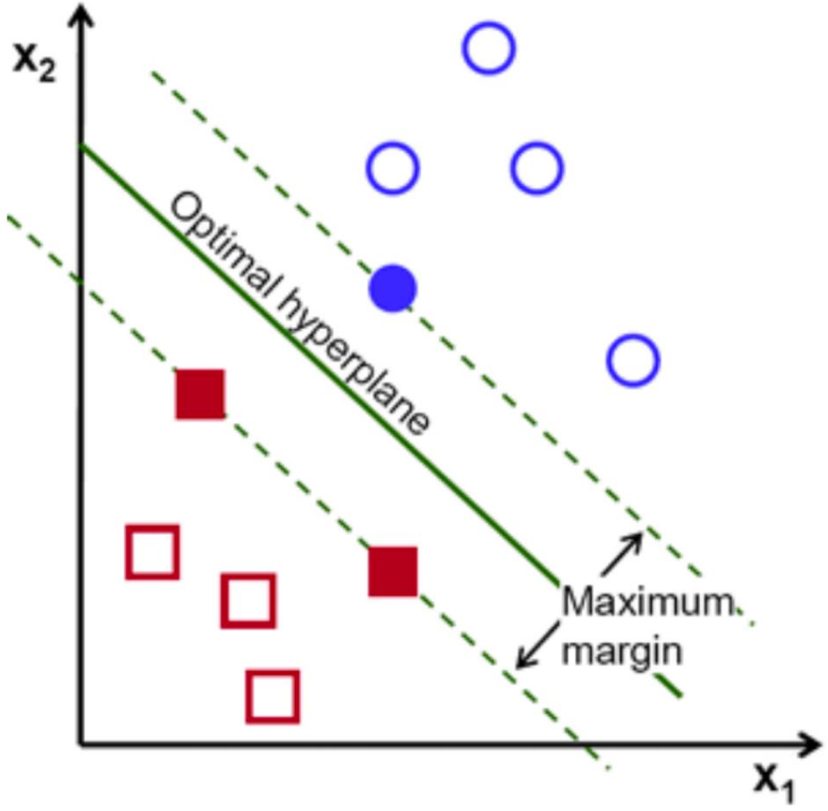
Support vectors and margin (SVMs)

- Linear classifier: Find **hyperplane** that best separates the classes in the feature space
- Any hyperplane ca be expressed as

$$\mathbf{w}^T\mathbf{x} - b = 0$$

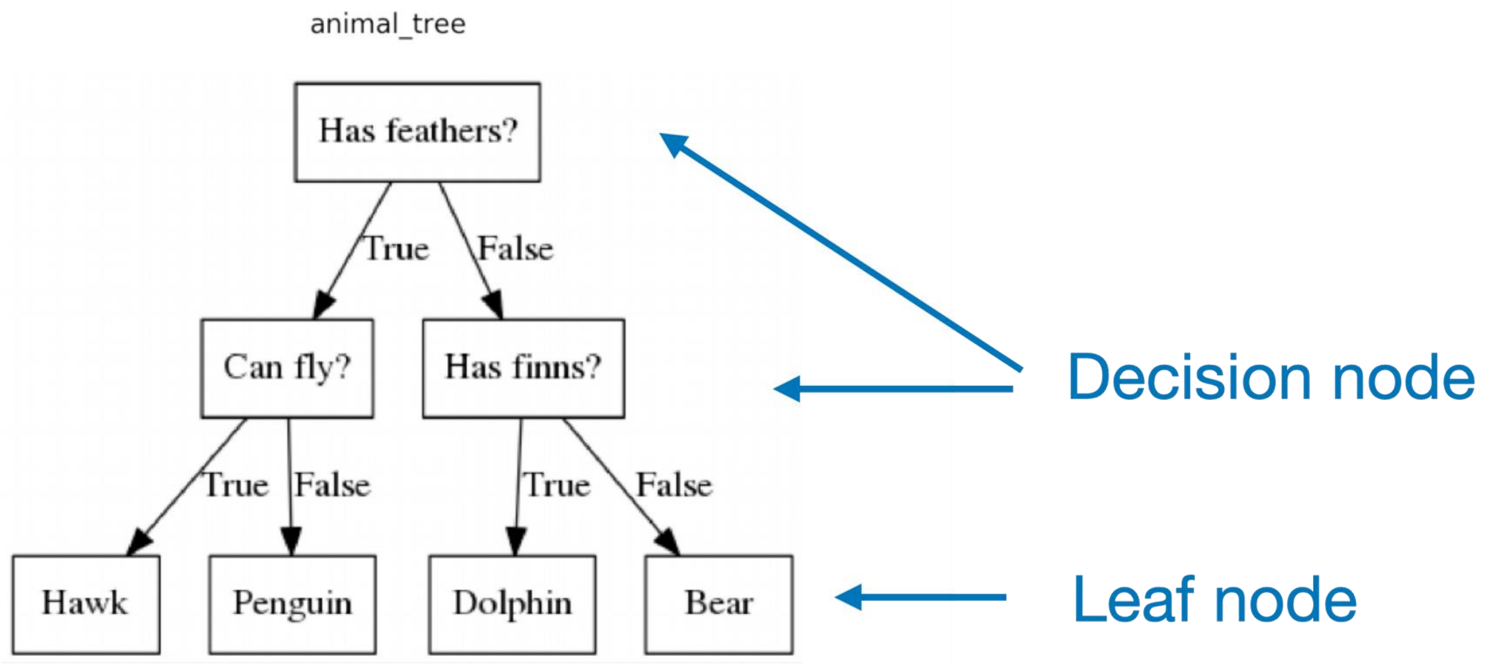
- Geometric interpretation: **w** is the normal of the decision boundary / hyperplane!

- **Definition**: The nearest points of each class to a hyperplane are called **support vectors**
- **Key idea**: SVMs find the hyperplane with maximal distance (**margin**) to the support vectors.



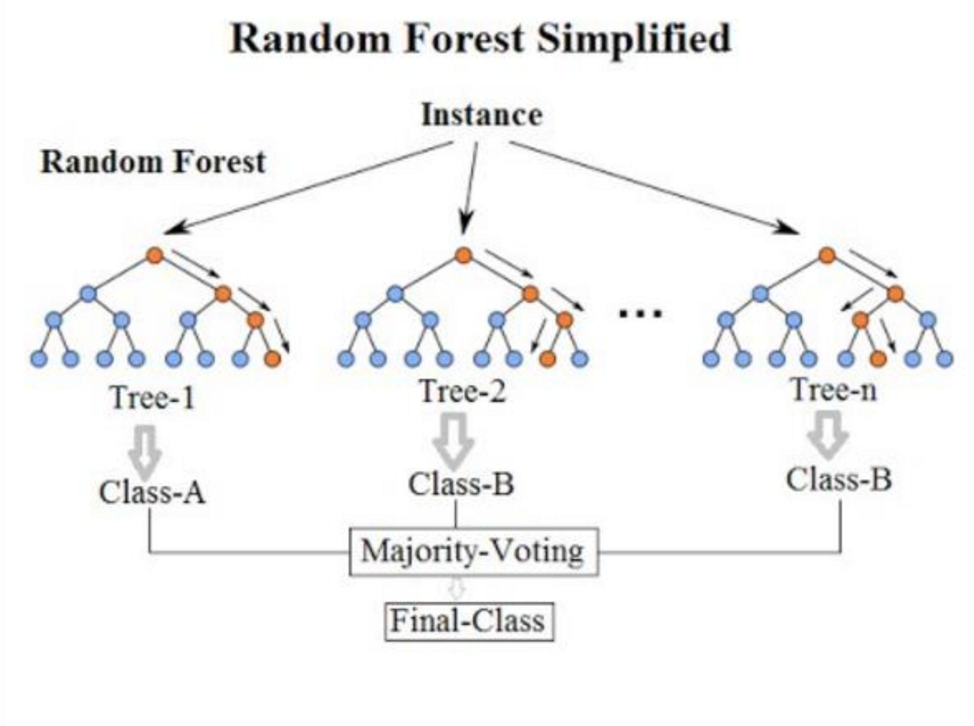
Decision trees

- Main idea: Algorithmically learn to construct a set of decision rules in the form of a tree.



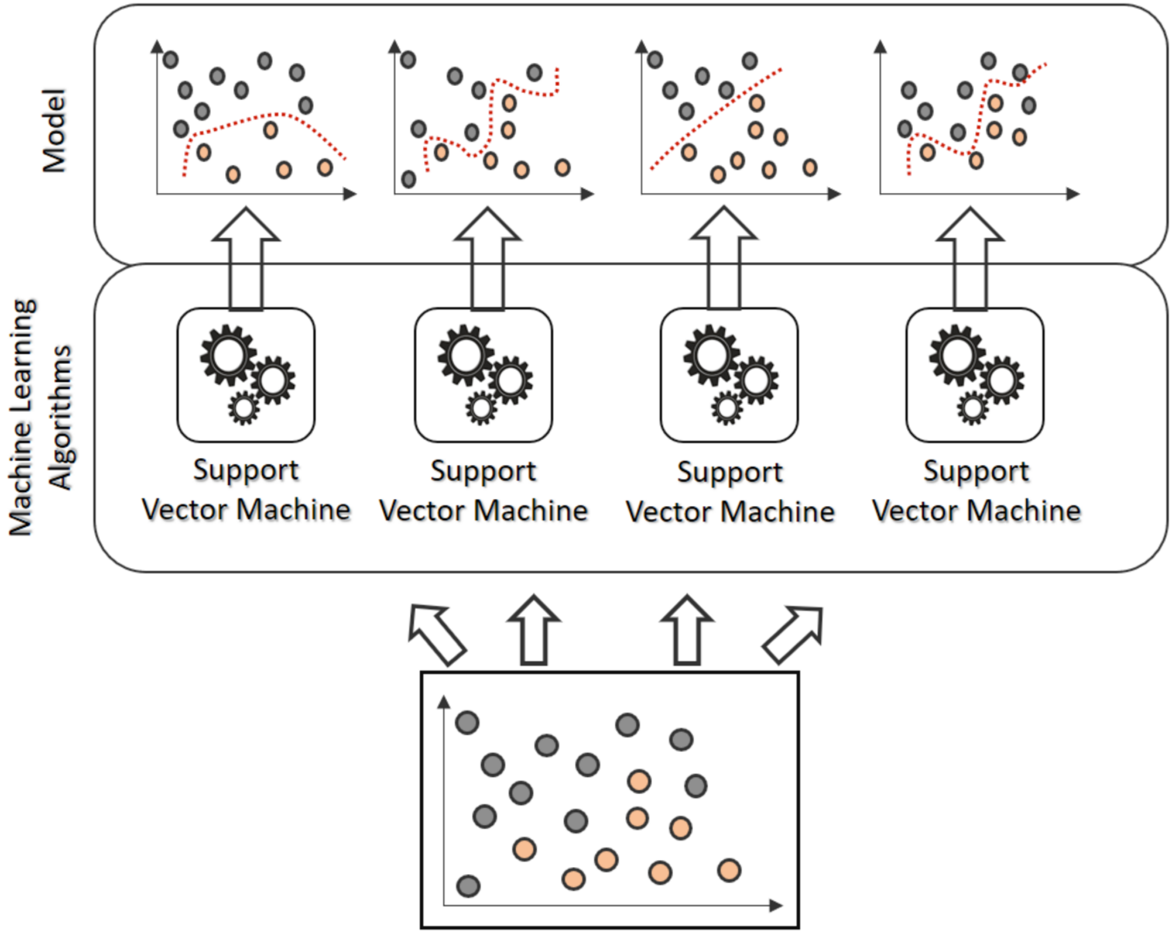
Random forest: Algorithm

- An ensemble of decision trees trained with bagging.
- Many decision trees are generated during training.
- Tree bagging is used to train trees (reduces variance and overfitting).
- Feature bagging: Each tree only uses a random subset of features, which ensures that trees remain decorrelated.
- Used for both regression and classification
- Training is easily parallelizable
- Robust to noise and outliers



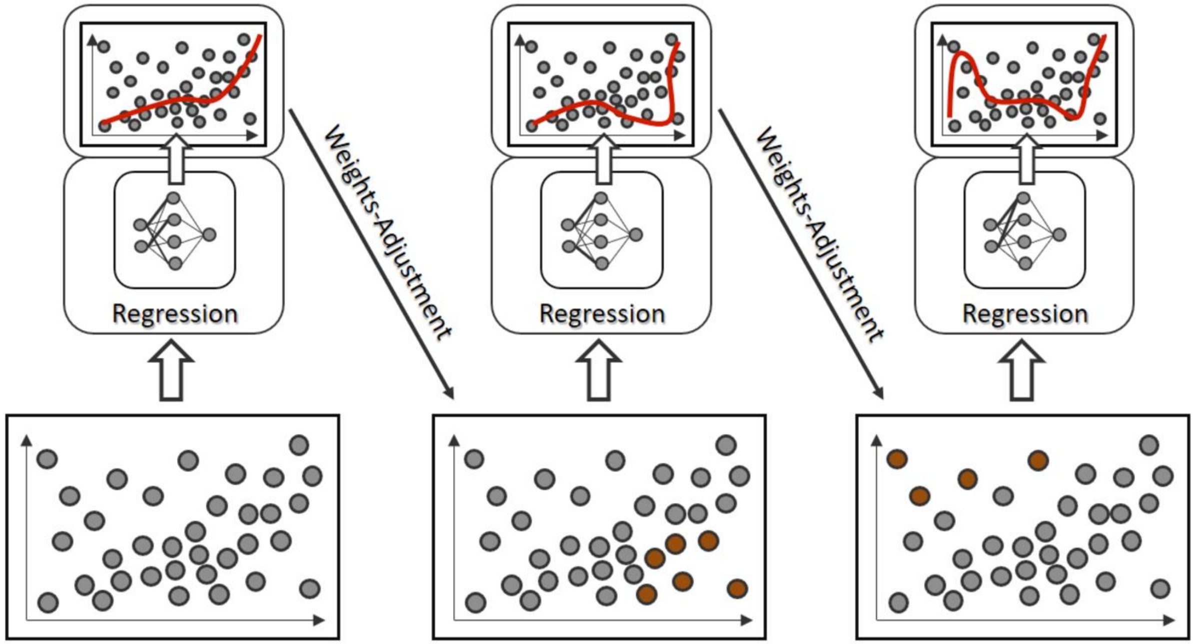
Ensemble methods: Bagging

- Short for bootstrap aggregating
- An ensemble technique where the constituent models are
  - of the same type, but
  - trained on different randomly sampled data subsets.
  - Bootstrapping ⇔ sampling with replacement
  - Final prediction by aggregation (clf: majority vote, reg: or averaging)
- Overall model becomes more robust/reduces overfitting.
- All constituent models can be trained in parallel.



Ensemble methods: Boosting

- An ensemble method in which the models are trained sequentially, with more weight given to misclassified samples.
- Advantage: Performance often higher than bagging
- Disadvantages (compared to bagging):
  - More susceptible to overfitting
  - Not parallelizable
- Popular methods:
  - AdaBoost (focus on misclassified cases)
  - Gradient boosting (focus on residuals)
  - XGBoost (popular instance of GB)



k-nearest neighbors for classification (and also regression)

- kNN uses local information from *nearby* training examples to predict new labels.

Notes:

- It is common to weight neighbors with the inverse of their distance, such that closer points have greater influence:

$$\text{weight} = \frac{1}{\text{distance}}$$

- Weights should also be applied in case of imbalanced data.
- Since feature-space distances combine different units, normalizing the training data is key!

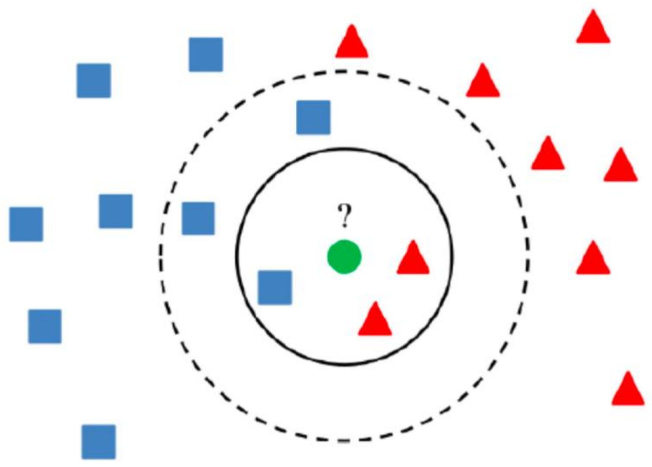


Illustration of kNN for supervised learning: Search for the k nearest neighbors of the **inference point** for:
— k = 3
- - k = 5

Clustering: Method overview

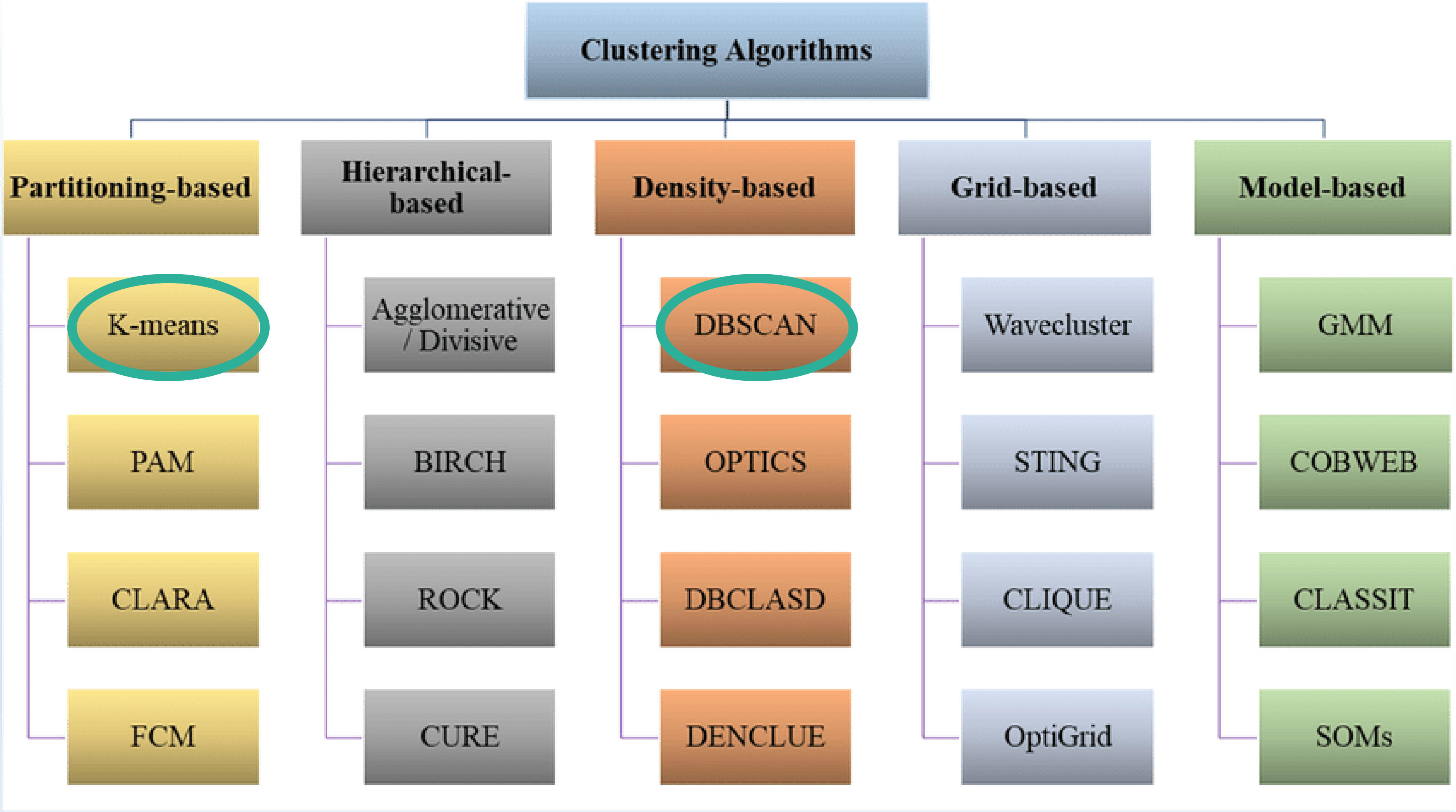
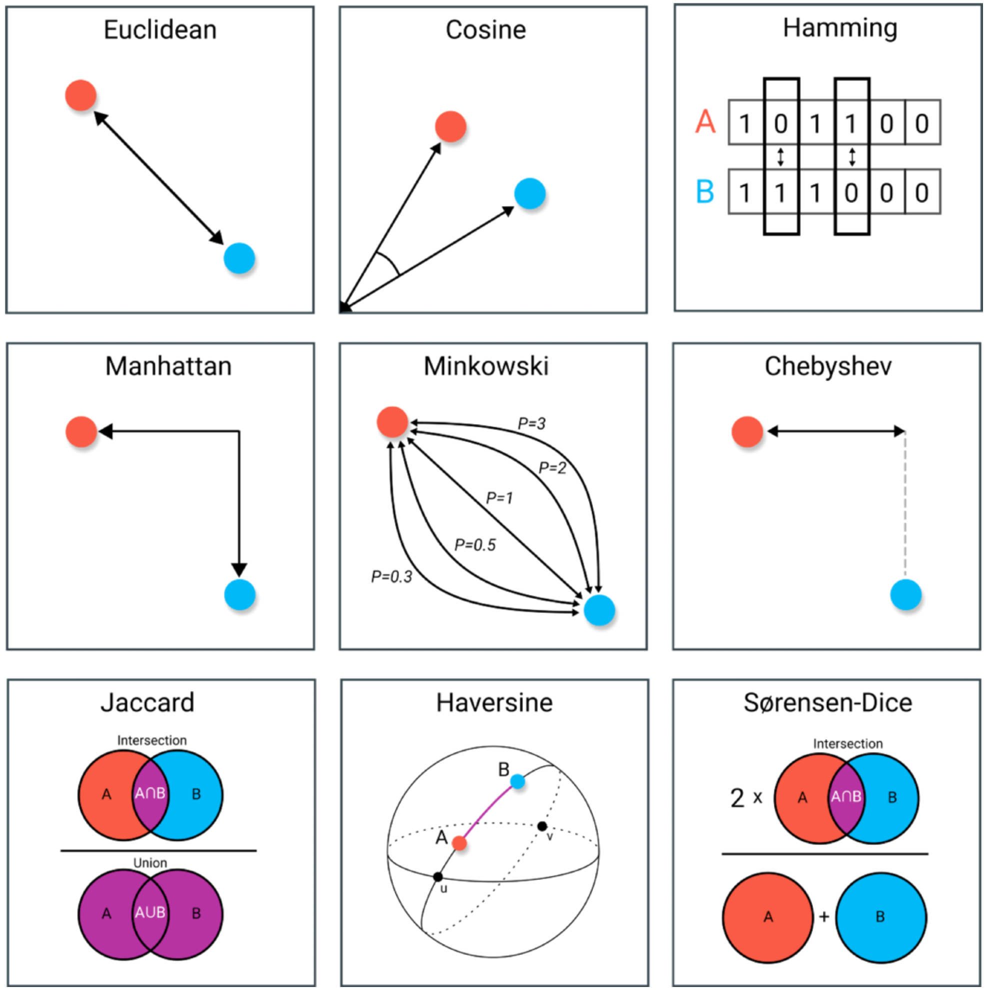


Illustration: Mehta et al, 2020. Link

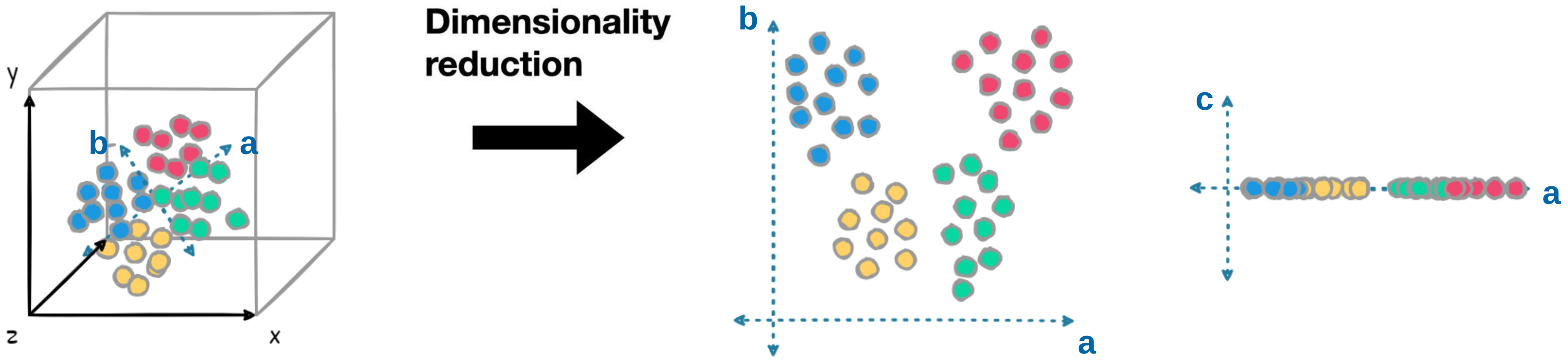
Distance metrics: How to quantify similarity / proximity?

- Euclidean $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Manhattan $\sum_{i=1}^{n} |p_i - q_i|$
- Minkowski $\left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}}$
- Cosine
- Hamming
- ...
- and any application specific distances...



Dimensionality reduction through projection

- Projection-based dimensionality reduction involves mapping high-dimensional data onto a lower-dimensional subspace while preserving as much of the data's structure as possible.



Two possible projections from a 3D space onto a 2D subspace. Here, the first subspace is the plane in which the data points lie (plane with the axes a-b). The second one is the plane a-c, which is perpendicular to the a-b plane.

Source of illustration: Link

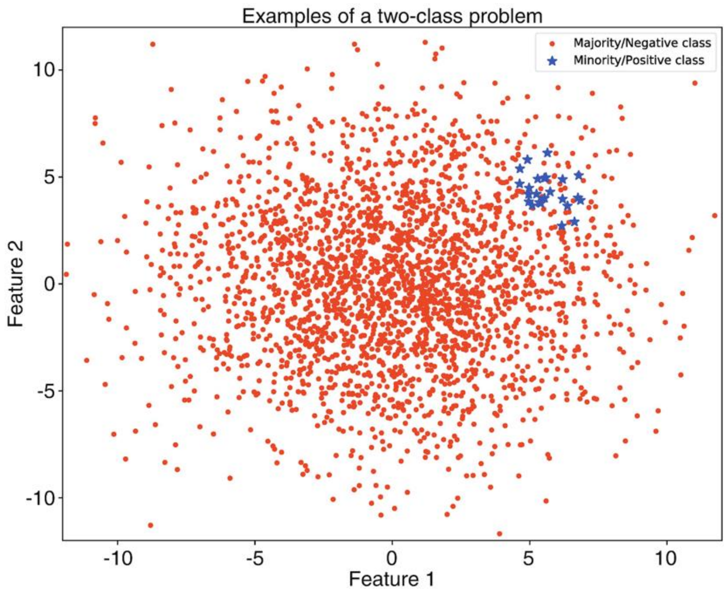
Miscellaneous

Imbalanced data

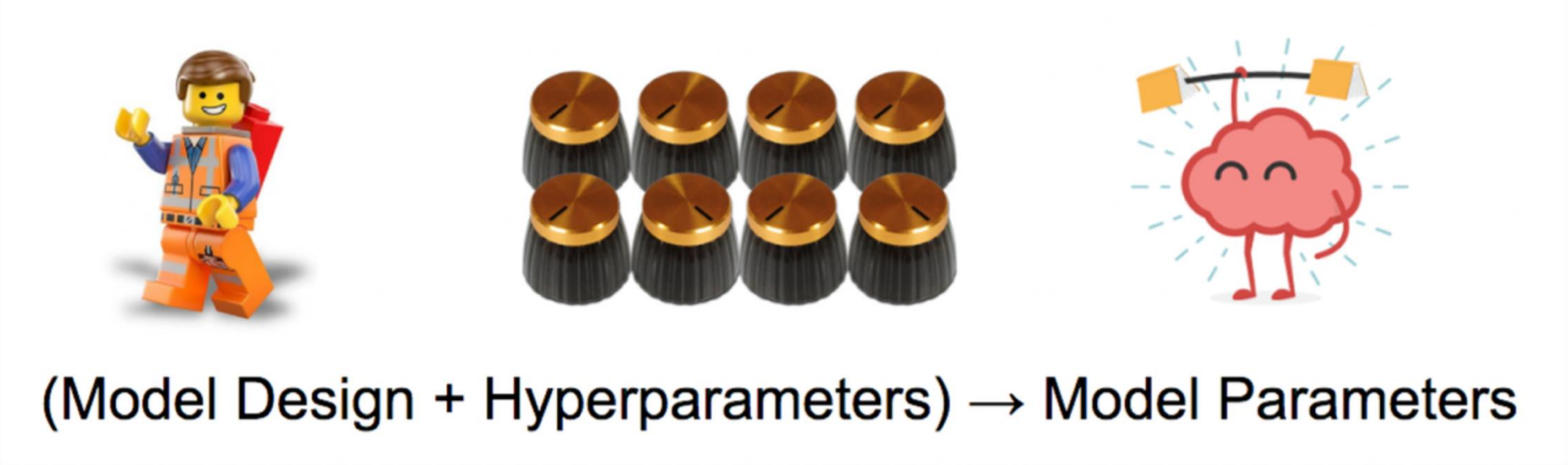
- **Definition**: Imbalanced data refers to a situation where certain categories, values, or groups in a dataset are underrepresented compared to others.

- **Examples**:
  - Credit card fraud
  - Natural disasters
  - Tumor cells
  - ...

- Many (if not most) real-world datasets exhibit imbalance
- Often, we are interested in the outlier (or rare) events



The three levels of model parameters



Examples

Function class (e.g. cosign or a polynomial)

Degree of the polynomial

The polynomial's parameters (a_0, a_1, etc.)

Core ideas of active learning

- **Iterative process**:
  The active learner is trained on an initial labeled dataset, then iteratively selects and requests labels for the most uncertain or impactful samples.

- **Uncertainty sampling**:
  The model identifies data points where it has the least confidence in its predictions (e.g., probabilities close to 0.5 in binary classification).

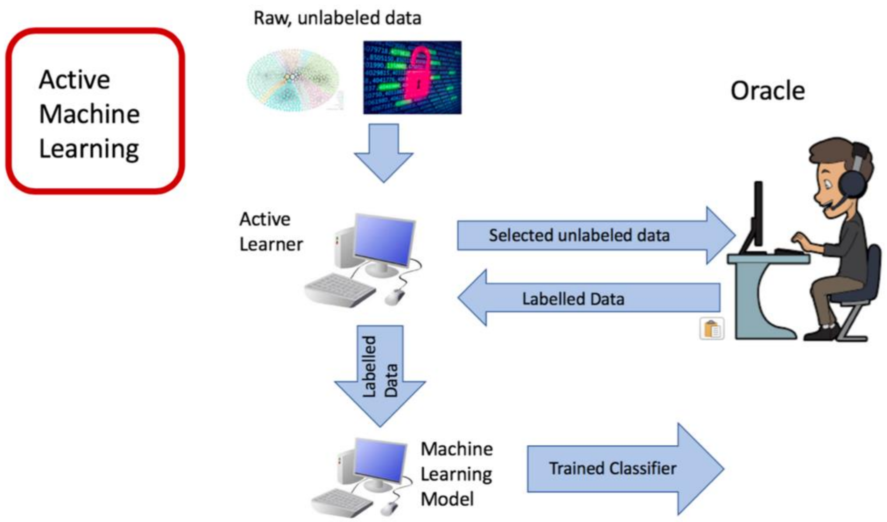
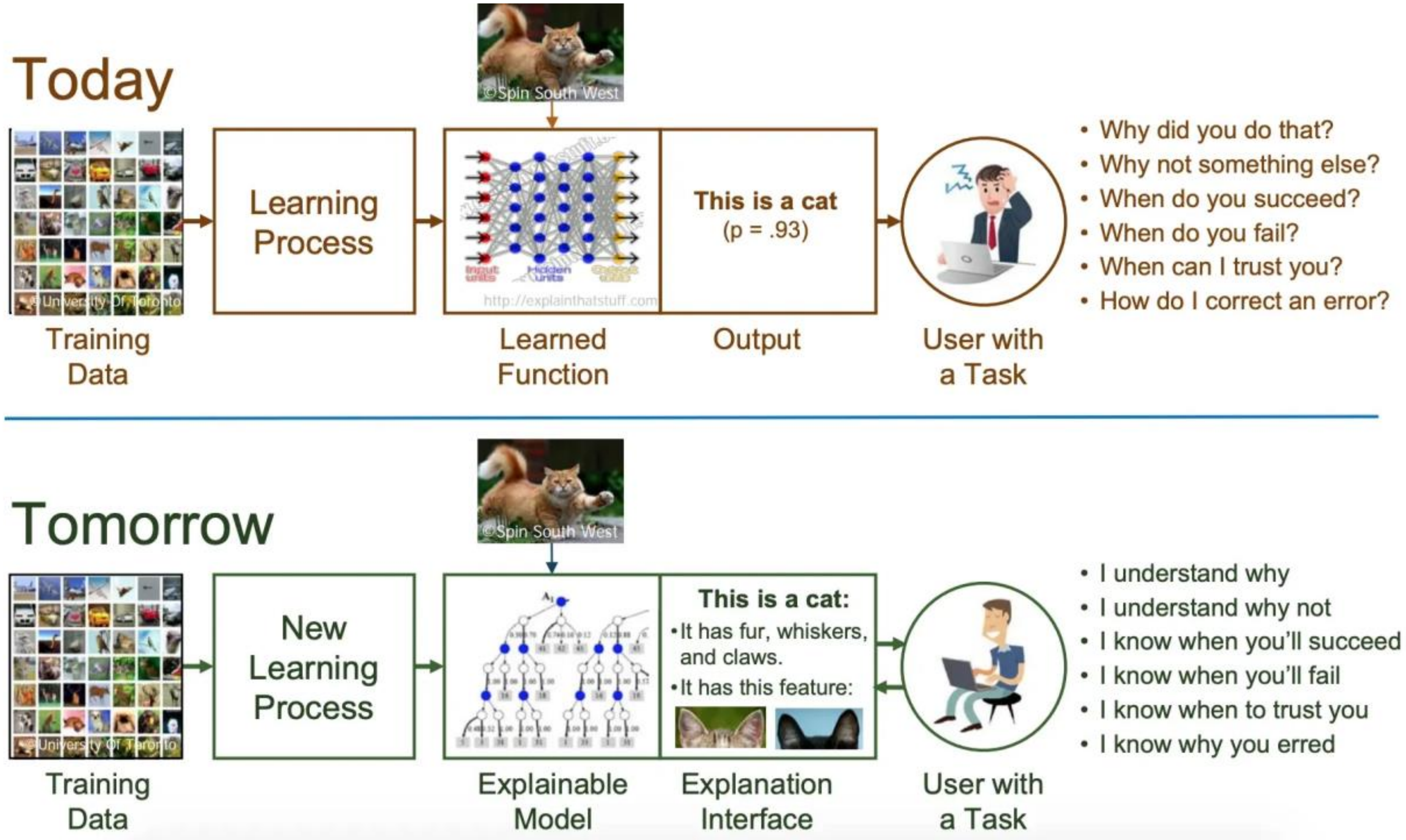


Illustration of active learning. The use of machine learning during a data labeling process. Image source: [Link]

The vision of explainable ML



Source: Could not identify the original source