Computational Analysis of Sound and Music

Machine Learning

Dr.-Ing. Jakob Abeßer

Fraunhofer IDMT

jakob.abesser@idmt.fraunhofer.de

Outline

- **Machine Learning**
    - Introduction
    - Application Scenarios
    - Learning Paradigms
- Machine Learning Pipeline

Machine Learning

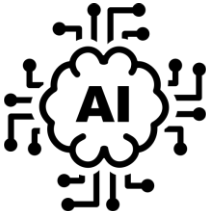
Introduction

- Human intelligence
  - “mental quality that consists of the abilities to **learn** from experience, **adapt** to new situations, **understand** and handle abstract concepts, and use knowledge to **manipulate** one’s environment.” [1]

- Human learning
  - “Learning is the process of acquiring new **understanding**, **knowledge**, **behaviors**, **skills**, **values**, **attitudes**, and **preferences**.”

Machine Learning

Introduction



- Artificial Intelligence
  - Agent (machine)
    - **Perceive** and **react** to environments
    - Performs **actions** to achieve **goals** [2]
- Levels of AI
  - **Narrow/weak AI** (single task, limited context)
    - Examples: Voice assistants, self-driving cars, chat bots
  - **Artificial general intelligence** (AGI)
    - Multiple task
    - Knowledge generalization across tasks

Machine Learning

Introduction

- Machine Learning (ML)
  - Sub-field of AI
  - "...give computers the ability to learn **without being explicitly programmed**" [3]
  - Learning structures in given (un)labeled data to make predictions on new / unseen data
- Paradigm change
  - Before: Use **domain knowledge** to design (general-purpose) features
  - Now: **Learn** suitable **representations** (features) & **models** (classification) **jointly** by analyzing (annotated) **data**

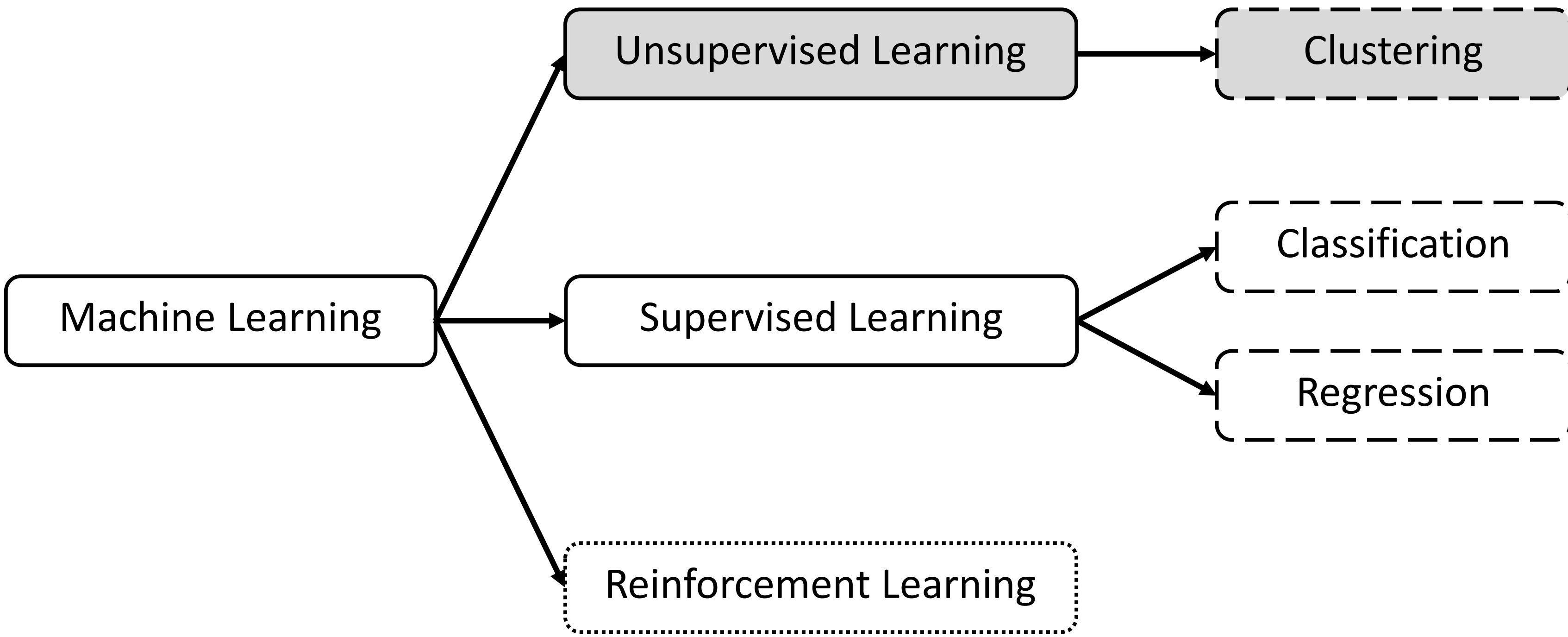
Machine Learning

Application Scenarios

- Computational finance (credit scoring, algorithmic trading)
- Computer vision (face & object recognition, motion detection)
- Computational biology (tumor detection, drug discovery, DNA sequencing)
- Energy (price & load forecasting)
- Predictive maintenance (automotive, aerospace, manufacturing)
- Natural language processing (sentiment classification, text search, translation)
- **Machine listening** (music transcription, instrument recognition, sound event detection, acoustic scene classification)

Machine Learning

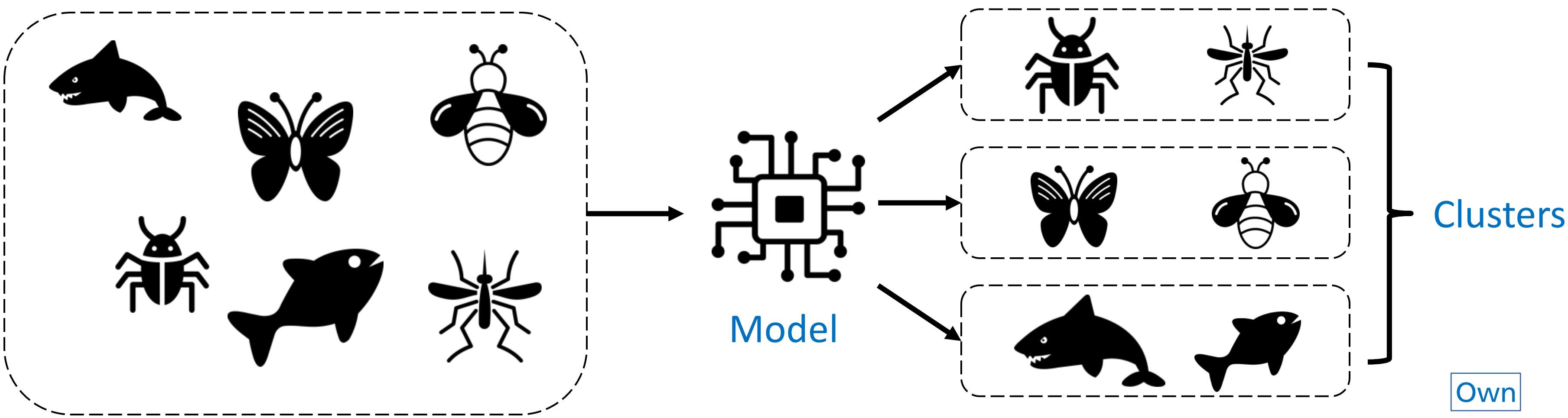
Learning Paradigms



Machine Learning

Unsupervised Learning

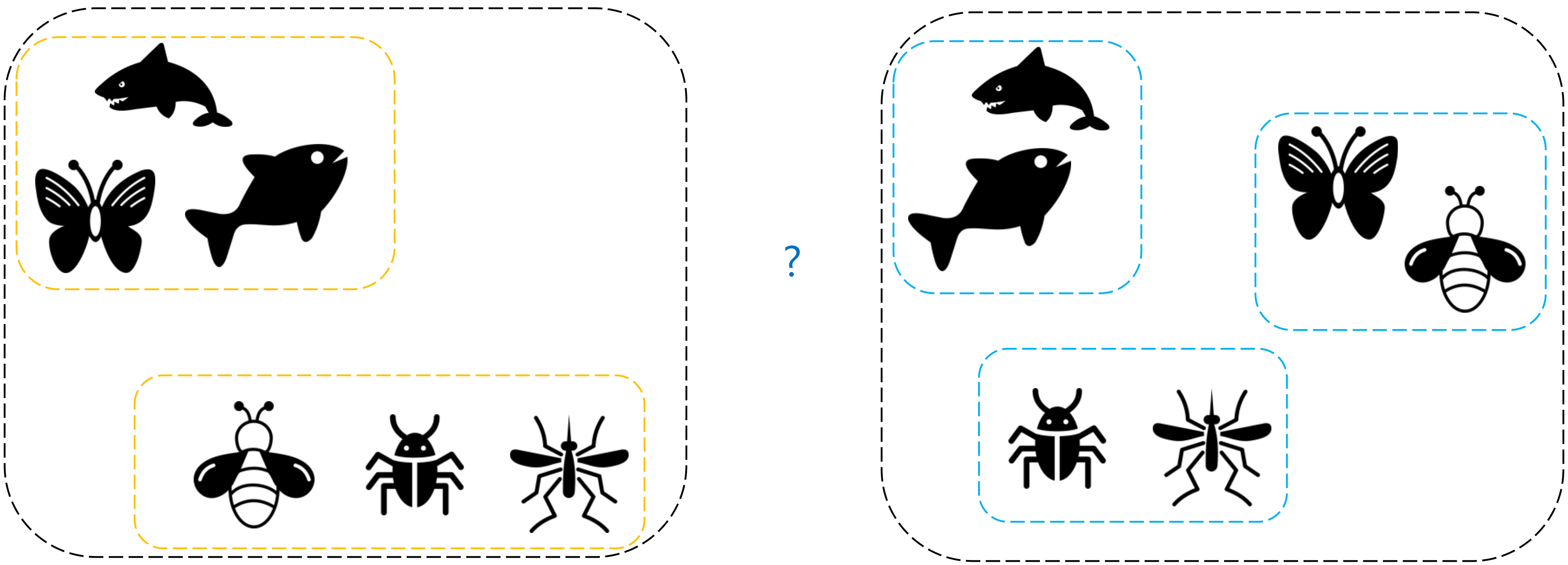
- Goal
  - Find hidden **structure** and **patterns** in data
  - **No annotations** available
- Clustering
  - **Grouping** of **similar** data instances



Machine Learning

Unsupervised Learning

- Challenges
  - What is the **optimal number of clusters**?



2 clusters

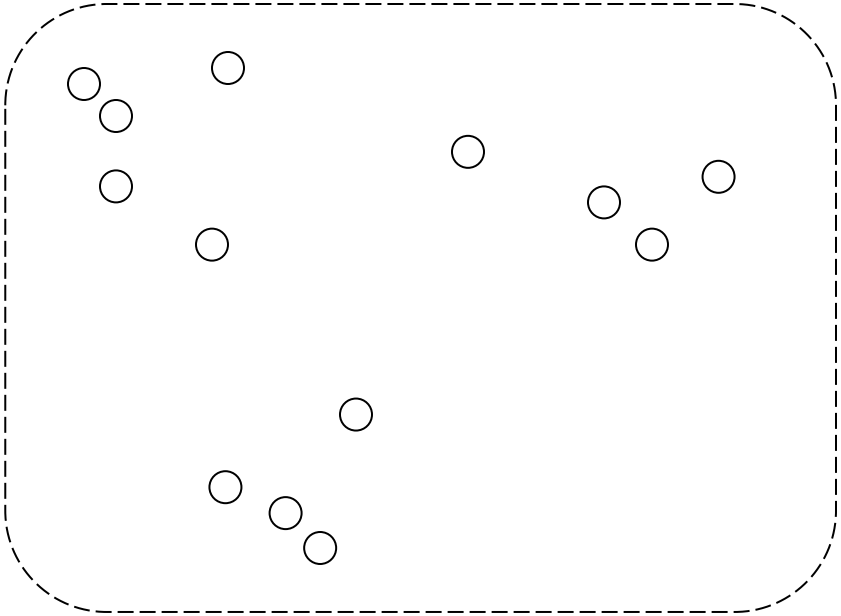
3 clusters

Own

Machine Learning

Unsupervised Learning

- k-means clustering

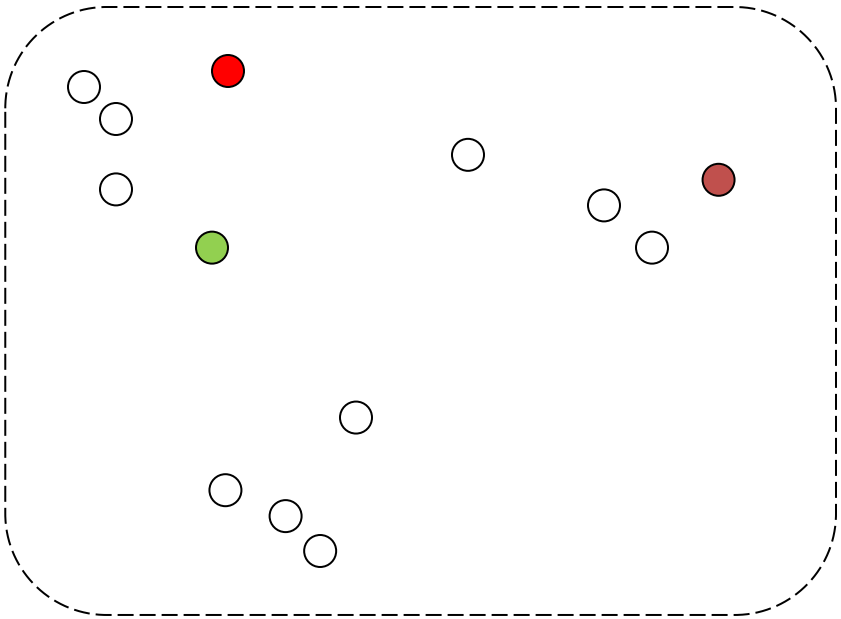


Own

Machine Learning

Unsupervised Learning

- k-means clustering
    - Initialize k randomly (here: $k = 3$) → number of clusters

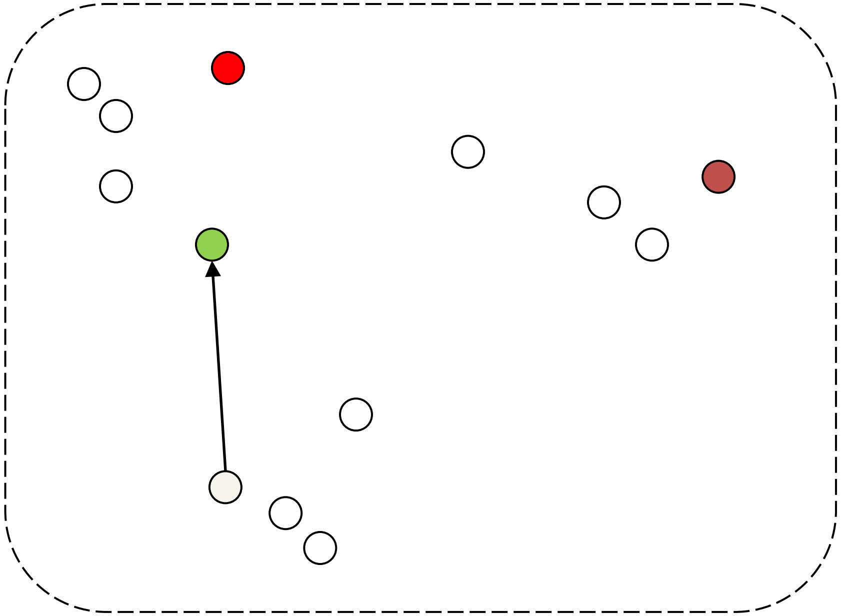


Own

Machine Learning

Unsupervised Learning

- k-means clustering
  - Assignment: assign each data point to its closest mean

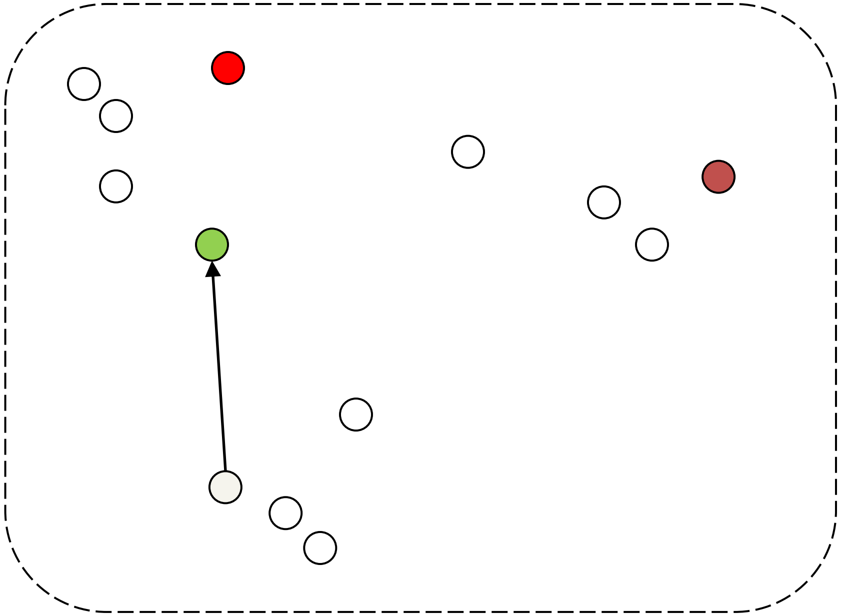


Own

Machine Learning

Unsupervised Learning

- k-means clustering
    - Assignment: assign each data point to its closest mean

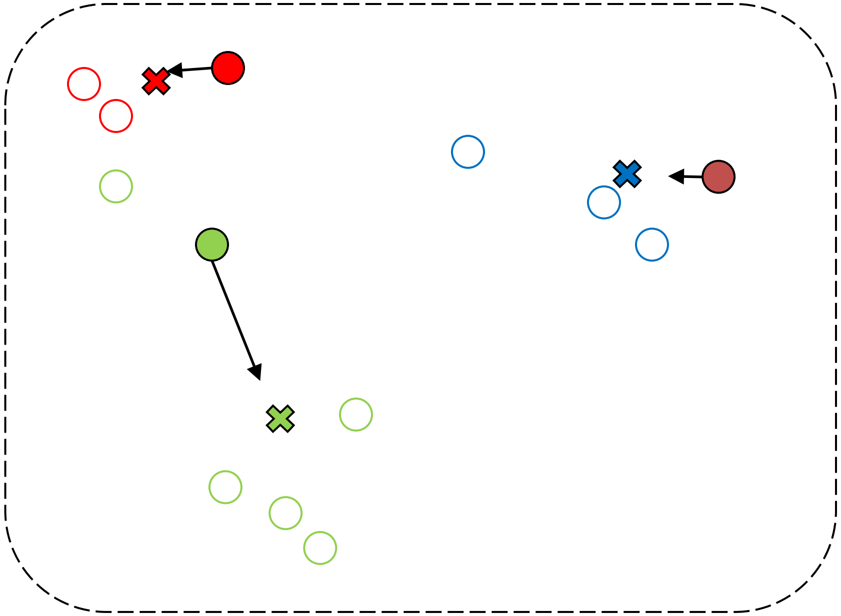


Own

Machine Learning

Unsupervised Learning

- k-means clustering
    - Update: update mean by average over all assigned data points

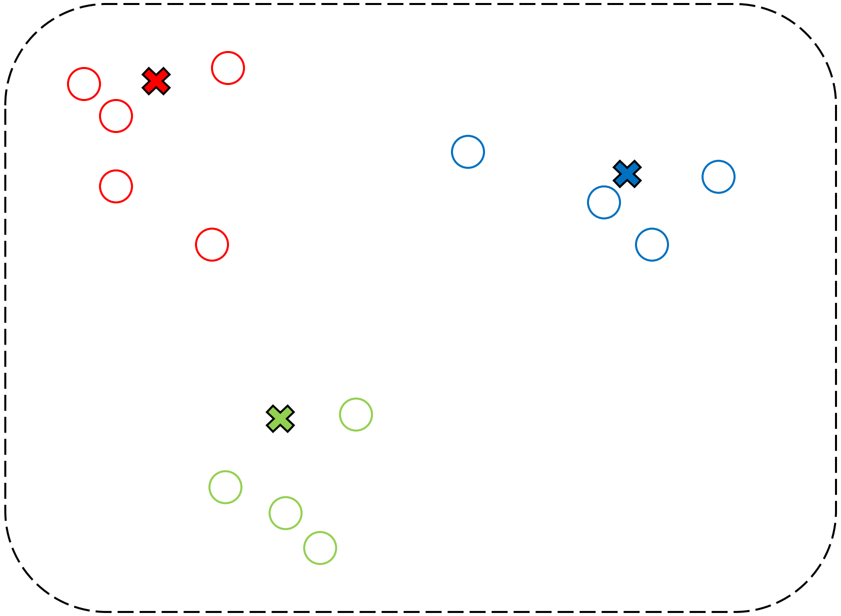


Own

Machine Learning

Unsupervised Learning

- k-means clustering
  - Assignment: re-assign data points to closest mean

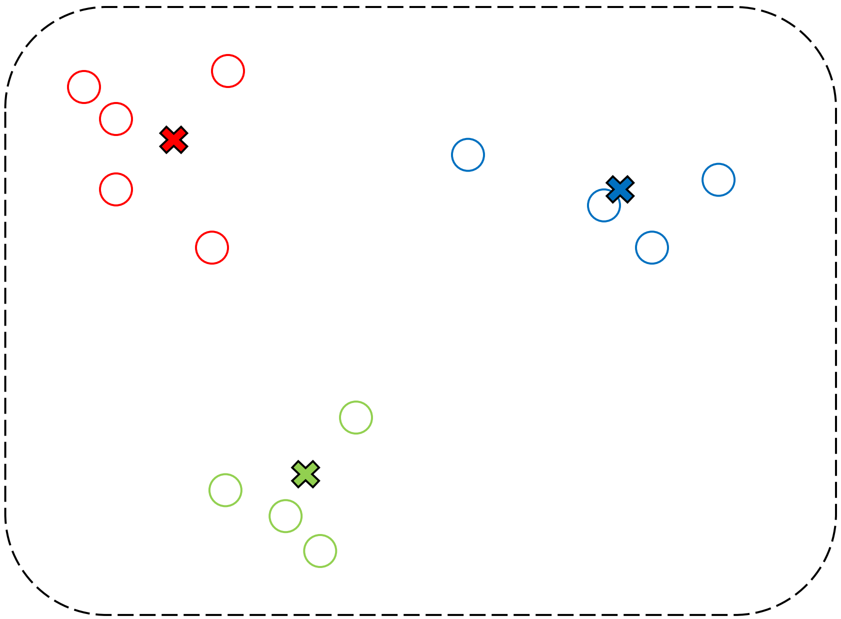


Own

Machine Learning

Unsupervised Learning

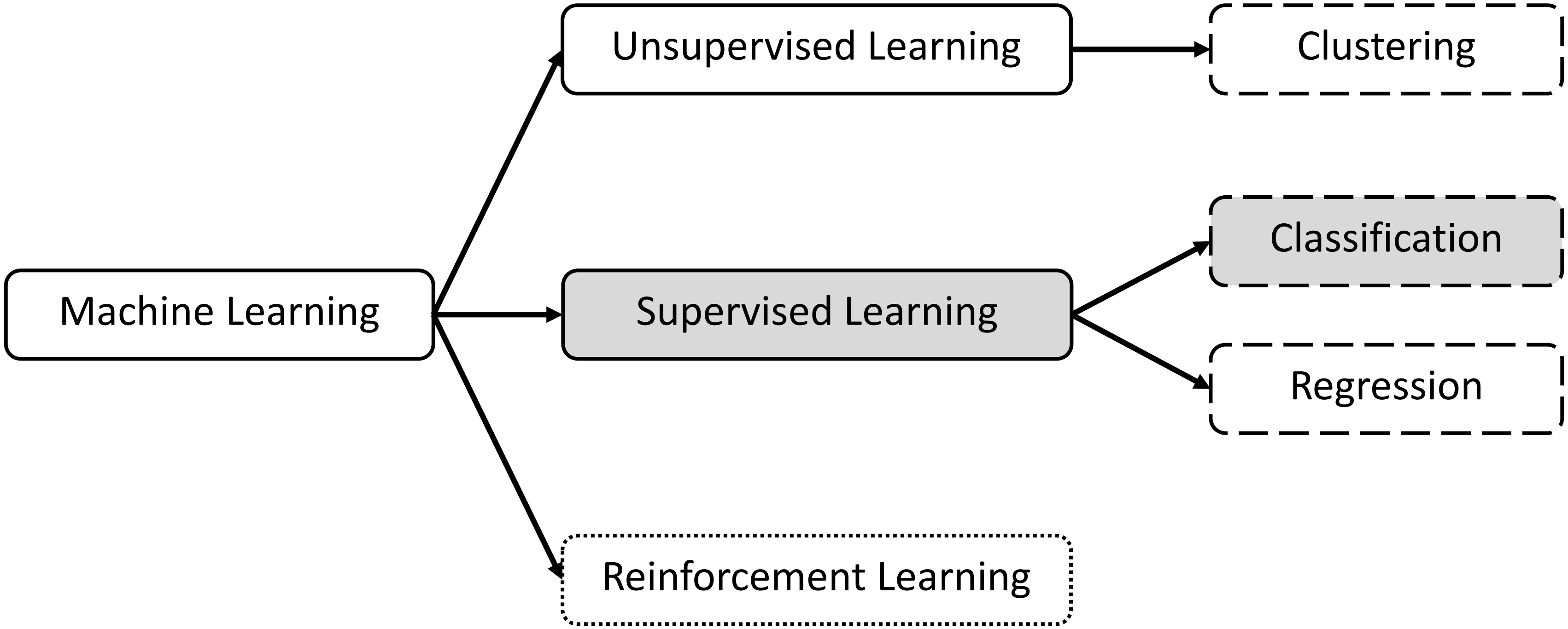
- k-means clustering
  - Repeat until convergence



Own

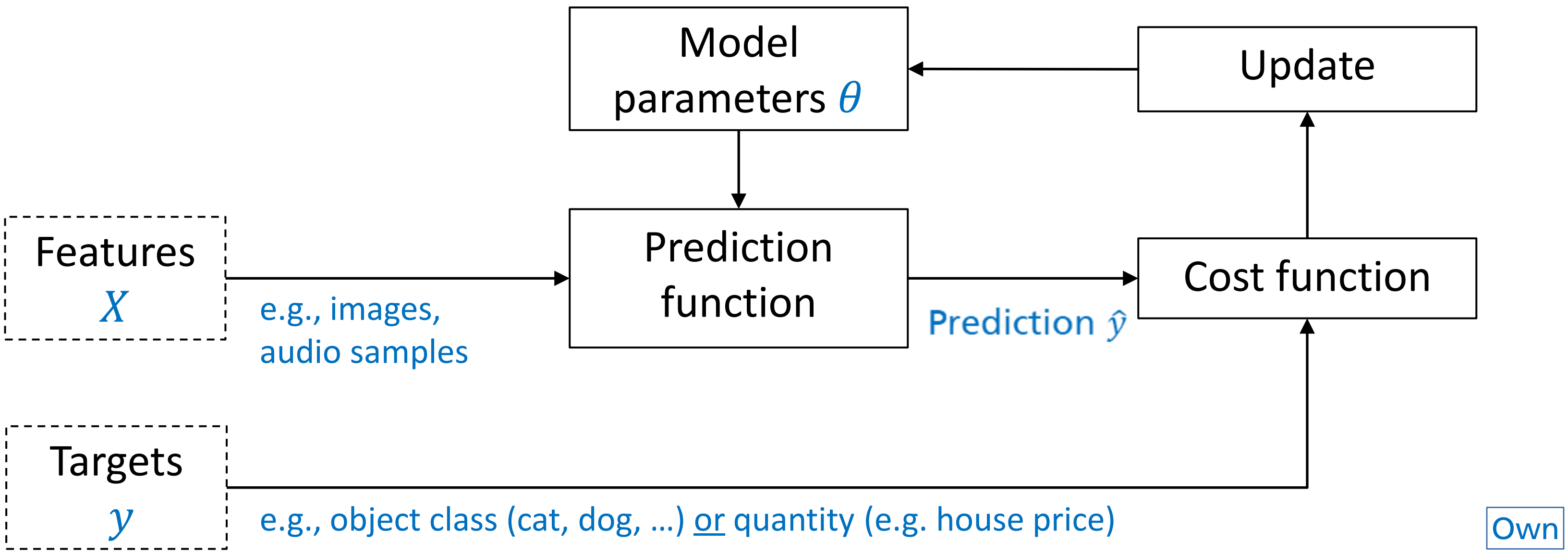
Machine Learning

Learning Paradigms



Machine Learning

Supervised Learning



Machine Learning

Supervised Learning (Classification)

- Predict one or multiple **categorical** labels from features
    - Examples → music genre, instrument(s), key
- **Feature space** (Example: 2 classes)

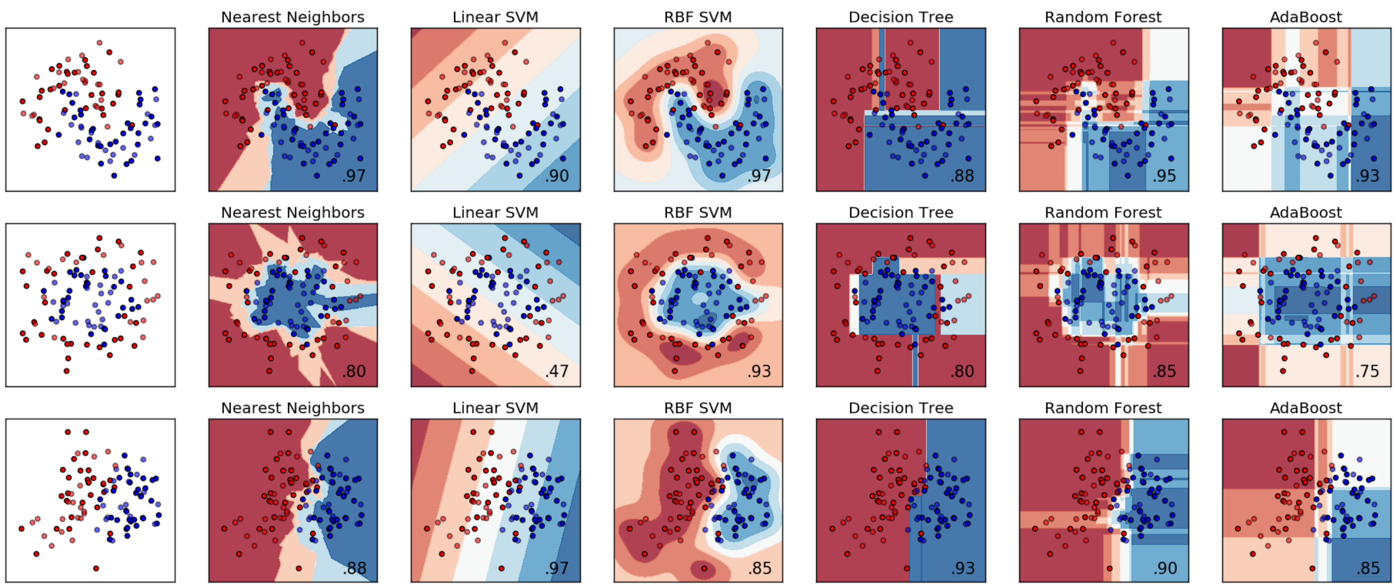
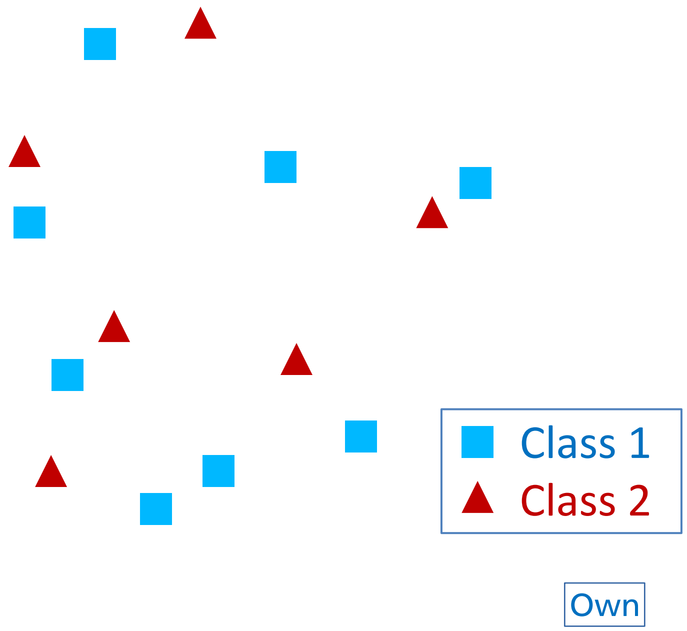


Fig-D1-1

Machine Learning

Supervised Learning (Classification)

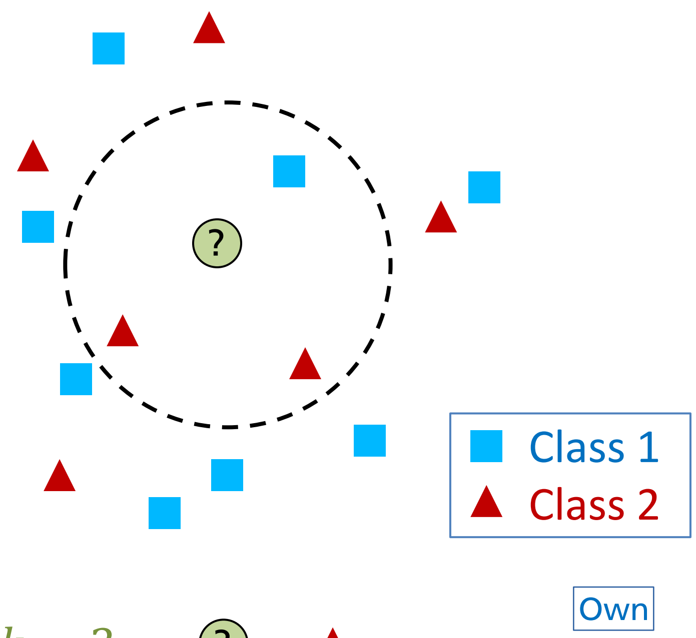
- k-nearest neighbors classifier
    - Training → Store all examples



Machine Learning

Supervised Learning (Classification)

- k-nearest neighbors classifier
  - Training → Store all examples
  - Test → Assign test item to dominant class label of the k closest training data items

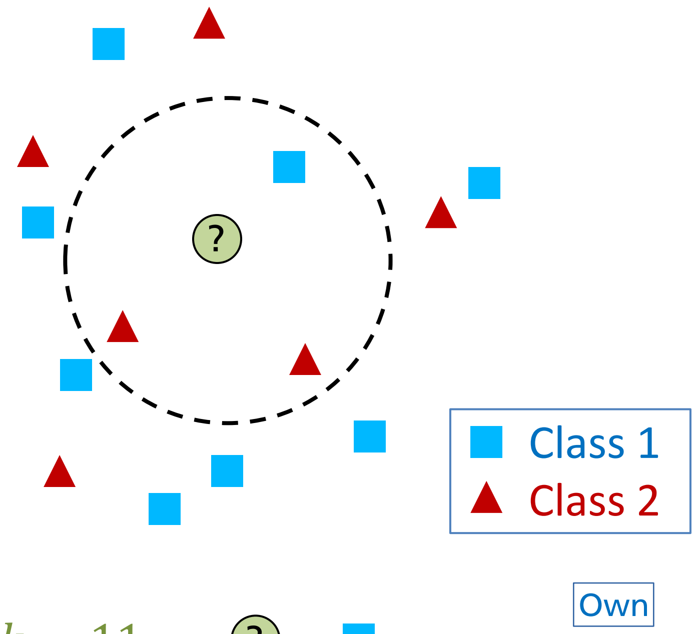


$k = 3 \rightarrow$? = ▲

Machine Learning

Supervised Learning (Classification)

- k-nearest neighbors classifier
    - Training → Store all examples
    - Test → Assign test item to dominant class label of the k closest training data items

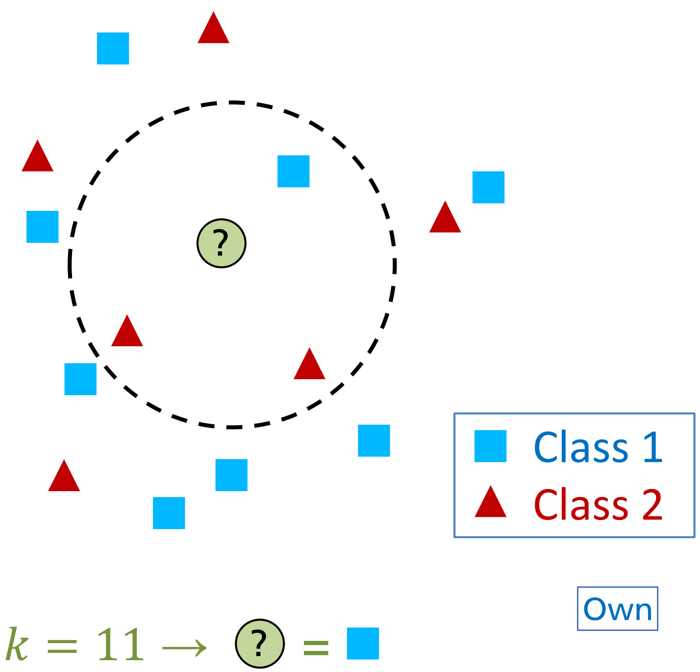


$k = 11 \rightarrow$? =

Machine Learning

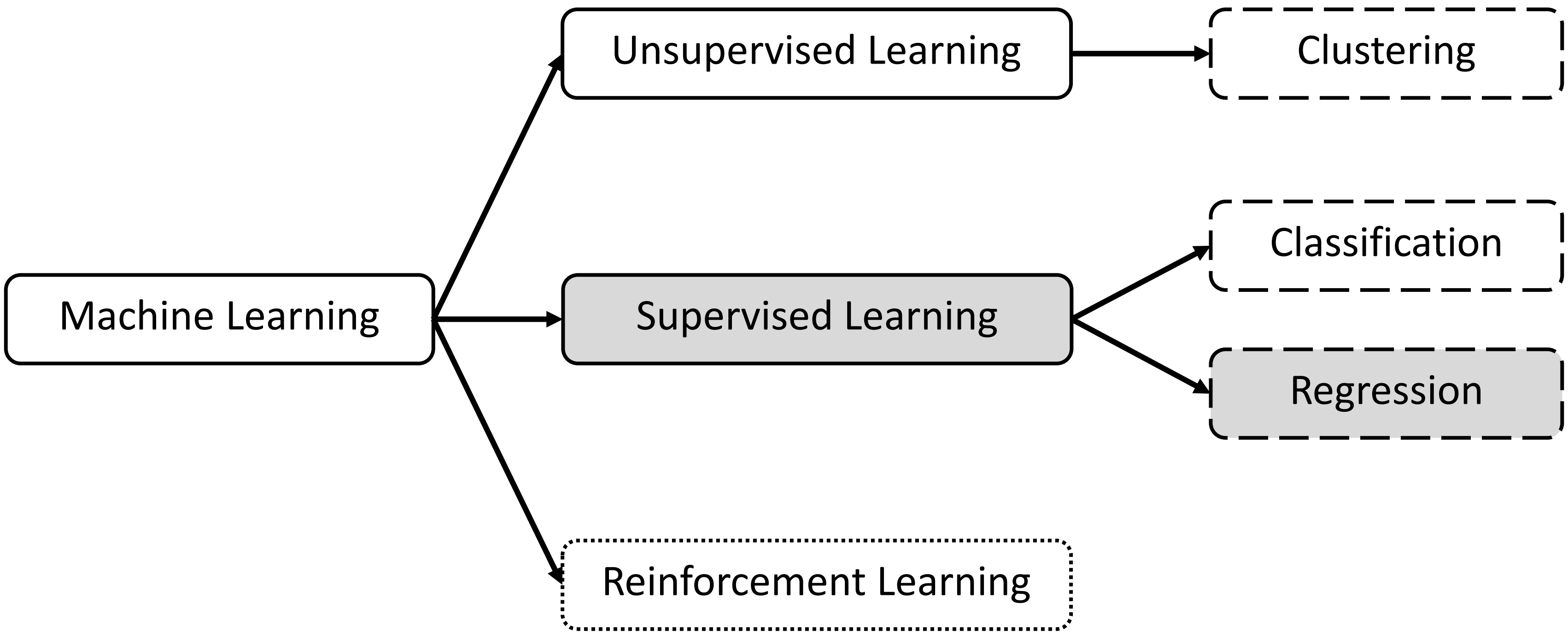
Supervised Learning (Classification)

- k-nearest neighbors classifier
    - Training → Store all examples
    - Test → Assign test item to dominant class label of the k closest training data items
    - Distance measures
        - Euclidean distance
        - Cosine distance, …



Machine Learning

Learning Paradigms



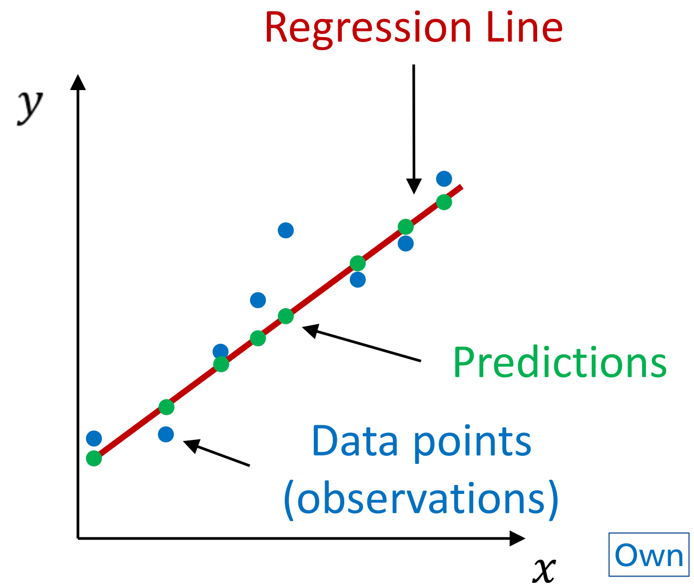
Machine Learning

Supervised Learning (Regression)

- Goal
    - Predict a **dependent** (response) **variable** given one or multiple **independent variables** (features)
    - **Continuous** quantities
- Examples
    - Univariate (linear) regression

$$y \sim \beta_0 + \beta_1 \cdot x$$

        - β_0 (bias)
        - β_1 (weight)

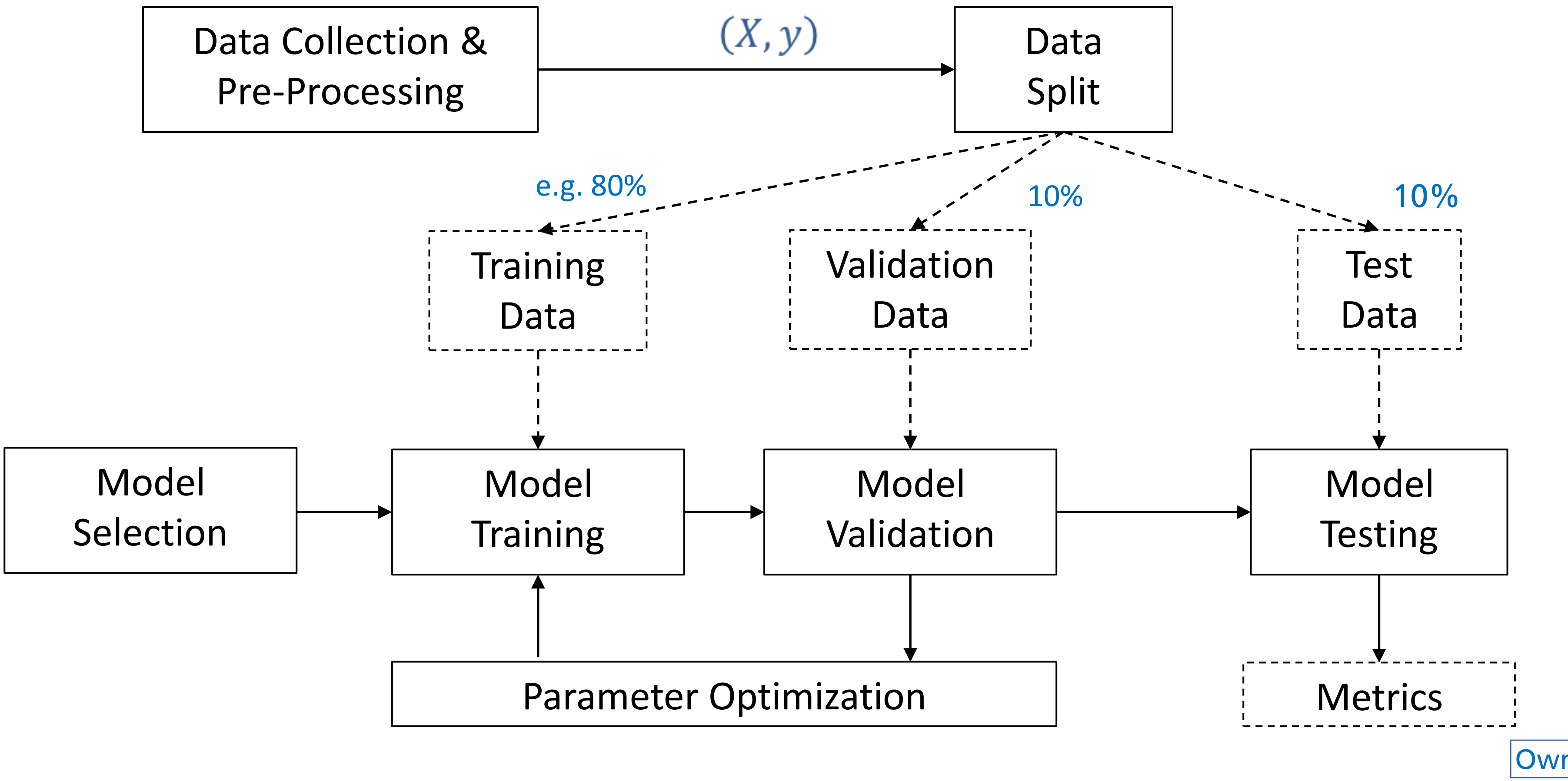


Outline

- Machine Learning
    - Introduction
    - Application Scenarios
    - Learning Paradigms
- **Machine Learning Pipeline**

Machine Learning

Machine Learning Pipeline



Machine Learning

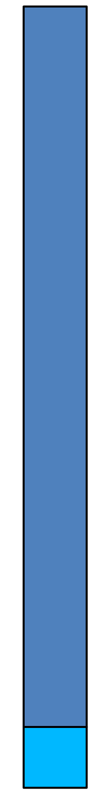
Machine Learning Pipeline (Data Split)

- Training Set
  - Model learns from this data

Machine Learning

Machine Learning Pipeline (Data Split)

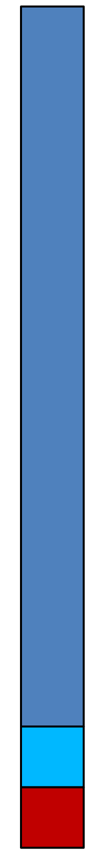
- Training Set
  - Model learns from this data
- Validation / Development Set
  - Used to fine-tune the model (hyper)parameters
  - Model occasionally sees but does not learn from this data



Machine Learning

Machine Learning Pipeline (Data Split)

- Training Set
    - Model learns from this data
- Validation / Development Set
    - Used to fine-tune the model (hyper)parameters
    - Model occasionally sees but does not learn from this data
- Test set
    - Only used once after the model training & tuning is completed
    - Should reflect the targeted real-world use case for the model
- Common split ratios
    - 80/10/10% or even 98/1/1% (for large datasets)



Machine Learning

Machine Learning Pipeline (Data Collection & Pre-Processing)

- Data collection
    - Check for **available data resources** for given (or related) task
    - Collect / record / annotate new data
    - Ensure data **variability**
        - Example (from acoustic condition monitoring) → include different motor engine types & conditions, recording locations, microphones, …
- Data cleanup / pre-processing
    - Remove errors, silence, empty files, …
    - **Balance** dataset (proportions among class examples)
    - **Normalize** (depends on the model)

Machine Learning

Machine Learning Pipeline (Model Selection)

- Many models and approaches exist
    - Types (SVM, GMM, logistic regression, DNNs)
    - Hyperparameters (SVM kernel functions, DNN layer types)
- Often constrained by the use-case / task
    - Model complexity (memory, training time, training data amount)
- Feature pre-processing depends on model type
- **Use simple models for simple tasks**

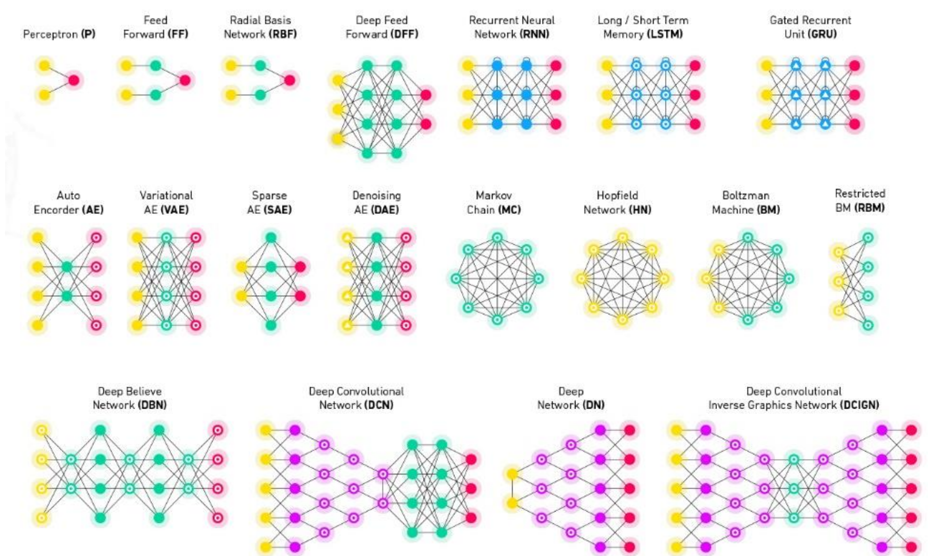


Fig-D1-2

Machine Learning

Machine Learning Pipeline (Model Training)

- Iterative process
  - (Random) parameter initialization
  - Use (**batches** of) training data to iteratively improve model predictions (optimization)
    - Learn from examples
  - Update model parameters according to loss function
  - Monitor improved performance
    - Repeat until convergence

Machine Learning

Machine Learning Pipeline (Model Validation)

- Regular model evaluation each or multiple training iteration
    - Optimize model (hyper)parameters
    - Detect overfitting on training data
    - Stop the training

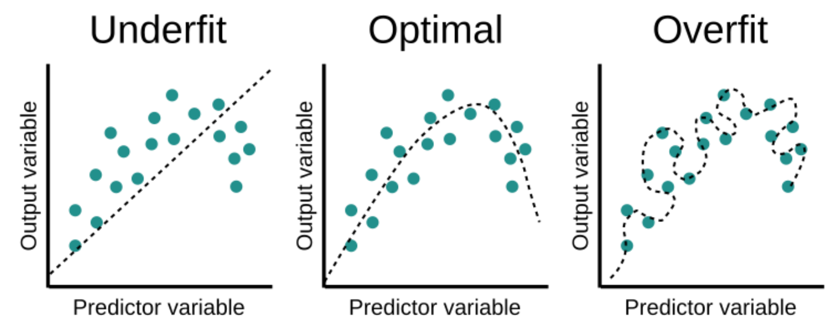
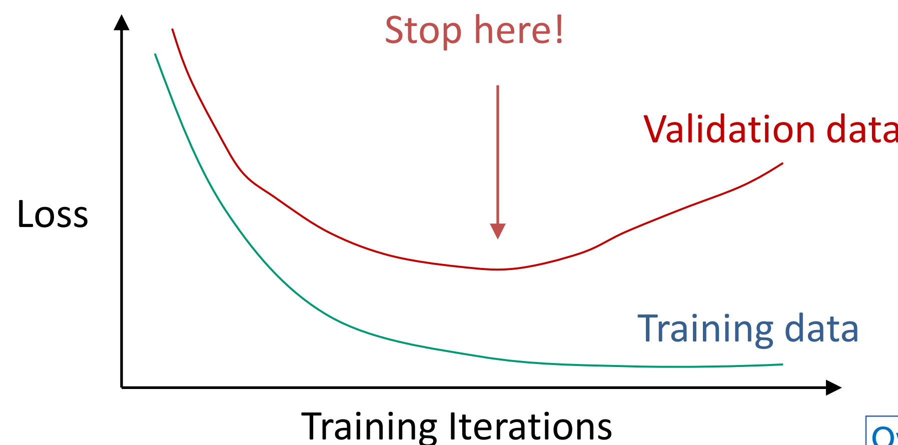


Fig-D1-3



Own

Machine Learning

Machine Learning Pipeline (Model Testing)

- Example: Binary classification evaluation
  - True/false positives (TP/FP)
  - True/false negatives (TN/FN)
  - Metrics
    - Precision
    - Recall
    - Accuracy
    - F-score

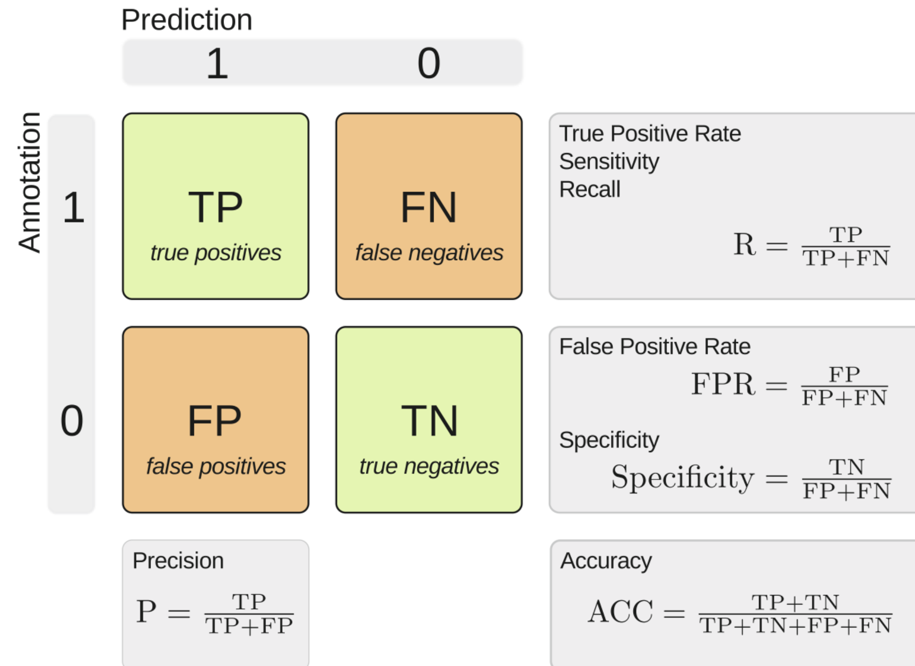


Fig-D1-4

Programming session

Fig-A2-13

References

Images

Fig-D1-1: https://i.stack.imgur.com/hsilO.png (https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)

Fig-D1-2: Neural Network Zoo (https://www.asimovinstitute.org/wp-content/uploads/2019/04/NeuralNetworkZoo20042019.png)

Fig-D1-3: https://www.educative.io/api/edpresso/shot/6668977167138816/image/5033807687188480

Fig-D1-4: Virtanen, T., Plumbley, M. D., & Ellis, D. (Eds.). (2018). Computational Analysis of Sound Scenes and Events. Cham, Switzerland: Springer International Publishing, p. 170, Fig. 6.7

References

References

[1] Introducing Machine Learning. (2016). Retrieved from https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/i/88174_92991v00_machine_learning_section1_ebook.pdf

[2] S. Legg, M. Hutter (2007). Universal Intelligence: A Definition of Machine Intelligence. Minds & Machines. 17 (4): 391-444.

[3] L. Samuel (1959). Some studies in machine learning using the game of checkers. IBM Journal of research and development. 3(3), 210-229