AI-based Audio Analysis of Music and Soundscapes

Machine Learning

Dr.-Ing. Jakob Abeßer
Fraunhofer IDMT

jakob.abesser@idmt.fraunhofer.de

Machine Learning
Outline

- Introduction & Definitions
- Learning Paradigms
- Common ML Pipeline

Machine Learning
Introduction

- Human intelligence

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]

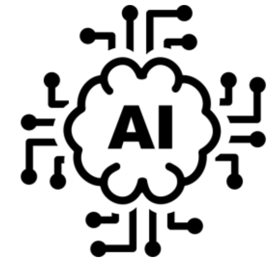
Machine Learning
Introduction

- 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

Machine Learning
Introduction

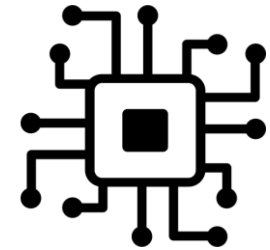


- Artificial Intelligence
  - Agent (machine)
    - **Perceive** and **react to** environments
    - Performs **actions** to achieve **goals** [3][4]

Machine Learning
Introduction

- Artificial Intelligence
    - Agent (machine)
        - **Perceive** and **react to** environments
        - Performs **actions** to achieve **goals** [3][4]
- 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**“ [5]
  - Learning structures in given (un)labeled data to make predictions on new / unseen data

Machine Learning
Introduction

- Machine Learning (ML)
    - Sub-field of AI
    - "...give computers the ability to learn **without being explicitly programmed**" [5]
    - 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)

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)

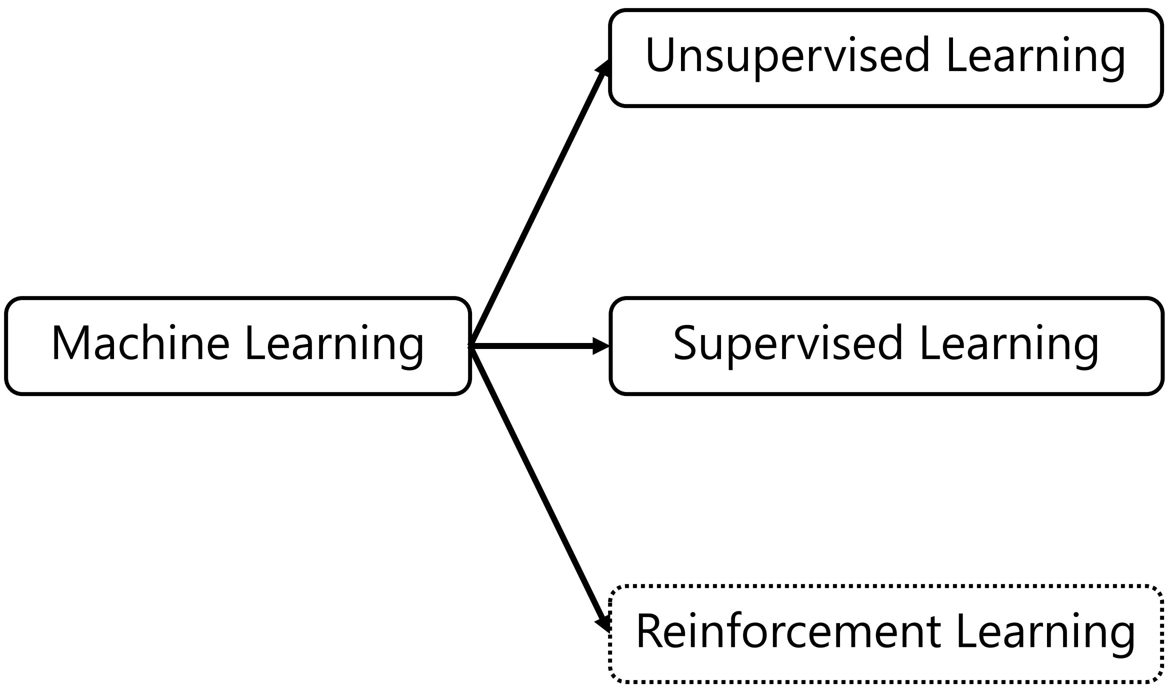
Machine Learning
Application Scenarios

- 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)

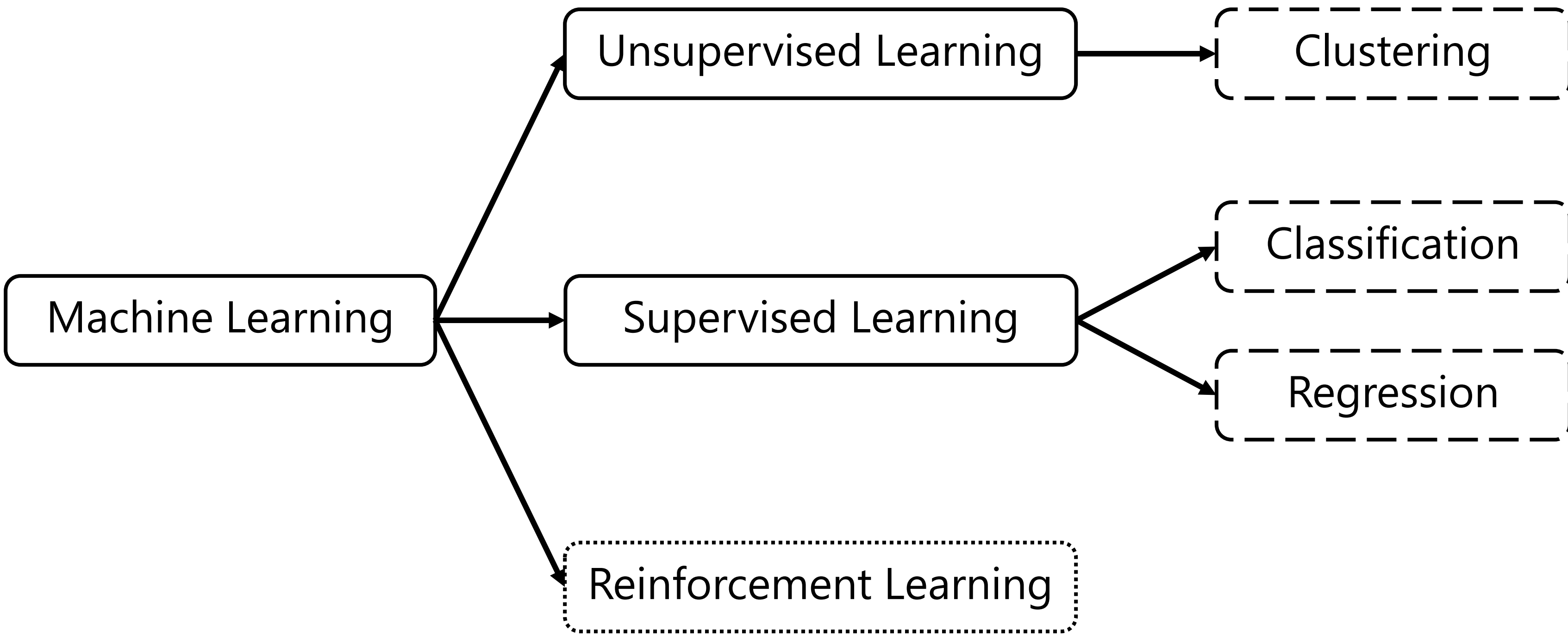
Machine Learning
Learning Paradigms

Machine Learning

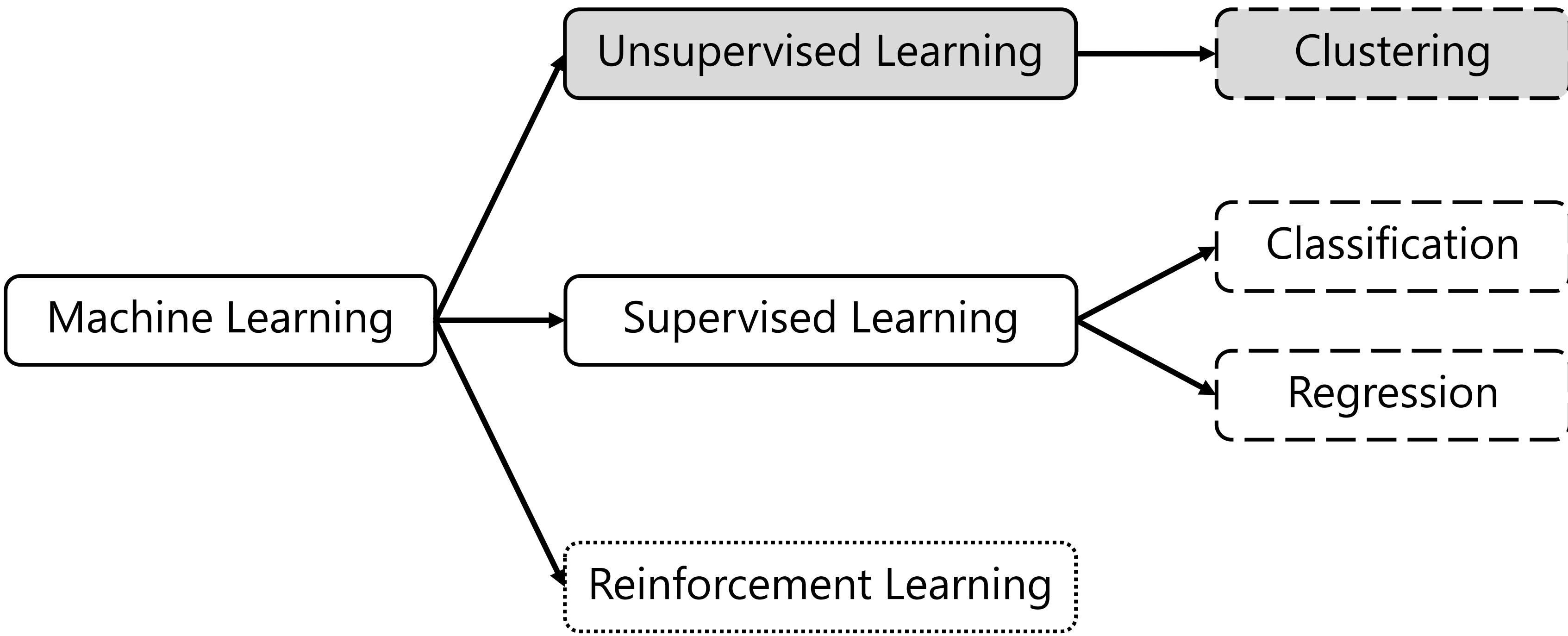
Machine Learning
Learning Paradigms



Machine Learning
Learning Paradigms



Machine Learning
Learning Paradigms

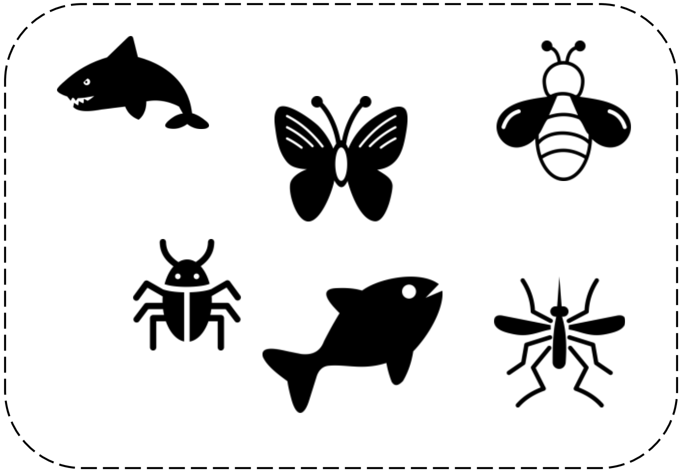


Machine Learning
Unsupervised Learning

- Goal
  - Find hidden **structure** and **patterns** in data
  - **No annotations** available

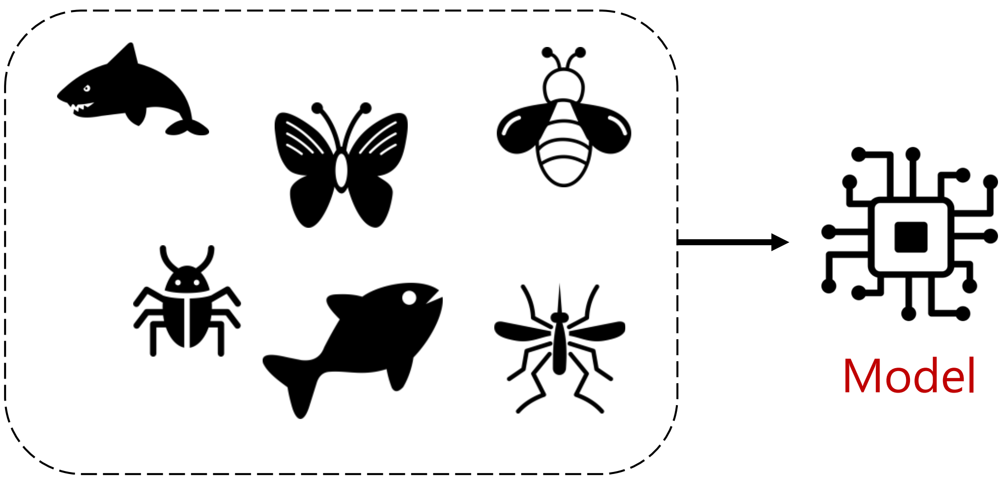
Machine Learning
Unsupervised Learning

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



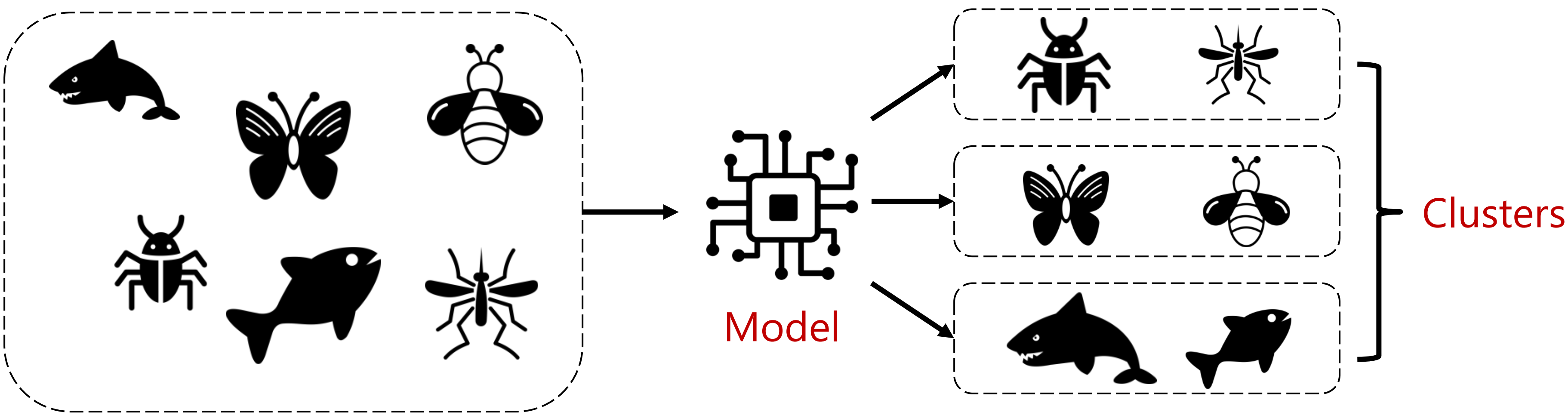
Machine Learning
Unsupervised Learning

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



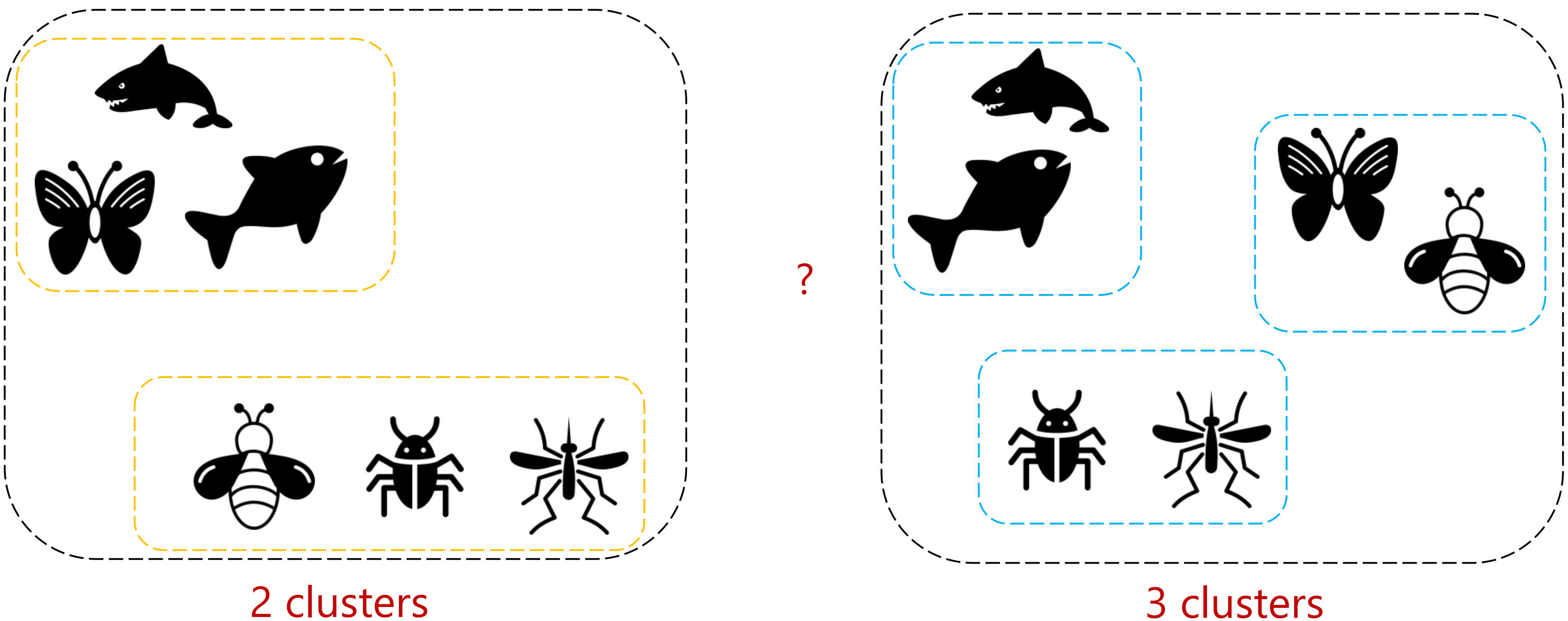
Machine Learning
Unsupervised Learning

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



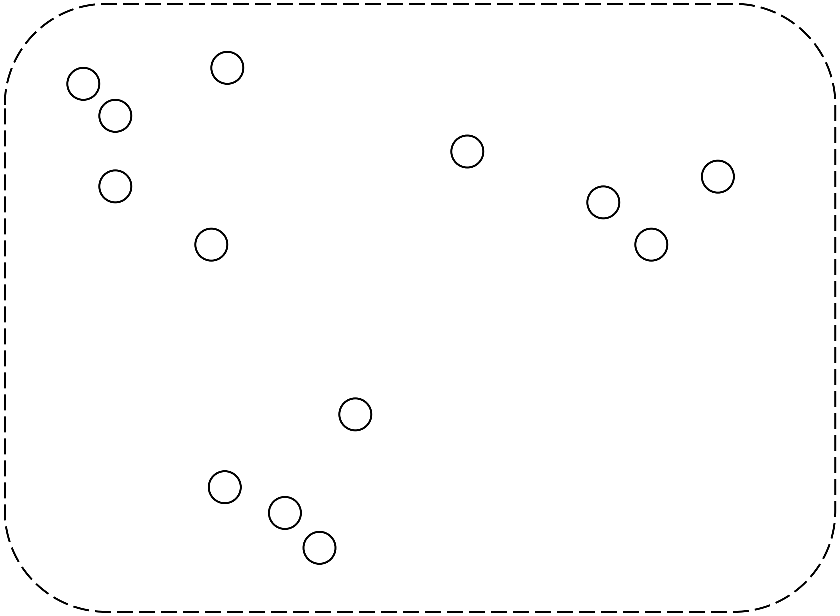
Machine Learning
Unsupervised Learning

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



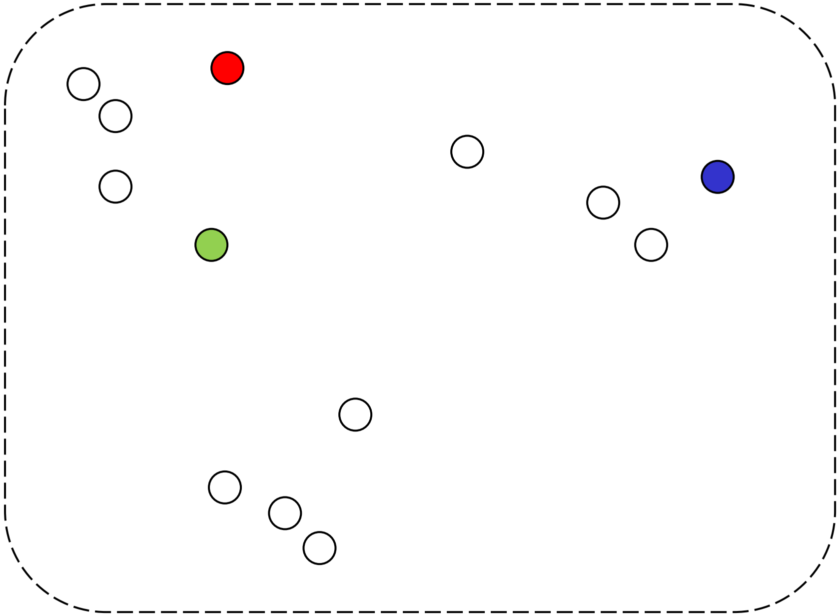
Machine Learning
Unsupervised Learning

- *K*-means clustering
  - Initialize *K* "means" randomly (=cluster centroids)



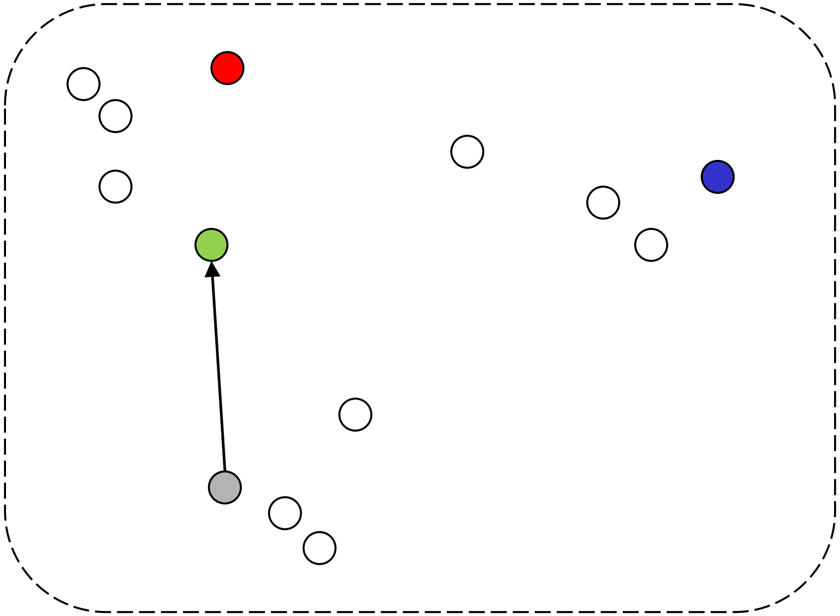
Machine Learning
Unsupervised Learning

- *K*-means clustering
    - K=3



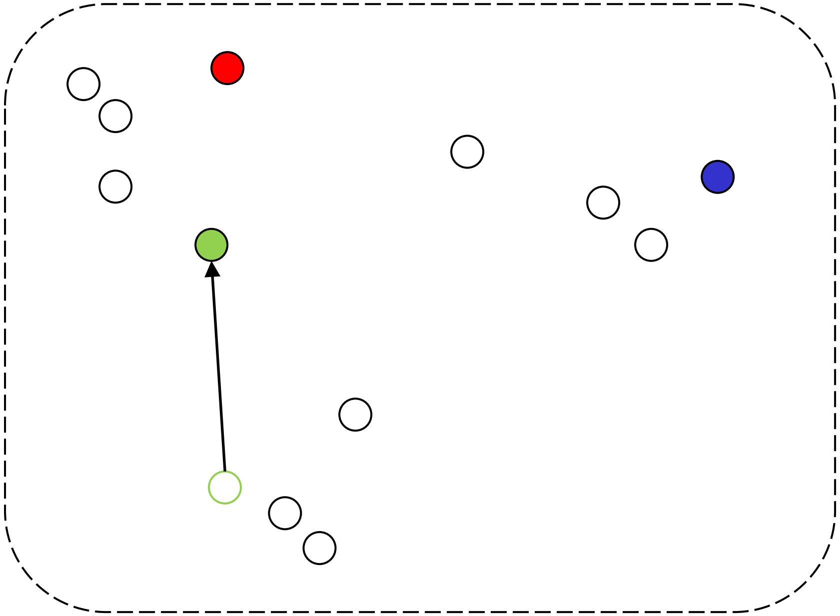
Machine Learning
Unsupervised Learning

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



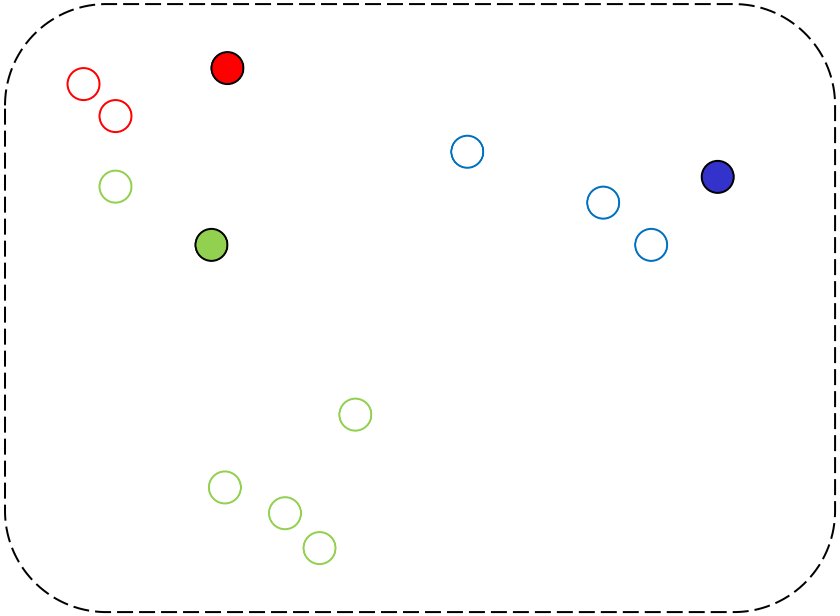
Machine Learning
Unsupervised Learning

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



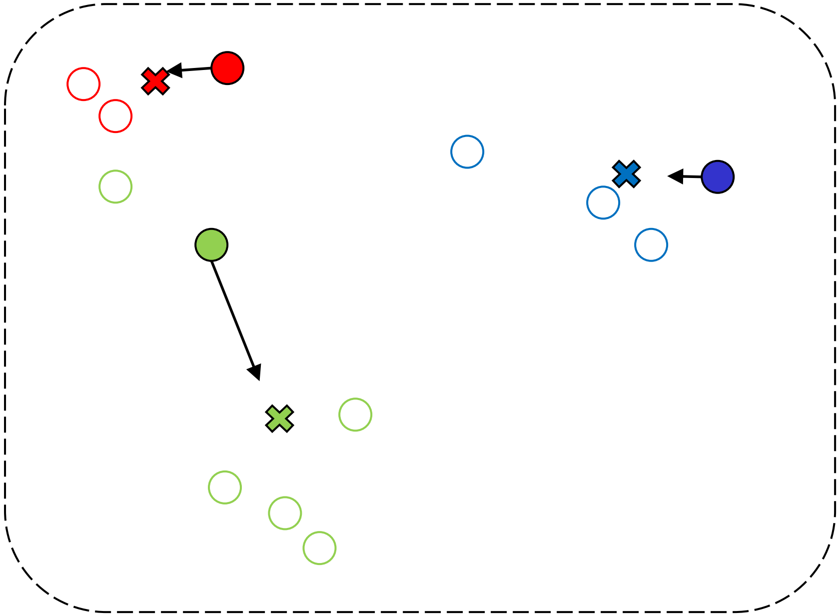
Machine Learning
Unsupervised Learning

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



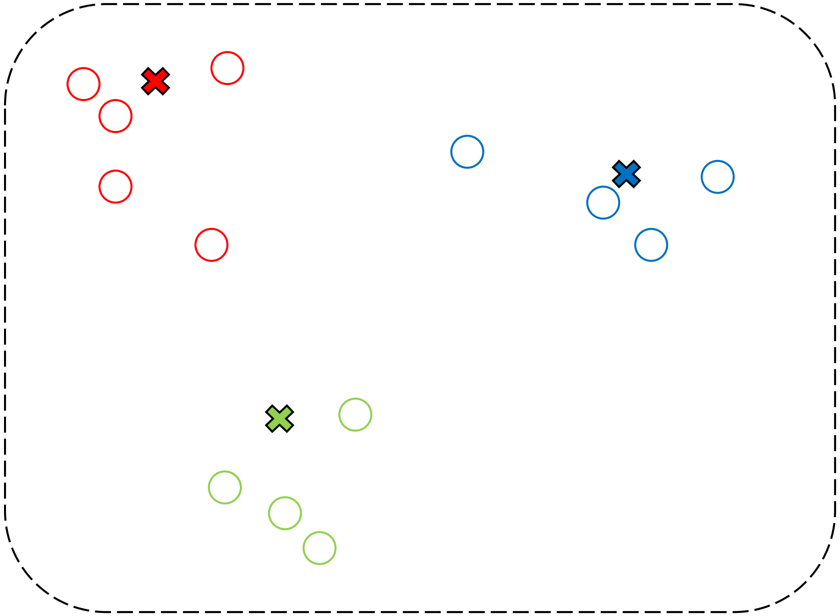
Machine Learning
Unsupervised Learning

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



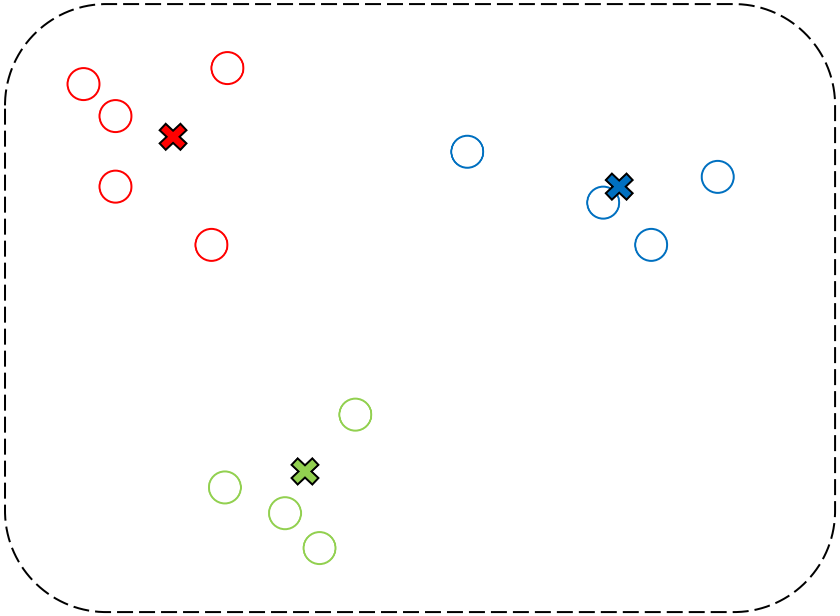
Machine Learning
Unsupervised Learning

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

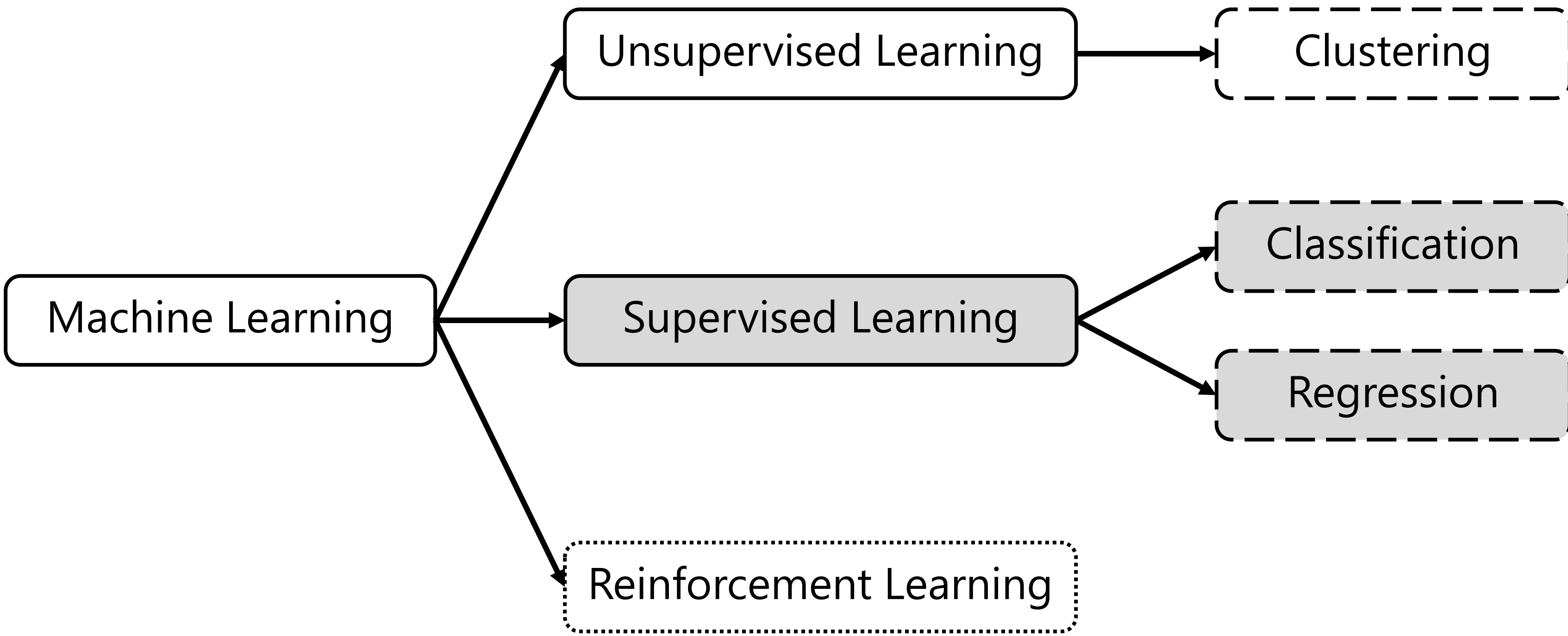


Machine Learning
Unsupervised Learning

- *K*-means clustering
  - Update: re-assign data points to closest mean (repeat until convergence)



Machine Learning
Learning Paradigms



Machine Learning
Supervised Learning

- Goal
    - Find hidden **structure** and **patterns** in data
    - **No annotations** available

Learning Paradigms
Supervised Learning

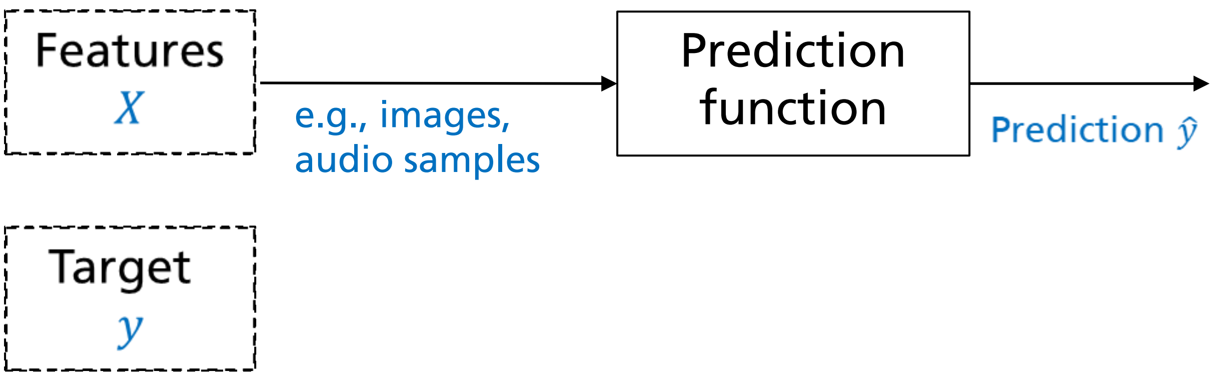
Features
X

e.g., images,
audio samples

Target
y

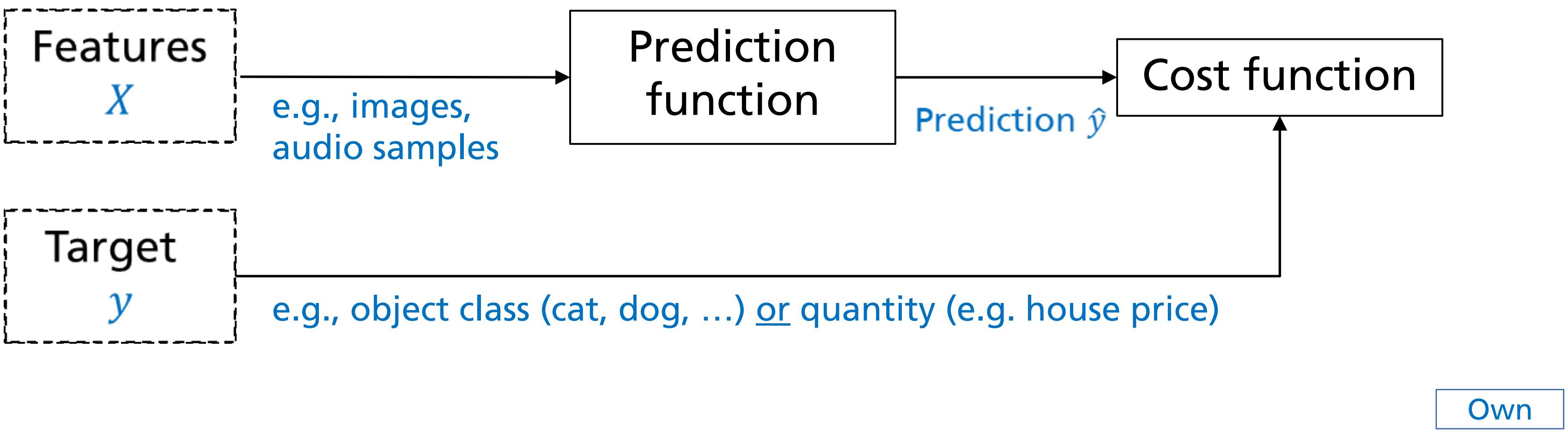
Own

Learning Paradigms
Supervised Learning

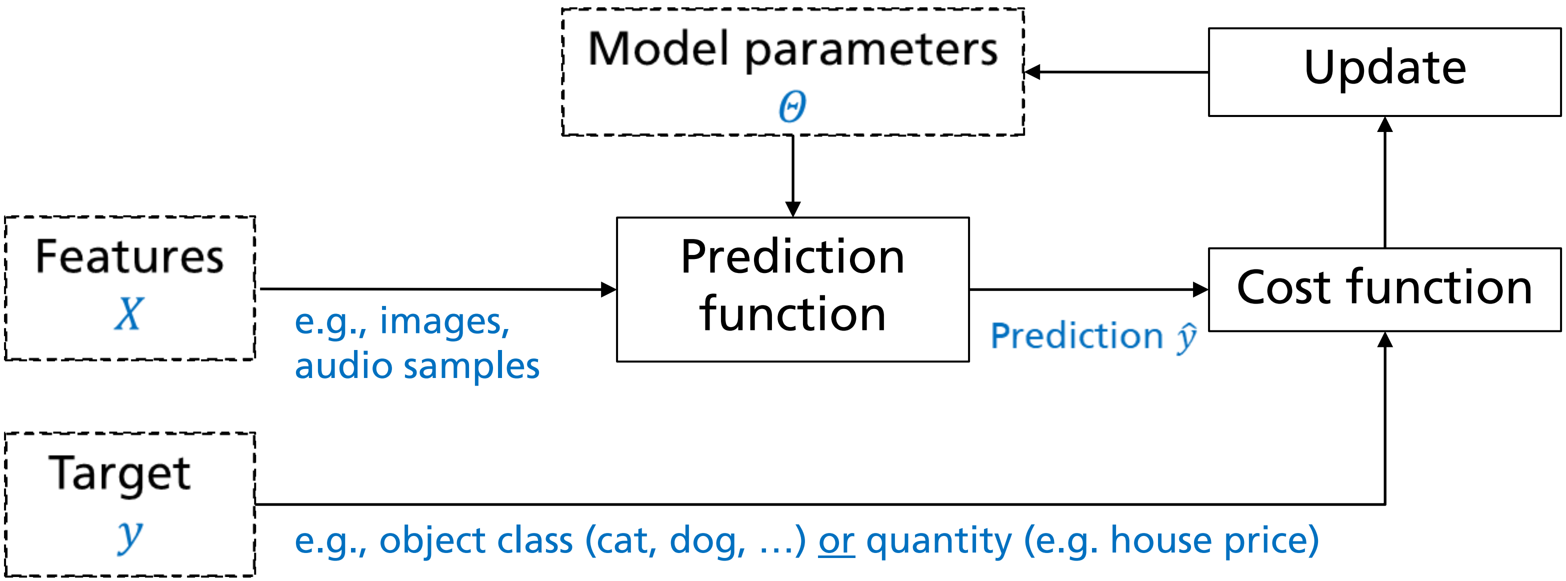


Own

Learning Paradigms
Supervised Learning



Learning Paradigms
Supervised Learning



Own

Learning Paradigms
Supervised Learning - Classification

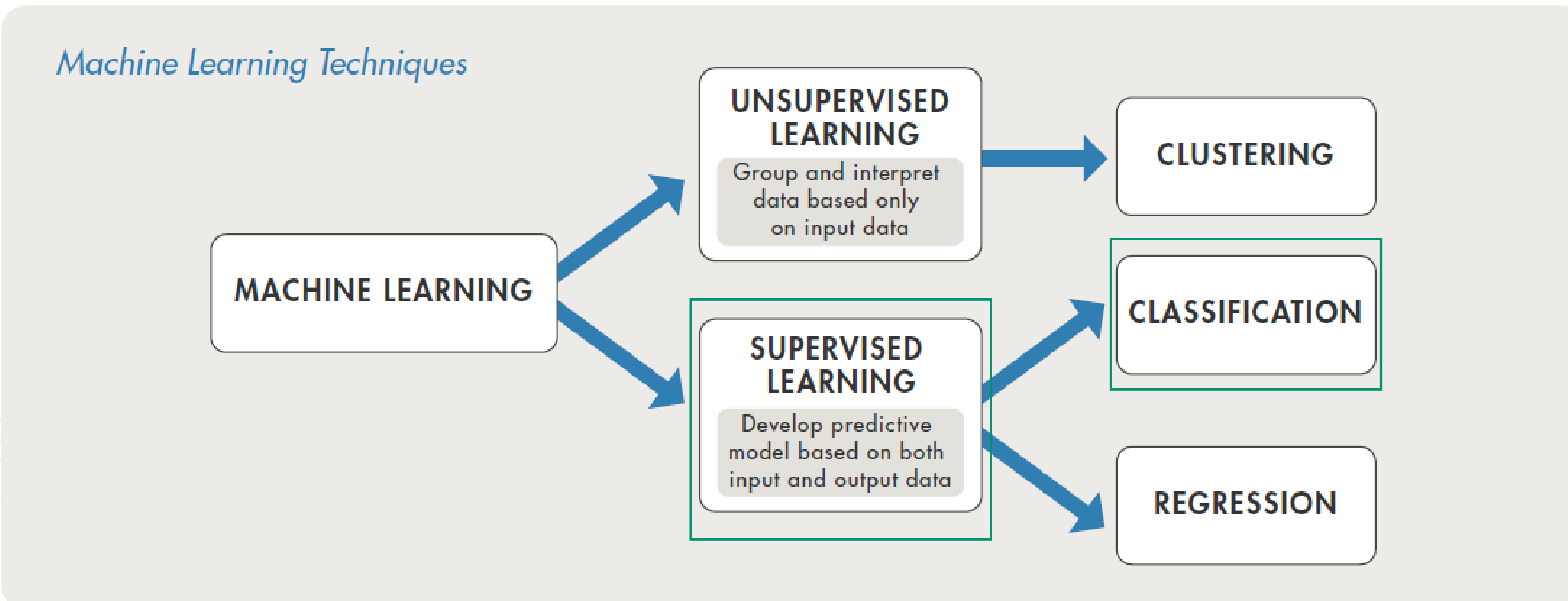


Fig. 1

[2]

Learning Paradigms
Supervised Learning - Classification

- Predict one or multiple categorical labels from features
  - Examples → music genre, instrument(s), key

Learning Paradigms
Supervised Learning - Classification

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

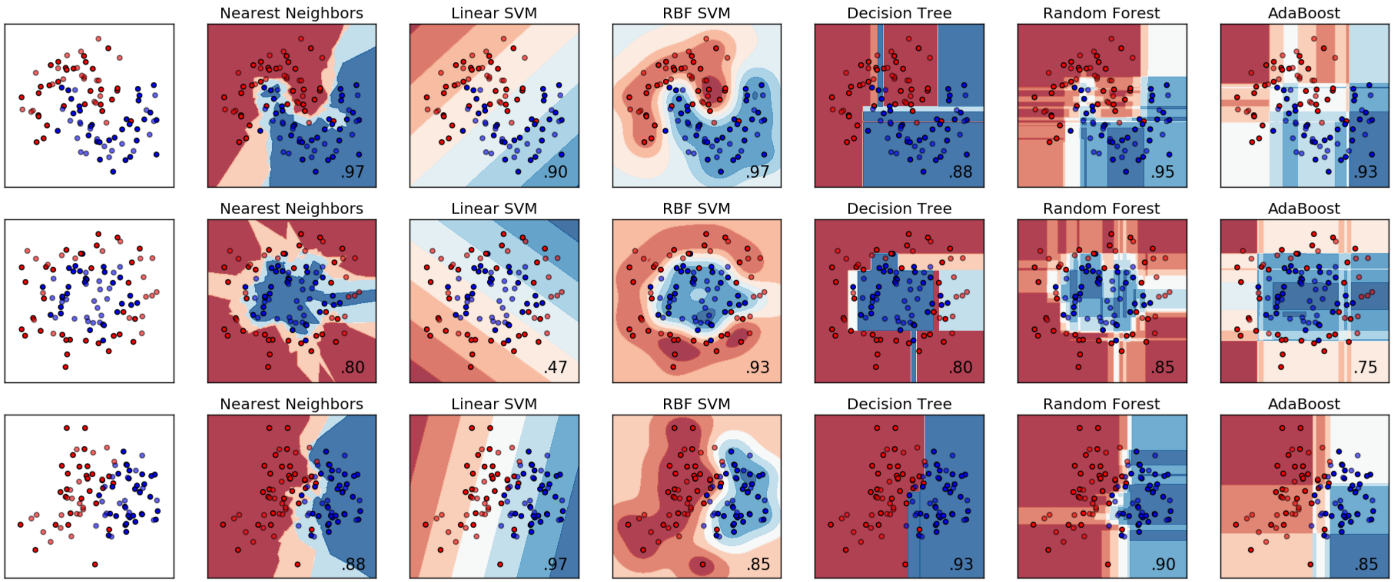


Fig. 3

Learning Paradigms
Supervised Learning - Classification

- Example: *k*-Nearest Neighbors
    - Training → Store all examples

Learning Paradigms
Supervised Learning - Classification

- Example: *k*-Nearest Neighbors
  - Training → Store all examples

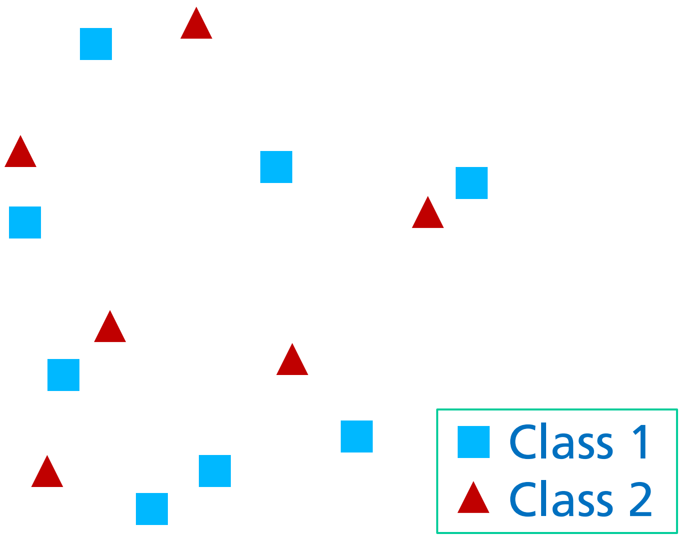


Fig. 4

Learning Paradigms
Supervised Learning - Classification

- Example: *k*-Nearest Neighbors
  - Training → Store all examples
  - Test → Assign test item to dominant class label of the *k* clostest training data items

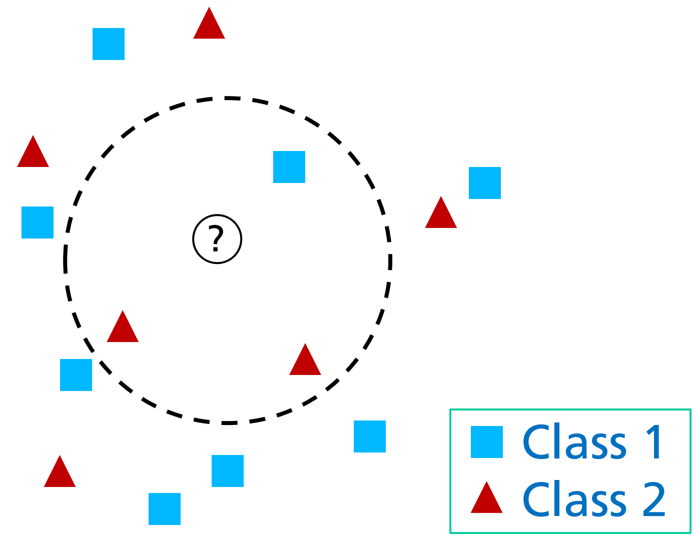


Fig. 4

$k = 3 \rightarrow$? = ▲

Learning Paradigms
Supervised Learning - Classification

- Example: *k*-Nearest Neighbors
  - Training → Store all examples
  - Test → Assign test item to dominant class label of the *k* clostest training data items

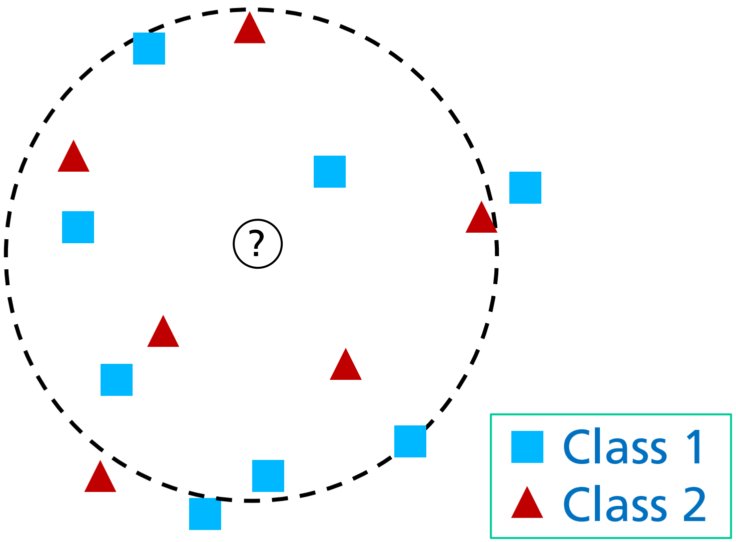


Fig. 4

$k = 11 \rightarrow$? =

Learning Paradigms
Supervised Learning - Classification

- Example: *k*-Nearest Neighbors
  - Training → Store all examples
  - Test → Assign test item to dominant class label of the *k* clostest training data items
- Distance measures
  - Euclidean distance, Manhatten distance, cosine distance, …

Learning Paradigms
Supervised Learning

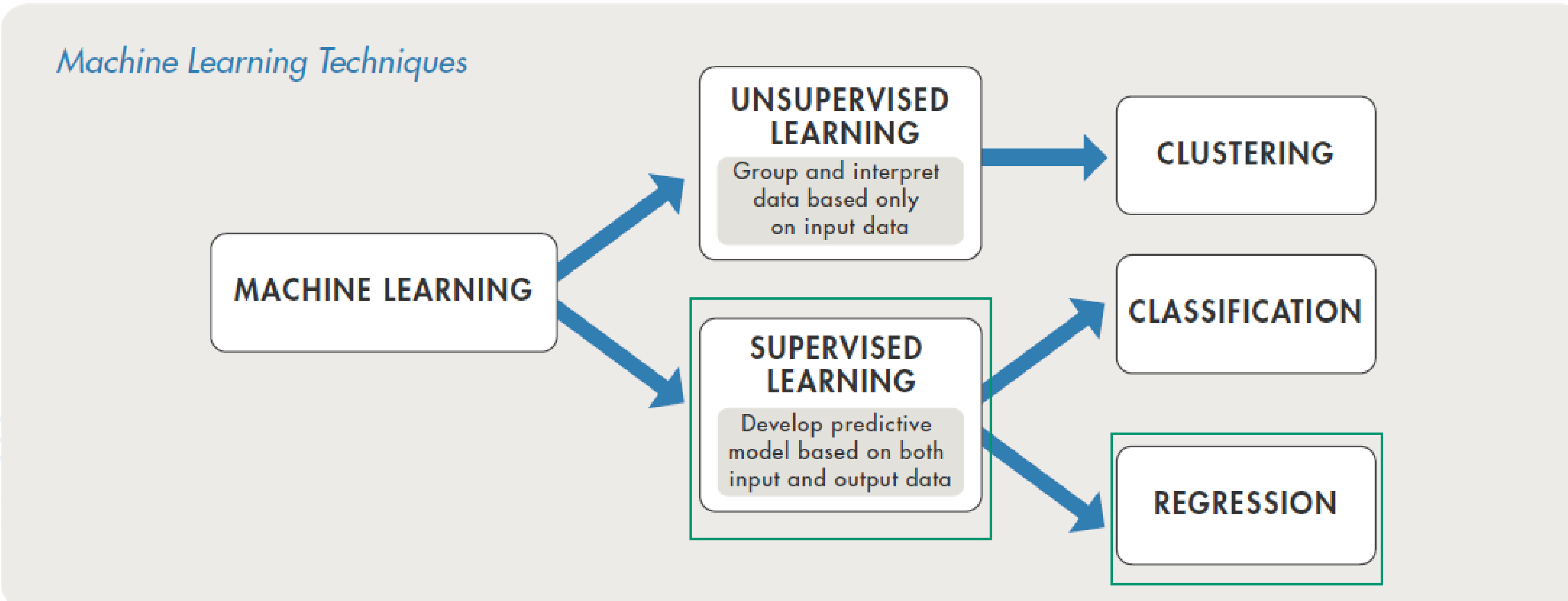


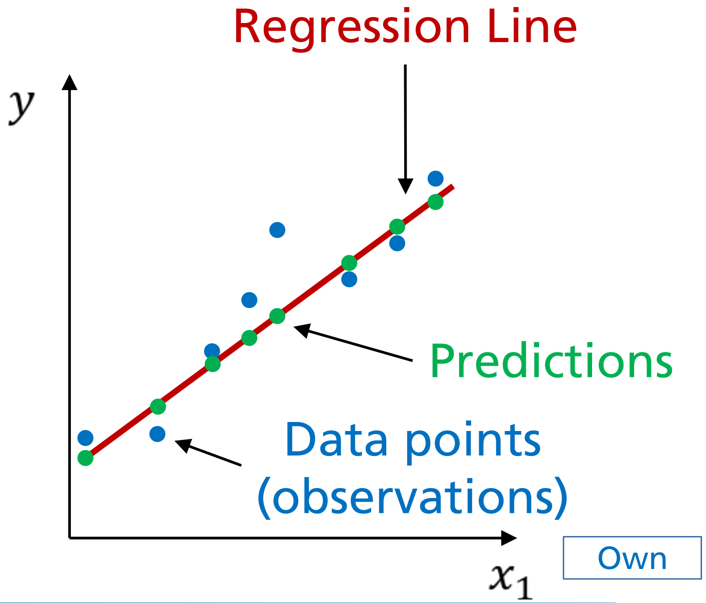
Fig. 1

Learning Paradigms
Supervised Learning - Regression

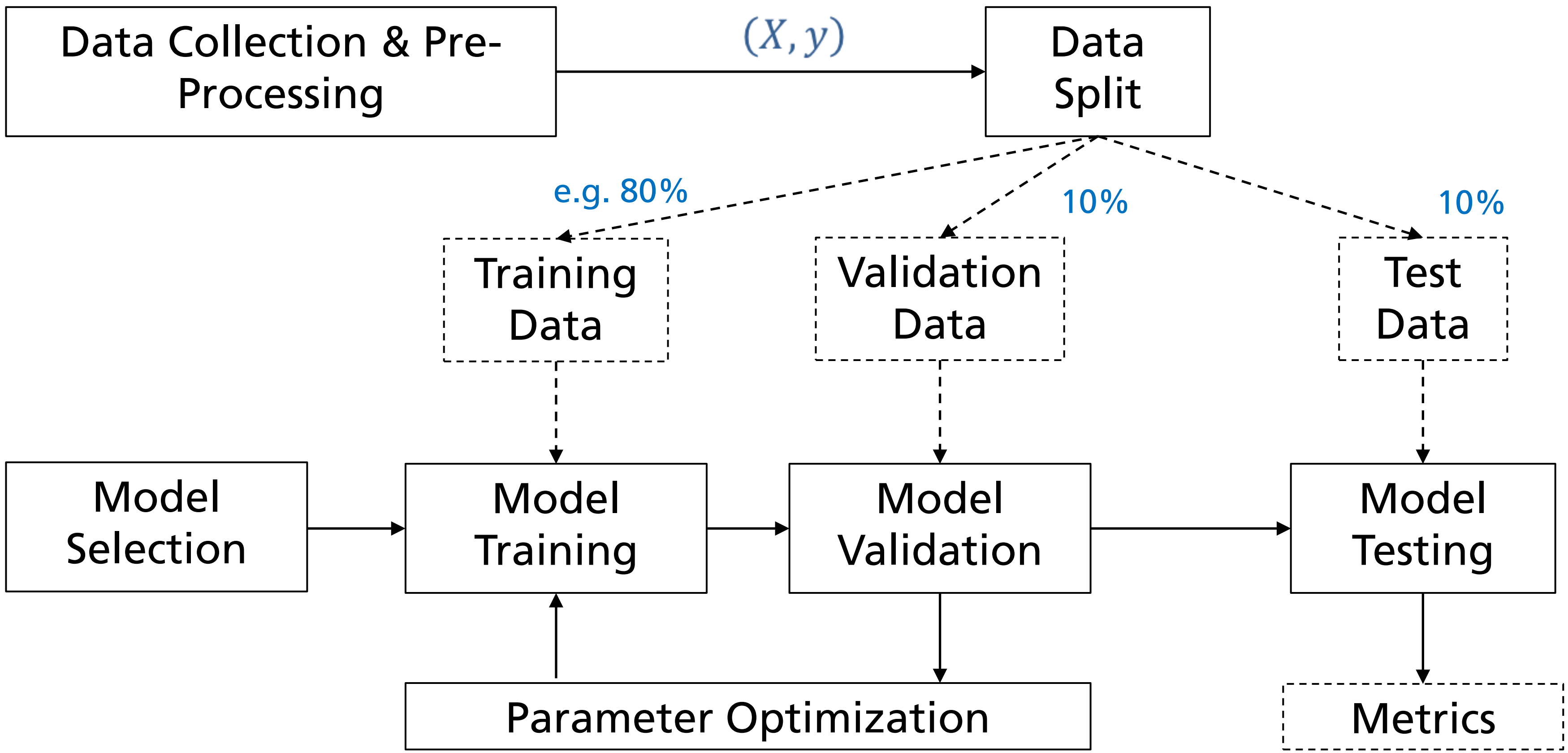
- Goal
  - Predict a dependent (response) variable given one or multiple independent variables (features)
  - Continuous quantities
- Examples
  - Univariate (linear) regression:
    - $y \approx \beta_0 + \beta_1 \, x_1$
      - $\beta_0 \rightarrow$ bias
      - $\beta_1 \rightarrow$ weight

Learning Paradigms
Supervised Learning - Regression

- Goal
  - Predict a dependent (response) variable given one or multiple independent variables (features)
  - Continuous quantities
- Examples
  - Univariate (linear) regression:
    - $y \approx \beta_0 + \beta_1 \, x_1$
      - $\beta_0 \rightarrow$ bias
      - $\beta_1 \rightarrow$ weight



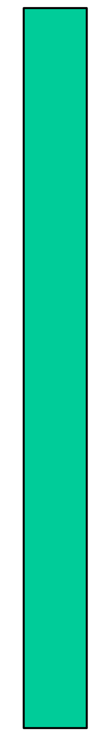
ML Project Pipeline
Overview



Own

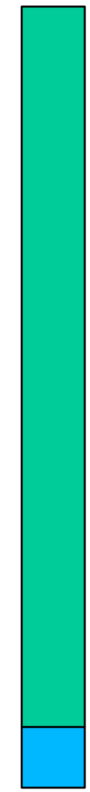
ML Project Pipeline
Data Split

- Training Set
  - Model learns from this data



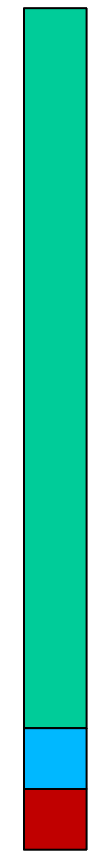
ML Project Pipeline
Data Split

- 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



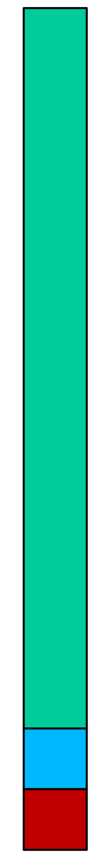
ML Project Pipeline
Data Split

- 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



ML Project 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)



ML Project 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, …

ML Project 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)

ML Project Pipeline
Model Selection

- Many models and approaches exist
  - Types (SVM, GMM, logistic regression, DNNs)
  - Hyperparameters (SVM kernel functions, DNN layer types)

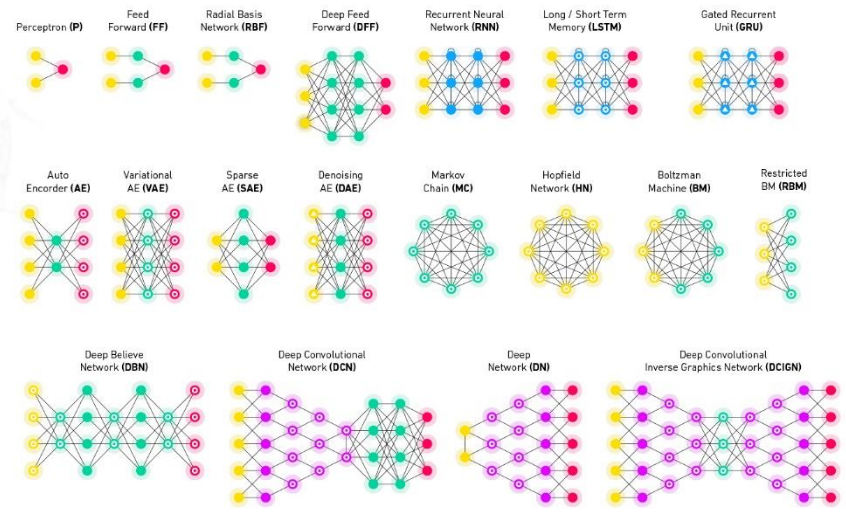


Fig. 6

ML Project 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)

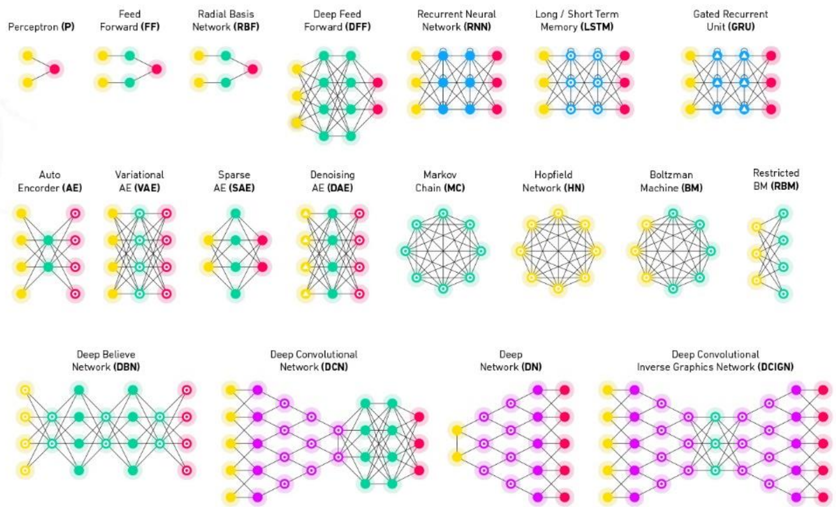


Fig. 6

ML Project 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

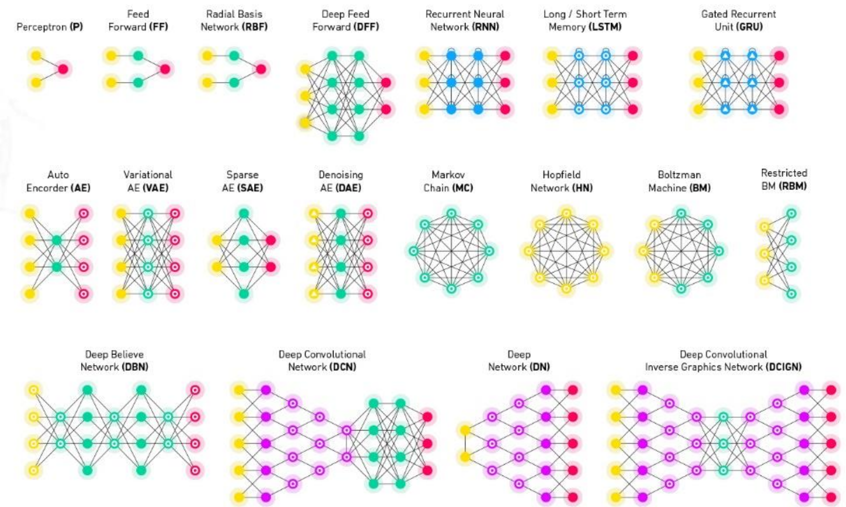


Fig. 6

ML Project Pipeline
Model Training

- Iterative process
  - Typically: start with random parameter initialization

ML Project Pipeline
Model Training

- Iterative process
    - Typically: start with random parameter initialization
    - Use (batches of) training data to iteratively improve model predictions (optimization)
        - Learn from examples

ML Project Pipeline
Model Training

- Iterative process
    - Typically: start with random parameter initialization
    - Use (batches of) training data to iteratively improve model predictions (optimization)
        - Learn from examples
    - Update model parameters according to loss function

ML Project Pipeline
Model Validation

- Regular model evaluation each or multiple training iteration

ML Project Pipeline
Model Validation

- Regular model evaluation each or multiple training iteration

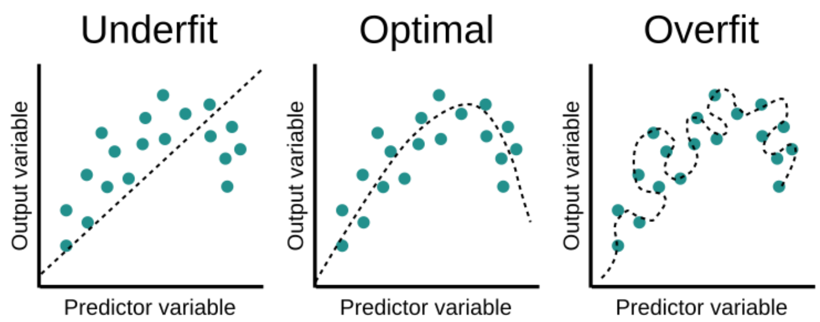


Fig. 7

ML Project Pipeline

Model Validation

- Regular model evaluation each or multiple training iteration
- Helps to
  - optimize model (hyper)parameters
  - detect overfitting on training data
  - stop the training

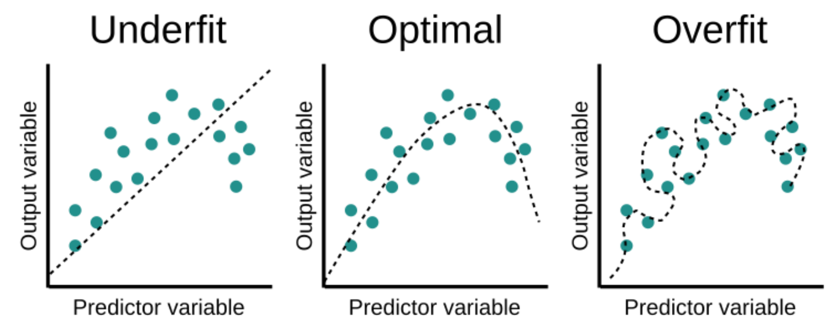
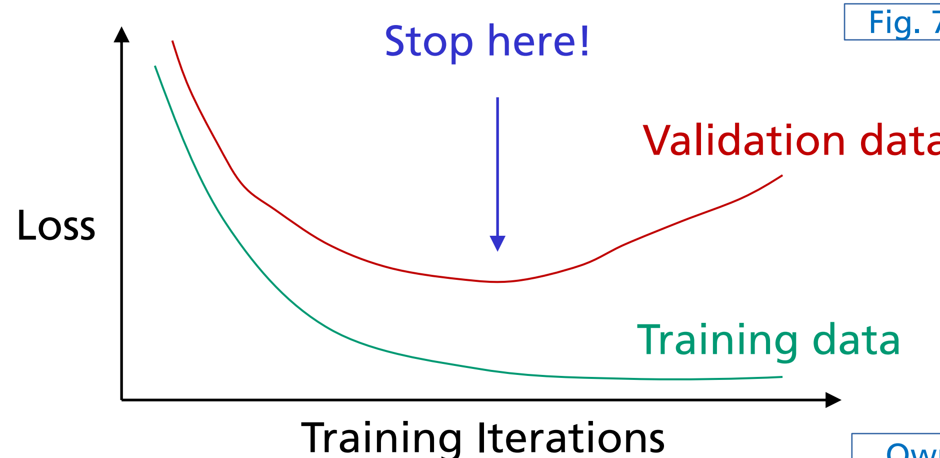


Fig. 7



Own

ML Project Pipeline
Model Testing

- Example: Binary classification evaluation
  - True/false positives (TP/FP)
  - True/false negatives (TN/FN)

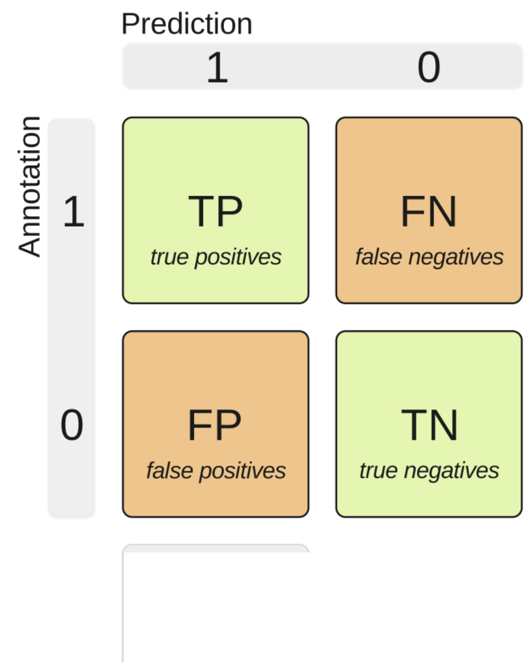


Fig. 8

ML Project Pipeline
Model Testing

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

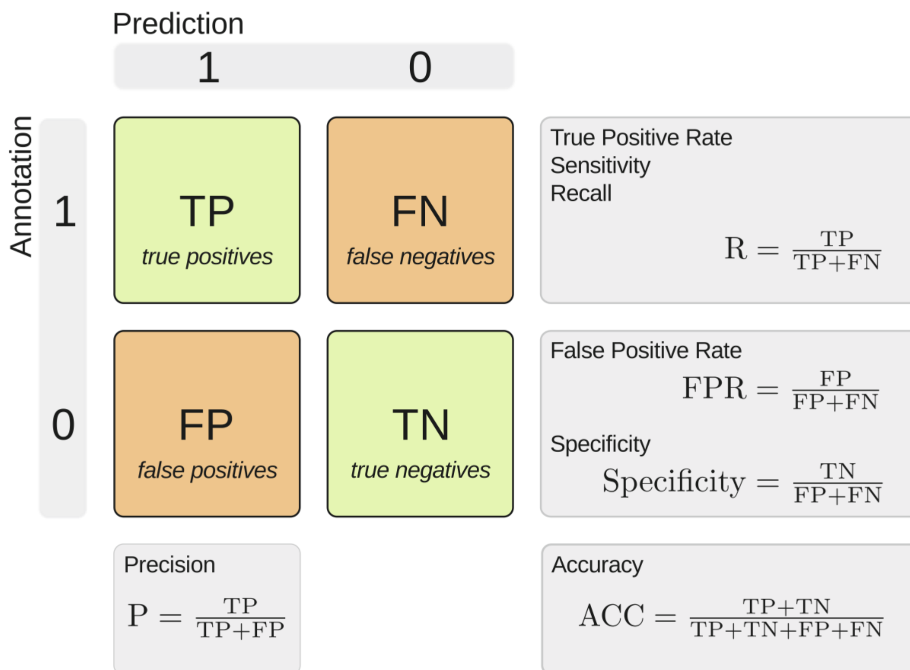


Fig. 8

Audio Processing
Programming Session

Fig. 2.1

References

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

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

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

Srihari, S. N. (2020). *Forward Propagation and Backward Propagation (Deep Learning Lecture)*. Retrieved from https://cedar.buffalo.edu/~srihari/CSE676/6.5.0 Forward Backward.pdf

Virtanen, T., Plumbley, M. D., & Ellis, D. (Eds.). (2018). *Computational Analysis of Sound Scenes and Events*. Cham, Switzerland: Springer International Publishing.

Images

Fig. 1: [Machine Learning, 2016], p. 4, Fig. 2

Fig. 2: https://i0.wp.com/www.sthda.com/sthda/RDoc/figure/clustering/ partitioning-cluster-analysis-k-means-plot-4-groups-1.png

Fig. 3: https://i.stack.imgur.com/hsilO.png (https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)

Fig. 4: https://miro.medium.com/max/975/1*OyYyr9qY-w8RkaRh2TKo0w.png (reproduced)

Fig. 5: https://lilianweng.github.io/lil-log/assets/images/self-sup-lecun.png

Fig. 6: https://www.asimovinstitute.org/wp-content/uploads/2019/04/NeuralNetworkZoo20042019.png

Fig. 7: https://www.educative.io/api/edpresso/shot/6668977167138816/image/5033807687188480

Fig. 8: [Virtanen, 2018], p. 170, Fig. 6.7

Fig. 9: https://miro.medium.com/max/915/1*SJPacPhP4KDEB1AdhOFy_Q.png

Fig. 10: https://www.skampakis.com/wp-content/uploads/2018/03/simple_neural_network_vs_deep_learning.jpg

Fig. 11: https://pic4.zhimg.com/80/v2-057b248288a8af2f01272a956f862873_1440w.png

Fig. 12: https://blog.e-kursy.it/deeplearning4j-workshop/video/html/presentation_specific/img/4_activation_functions.png

Images

Fig. 13: https://blog.paperspace.com/content/images/2018/05/challenges-1.png

Fig. 14: https://www.cs.umd.edu/~tomg/img/landscapes/noshort.png

Fig. 15: https://blog.paperspace.com/content/images/2018/05/grad.png

Fig. 16: https://www.wandb.com/articles/intro-to-cnns-with-wandb

Fig. 17: https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Fig. 18: https://wiki.tum.de/download/attachments/22578349/RNN1.png

Fig. 19: https://stanford.edu/~shervine/teaching/cs-230/illustrations/architecture-rnn-ltr.png

Fig. 20: [Srihari, 2020], p.8, (Fig. 1)

Images

Fig. 1:

References

[1] Sternberg, R. J. (2022). human intelligence. Encyclopedia Britannica. https://www.britannica.com/science/human-intelligence-psychology

[2] Gross, R., Psychology (2015). The Science of Mind and Behaviour, Hodder Education

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

[4] Russell, S., Norvig, P. (2016). Artificial Intelligence: A Modern Appoach, **PEV**, third ed.

[5] Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96. Springer. pp. 151–170.

Audio

[Audio 1] https://freesound.org/people/xserra/sounds/196765/

[Audio 2] https://freesound.org/people/IliasFlou/sounds/498058/ (~0:00 – 0:05)

[Audio 3] https://freesound.org/people/danlucaz/sounds/517860/ (~0:00 – 0:05)

[Audio 4] https://freesound.org/people/IENBA/sounds/489398/ (~0:00 – 0:07)