UiT The Arctic University of Norway

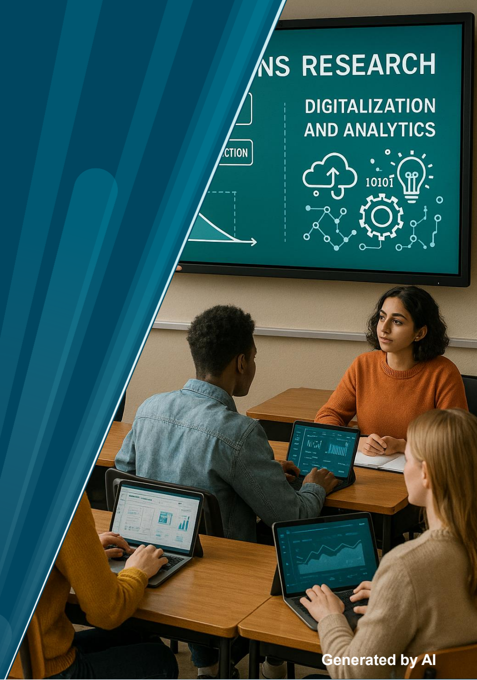
Introduction to Machine Learning: Methods and Applications

INE-3800 Operation Research I

Wahyu Andy Prastyabudi

Department of Industrial Engineering
Faculty of Engineering Science and Technology

2025-09-19



Outlines

Introduction

What is machine learning (ML)?

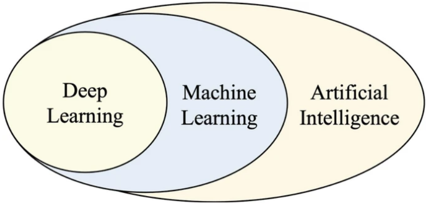


Figure 1: Subfields of AI. Taken from Wang (2025)

- ML is a subfield of AI focusing on building a machine (i.e., model) to learn from experience with minimal human intervention
- ML finds generalizable predictive patterns from the data
- It involves mathematical models for approximating the data

What is machine learning (ML)? (cont.)

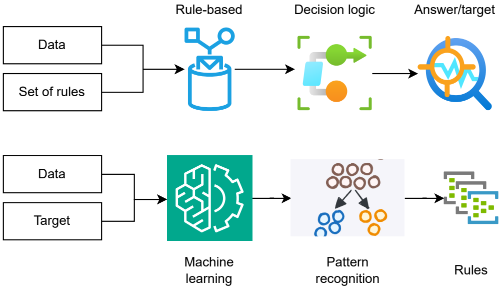


Figure 2: Rule-based learning vs machine learning

- Rule-based systems require pre-existing knowledge to build robust set of rules
- Rule-based is often seen in statistical modelling, aiming at inferencing and estimating unknown parameters
- ML allows to detect the pattern automatically using the data without explicit rule-based instruction
- It uses the rules to predict future unseen data
- ML has capabilities to extract knowledge from data and improve the performance (i.e., adaptive learning capability)

Key Concepts of ML

Key Concepts of ML

- **Data-driven**
  - The goal is to extract valuable patterns from data
  - Data quality plays a critical role in producing reasonable outputs
- **Model**
  - It maps the input features to target outputs
  - Good models generalize well to unseen data
- **Learning**
  - Automatic pattern recognition
  - Parameters optimization of the model

Types of ML

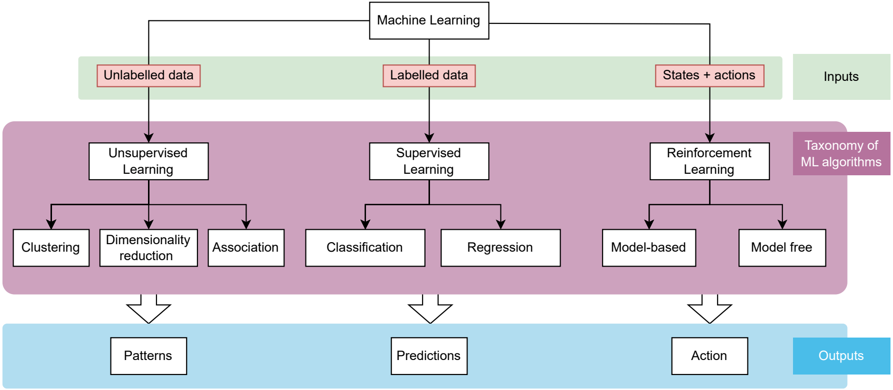


Figure 3: Taxonomy of ML algorithms

General Workflow

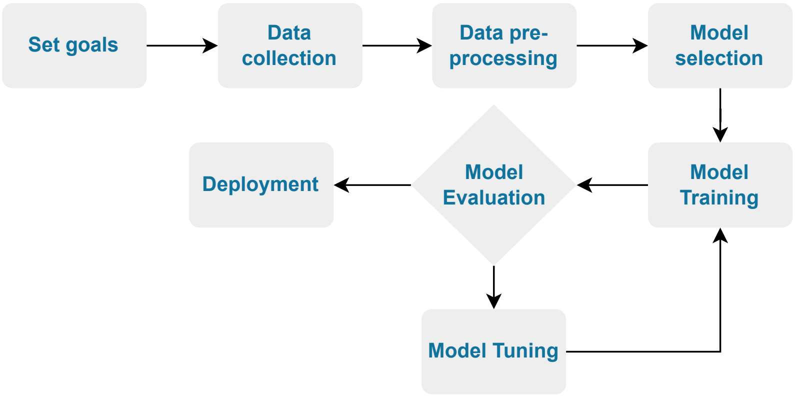


Figure 4: Typical ML workflow

Tools and Ecosystems

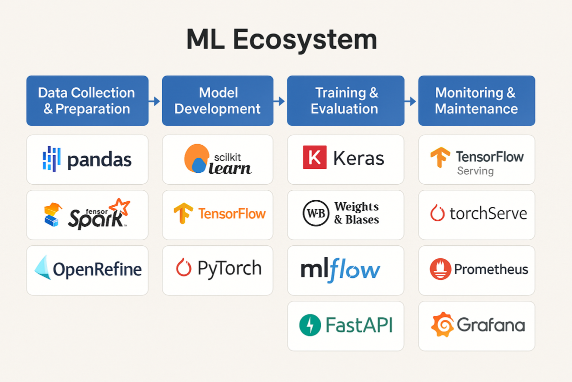


Figure 5: Popular ML tools. Generated by Copilot.

- GUI-based tools:
  - RapidMiner
  - KNIME
  - AzureML
- Interactive Development Environments (IDEs):
  - Jupyter Notebook hosted by UiT
  - Google Colab

Data Retrieval and Pre-processing

Data retrieval

Open data site	Description
Data.gov	The home of U.S. Government's open data
open-data.europa.eu	The home of European Commission's open data
Data.worldbank.org	Open data initiative from the World Bank
data.norge.no	Open data provided by Norwegian public sector
Opendatasoft	Market place of various open dataset
UCI ML repository	Collection of databases used by ML community
OpenML	Open platform for sharing datasets, algorithms, and experiments

Data pre-processing

Do data quality checks now! You will save your time and prevent further problems

Data pre-processing

Do data quality checks now! You will save your time and prevent further problems

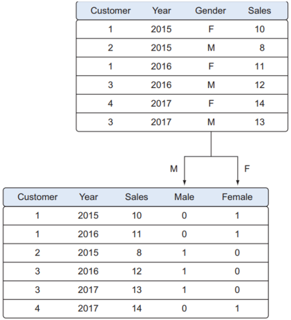
- Cleansing
  - Missing values
  - Outliers
  - Spaces, typos, ...
- Integration
- Transformation
  - Aggregating
  - Extrapolating
  - Normalization

Pre-processing examples

Table 2.4 An overview of techniques to handle missing data

Technique	Advantage	Disadvantage
Omit the values	Easy to perform	You lose the information from an observation
Set value to `null`	Easy to perform	Not every modeling technique and/or implementation can handle `null` values
Impute a static value such as 0 or the mean	Easy to perform You don't lose information from the other variables in the observation	Can lead to false estimations from a model
Impute a value from an estimated or theoretical distribution	Does not disturb the model as much	Harder to execute You make data assumptions
Modeling the value (nondependent)	Does not disturb the model too much	Can lead to too much confidence in the model Can artificially raise dependence among the variables Harder to execute You make data assumptions

Figure 6: Handling missing data. Taken from Cielen and Meysman (2016).



Customer	Year	Gender	Sales
1	2015	F	10
2	2015	M	8
1	2016	F	11
3	2016	M	12
4	2017	F	14
3	2017	M	13

Customer	Year	Sales	Male	Female
1	2015	10	0	1
1	2016	11	0	1
2	2015	8	1	0
3	2016	12	1	0
3	2017	13	1	0
4	2017	14	0	1

Figure 7: Transform data into dummies. Taken from Cielen and Meysman (2016).

Outliers removal

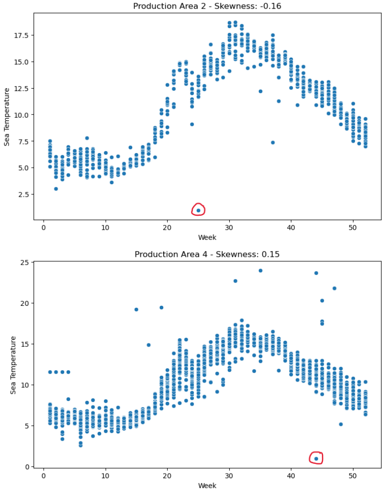


Figure 8: Outliers in sea temperature.

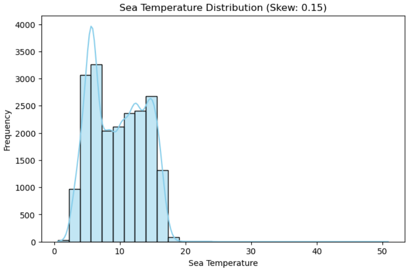


Figure 9: Data skewness of sea temperature.

- Z-Score
  - measures how many standard deviations a data point is from the mean
- IQR (Inter Quartile Range)
  - the range between Q1 and Q3 percentile
- Replace with mean/median

Outliers removal (cont.)

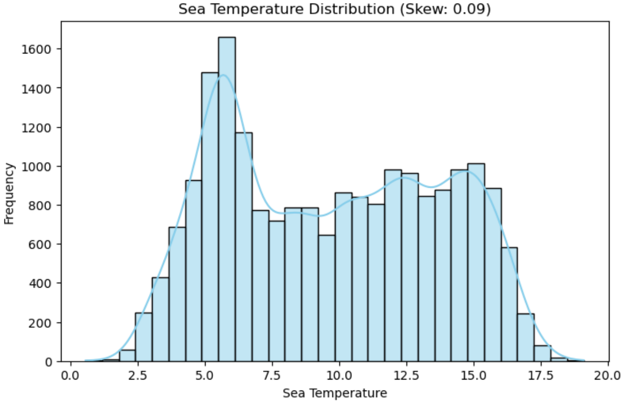


Figure 10: Data skewness reduce after outliers removal.

Transformation

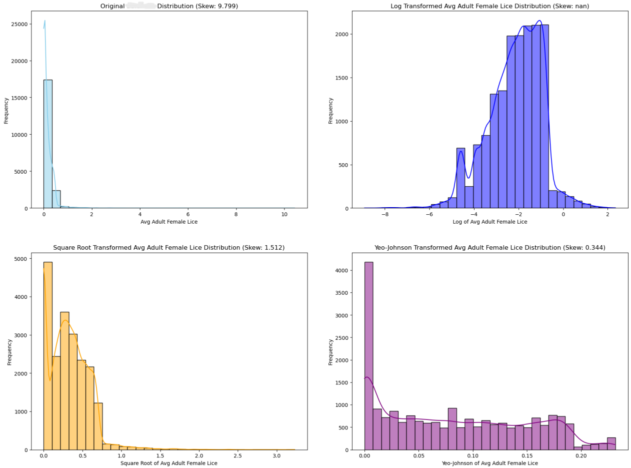


Figure 11: Transforming highly skew data

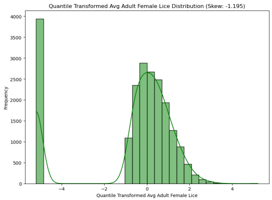


Figure 12: Quantile data transformation.

Transformation techniques:

- Min-max scaling
- Z-score scaling
- Log transformation
- Box-cox transformation
- Yeo-Johnson transformation
- Quantile transformation
- Square root transformation

Algorithms and Techniques

Popular Algorithms

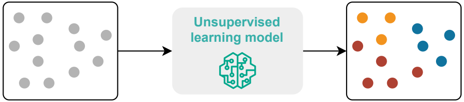


Figure 13: Unsupervised learning illustration

Table 2: Unsupervised learning algorithms.

Category	Algorithm	Use case example
Clustering	K-means clustering	Customer segmentation, supplier grouping by performance
	Dbscan clustering	Detecting anomalies in delivery patterns
	Hierarchical clustering	Analysing production batch similarities
Dimensionality reduction	PCA	Visualizing high-dimensional sensor data
Association mining	Apriori	Discover frequent items in order transaction
	FP-growth	Identifying co-occurring parts in inventory

Popular Algorithms (cont.)

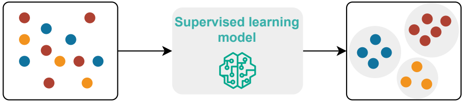


Figure 14: Supervised learning illustration

Table 3: Supervised learning algorithms.

Category	Algorithm	Use case example
Regression	Linear regression	Predicting maintenance cost
	Logistic regression	Predicting machine failure
Classification	KNN	Classifying defective and non-defective products
	SVM	Safety incidents classification
	Naïve Bayes	Transaction records filtering
	Neural Network (NN)	Product quality monitoring using image classification

Popular Algorithms (cont.)

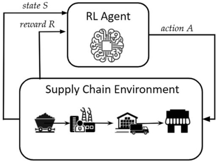


Figure 15: Reinforcement learning in SCM. Taken from Rolf et al. (2023)

Table 4: Refinforcement learning algorithms.

Category	Algorithm	Use case example
Model-free	Q-learning	Optimizing robot navigation, optimizing inventory
	Monte Carlo methods	Evaluating long-term supply chain policies
Model-based	Markov decision process (MDP)	Dynamic scheduling in manufacturing lines

Classification Evaluation Metrics

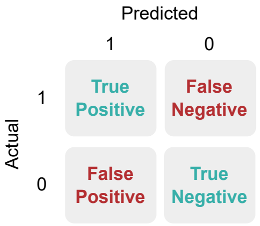


Figure 16: Confusion matrix

Classification Evaluation Metrics

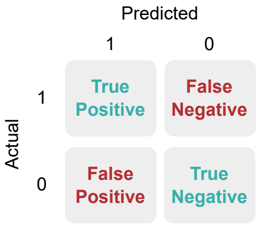


Figure 16: Confusion matrix

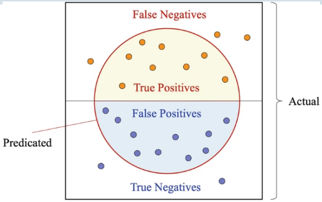


Figure 17: Illustration of classifier evaluation terms. Taken from Wang (2025).

Table 5: Classification metrics.

Metric	Formula	Intrepretation
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Overall correctness
Precision	$\frac{TP}{TP+FP}$	How reliable the positive prediction
Recall	$\frac{TP}{TP+FN}$	Ability to detect actual positive sample
Specificity	$\frac{TN}{TN+FP}$	Ability to detect actual negative sample
F1 score	$\frac{2TP}{2TP+FP+FN}$	Balance between precision and recall

What metric to choose?

- Measure choice depends on the nature of problem
- Consider the importance of identifying TP/TN and the cost of FP/FN
- **Recall** is used when missing a positive case is risky. Ex: predicting defective products using sensitive sensor
- **Specificity** ensures normal operations are not wrongly terminated. Ex: safety maintenance
- See the trade-off for classification metrics at Precision & Recall

Regression Evaluation Metrics

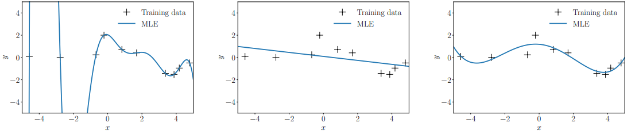
Table 6: Regression metrics.

Metric	Formula	Notes
Mean Absolute Error	MAE $= \frac{1}{n}\sum_{i=0}^{n-1} \lVert y_i - \hat{y}_i \rVert$	$\hat{y}_i$ is the predicted value of the i-th sample, and y_i is the corresponding true value. n is the number of samples
Mean Squared Error	MSE $= \frac{1}{n}\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2$	
Mean Absolute Percentage Error	MAPE $= \frac{1}{n}\sum_{i=0}^{n-1} \frac{\lVert y_i - \hat{y}_i \rVert}{\max(\epsilon, \lVert y_i \rVert)}$	ϵ is an arbitrary small yet strictly positive number to avoid undefined results when y is zero
Root Mean Squared Error	$RMSE = \sqrt{MSE}$	
R^2 Score	$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}$	Represents the proportion of variance
Adjusted R^2	Adj$R^2 = 1 - (1 - R^2)\frac{n-1}{n-k-1}$	Comparing models with different numbers of features, often used in multiple regression. k is number of features or predictors

Clustering Evaluation Metrics

Metric	Notes
Silhouette Coefficient	Measure the similarity of a sample point in its own cluster vs others
Davies-Bouldin Index	Measure average similarity between clusters
Rand index	Similarity between predicted and true clusters, adjusted for chance

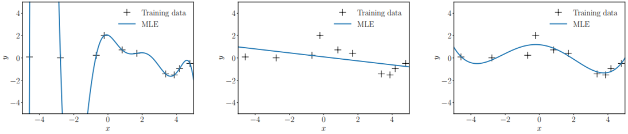
Model Fitting



(a) Overfitting (b) Underfitting. (c) Fitting well.

Figure 18: Fitting problems in ML. Taken from Deisenroth, Faisal, and Ong (2020).

Model Fitting



(a) Overfitting (b) Underfitting. (c) Fitting well.

Figure 18: Fitting problems in ML. Taken from Deisenroth, Faisal, and Ong (2020).

Underfitting

- Build more complex model
- Add relevant features
- Feature engineering
- Check data quality
- Adjust training parameters (e.g., reduce regularization)

Overfitting

- Perform appropriate cross-validation
- Apply regularization
- Remove irrelevant features
- Simplify the model
- Consider early stopping

Incorporating ML into OR

Combining Optimization with ML for Better Decisions

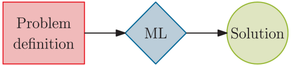


Figure 19: End to end learning

- The ML model is trained to produce solutions based on input instances

Combining Optimization with ML for Better Decisions

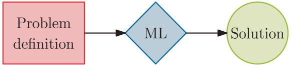


Figure 19: End to end learning

- The ML model is trained to produce solutions based on input instances

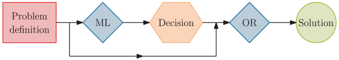


Figure 20: Learning to configure algorithms

- ML can provide a parametrization for the combinatorial optimization algorithm
- ML solutions may reduce the size of input data and parameters to scale up the optimization

Combining Optimization with ML for Better Decisions

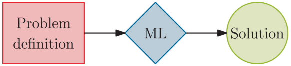


Figure 19: End to end learning

- The ML model is trained to produce solutions based on input instances

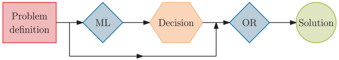


Figure 20: Learning to configure algorithms

- ML can provide a parametrization for the combinatorial optimization algorithm
- ML solutions may reduce the size of input data and parameters to scale up the optimization

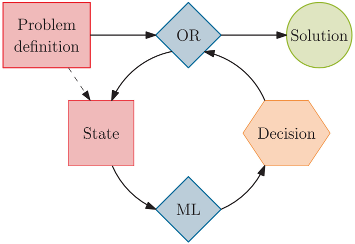


Figure 21: Combinatorial optimization algorithms repeatedly queries ML to make decisions

- ML models can be trained to improve branching decisions in the branch-and-bound framework for MILP
- Bengio, Y., Lodi, A. & Prouvost, A. (2021). Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, *290*(2), 405–421. https://doi.org/10.1016/j.ejor.2020.07.063

What's next?

`.: Continue to Part 2 ->`

References

References

Cielen, Davy, and Arno Meysman. 2016. *Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools*. Simon and Schuster.

Deisenroth, Marc Peter, A Aldo Faisal, and Cheng Soon Ong. 2020. *Mathematics for Machine Learning*. Cambridge University Press.

Rolf, Benjamin, Ilya Jackson, Marcel Müller, Sebastian Lang, Tobias Reggelin, and Dmitry Ivanov. 2023. "A Review on Reinforcement Learning Algorithms and Applications in Supply Chain Management." *International Journal of Production Research* 61 (20): 7151–79. https://doi.org/10.1080/00207543.2022.2140221.

Wang, Wenmin. 2025. *Principles of Machine Learning: The Three Perspectives*. Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-97-5333-8.