Applied Machine Learning
Class 1

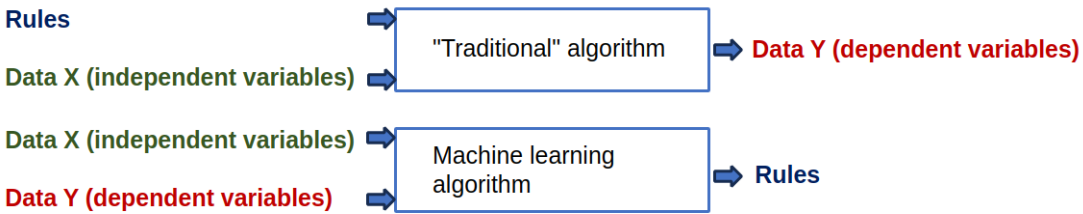
Master 2 Physique Fondamentale et Applications

Daniel Förster
daniel.forster@univ-orleans.fr
UFR Sciences et Techniques Université d'Orléans

Chapter 1: Introduction

- What is machine learning
- Job market
- Common tools
- Literature
- Course overview

What is Machine Learning?



Leads to systems that can accomplish tasks you are able to do, but you would not be able to tell someone how to do them.
Usually computational intensive, but more robust and general compared to handcrafted algorithms

Machine Learning Is Not Magic: Things to consider

- Human input is required for feature engineering, model selection, and interpretation
- Algorithms identify patterns and optimize performance metrics
- Simple statistical or rule-based methods may outperform ML for small datasets
  $\rightarrow$ Establish simple baseline result first

A Brief History of Machine Learning

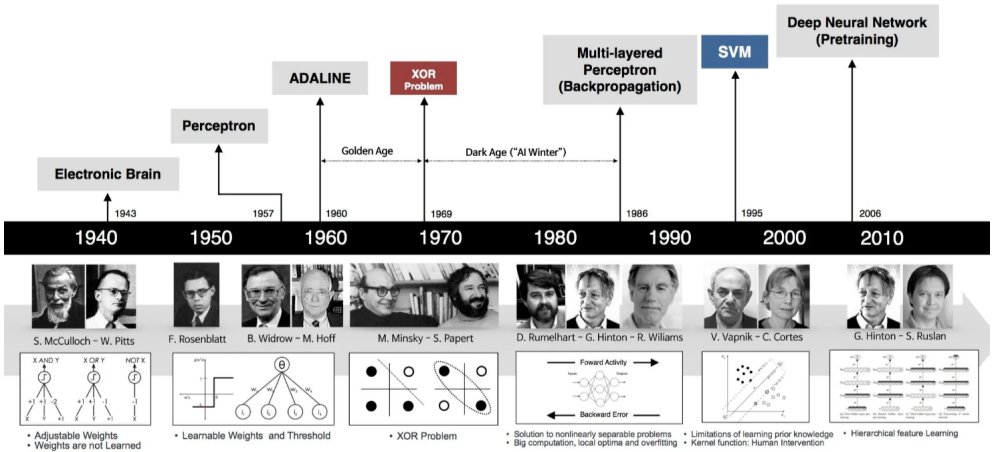


Image source [1]

Machine Learning as a new paradigm in science

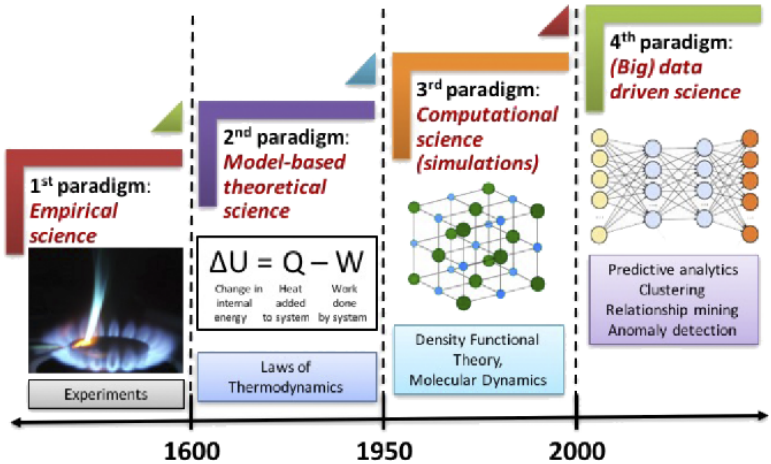


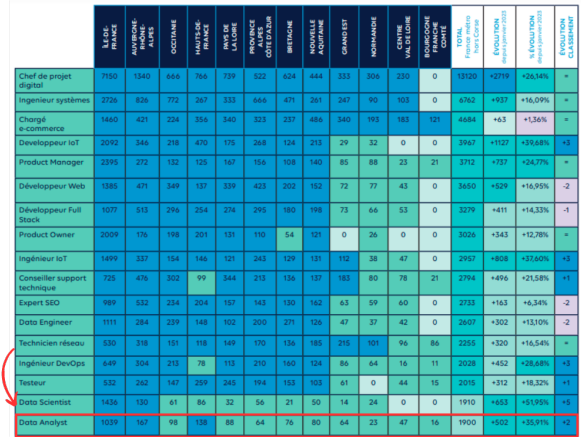
Image source [2]

Applications of Machine Learning

- Chess/Go engine
- Handwritten digit recognition
- Credit card fraud detection
- Face detection
- Biometric identification
- Automatic translation
- Speech recognition
- Amazon article recommendation
- Youtube video recommendations
- Traffic sign recognition
- Lane assist
- Protein folding
- Chat-GPT
- Image generation

So far, niche applications and limited economic impact, but trend to more general, impactful systems.

The Jobs Market for Machine Learning France 2023



	ÎLE-DE-FRANCE	AUVERGNE-RHÔNE-ALPES	OCCITANIE	HAUTS-DE-FRANCE	PAYS DE LA LOIRE	PROVENCE-ALPES-CÔTE D'AZUR	BRETAGNE	NOUVELLE AQUITAINE	GRAND EST	NORMANDIE	CENTRE VAL DE LOIRE	BOURGOGNE FRANCHE-COMTÉ	TOTAL France métro hors Corse	ÉVOLUTION depuis janvier 2022	% ÉVOLUTION depuis janvier 2022	ÉVOLUTION CA-CONTRE
Chef de projet digital	7150	1340	666	766	739	522	624	444	333	306	230	0	13120	+2719	+26,14%	=
Ingenieur systèmes	2726	826	772	267	333	666	471	261	247	90	103	0	6762	+937	+16,09%	=
Chargé e-commerce	1460	421	224	356	340	323	237	486	340	193	183	121	4684	+63	+1,36%	=
Developpeur IoT	2092	346	218	470	175	268	124	213	29	32	0	0	3967	+1127	+39,68%	+3
Product Manager	2395	272	132	125	167	156	108	140	85	88	23	21	3712	+737	+24,77%	=
Développeur Web	1385	471	349	137	339	423	202	152	72	77	43	0	3650	+529	+16,95%	-2
Développeur Full Stack	1077	513	296	254	274	295	180	198	73	66	53	0	3279	+411	+14,33%	-1
Product Owner	2009	176	198	201	131	110	54	121	0	26	0	0	3026	+343	+12,78%	=
Ingénieur IoT	1499	337	154	146	121	243	129	131	112	38	47	0	2957	+808	+37,60%	+3
Conseiller support technique	725	476	302	99	344	213	136	137	183	80	78	21	2794	+496	+21,58%	+1
Expert SEO	989	532	234	204	157	143	130	162	63	59	60	0	2733	+163	+6,34%	-2
Data Engineer	1111	284	239	148	102	200	271	126	47	37	42	0	2607	+302	+13,10%	-2
Technicien réseau	530	318	151	118	149	170	136	185	215	101	96	86	2255	+320	+16,54%	=
Ingénieur DevOps	649	304	213	78	113	210	160	124	86	64	16	11	2028	+452	+28,68%	+3
Testeur	532	262	147	259	245	194	153	103	61	0	44	15	2015	+312	+18,32%	+1
Data Scientist	1436	130	61	86	32	56	21	50	14	24	0	0	1910	+653	+51,95%	+5
Data Analyst	1039	167	98	138	88	64	76	80	64	23	47	16	1900	+502	+35,91%	+2

Image source [3]

Common Tools for Machine Learning
Kaggle survey

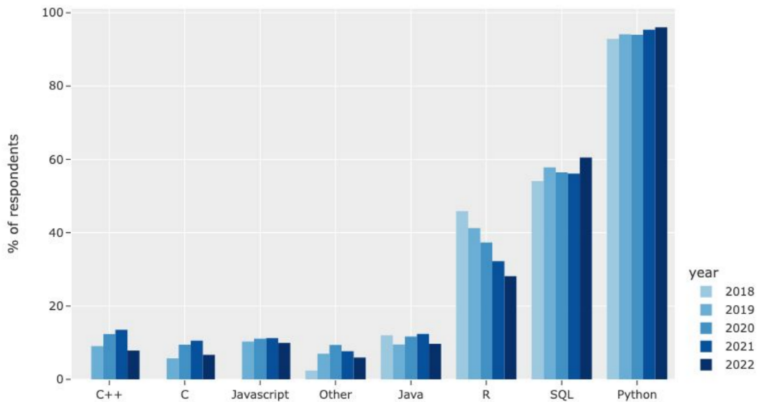


Image source [4]

Common Techniques for Machine Learning

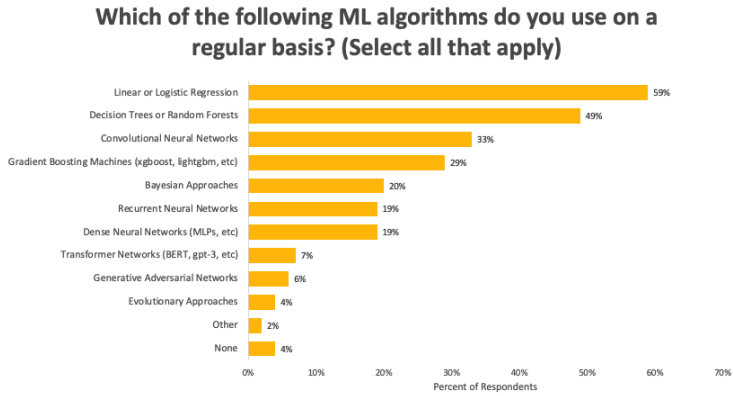


Image source [5]

Types of Learning Problems

- **Supervised Learning:**
  - Learn from labeled data (input-output pairs).
  - Examples: Classification (e.g., spam detection), Regression (e.g., house price prediction).
- **Unsupervised Learning:**
  - Discover patterns in unlabeled data.
  - Examples: Clustering (e.g., customer segmentation), Dimensionality Reduction (e.g., PCA).
- **Reinforcement Learning:**
  - Learn by interacting with an environment to maximize rewards.
  - Examples: Game playing (e.g., AlphaGo), Robotics.

Limitations of Machine Learning

Practical Difficulties

- **Data Dependency:** Garbage in, garbage out.
- **Generalization:** Struggles with unseen data distributions.
- **Bias & Fairness:** Amplifies existing biases.
- **Interpretability:** "Black box" models.

Fundamental Limitations

- **Causality:** Correlation $\neq$ Causation.
- **Goodhart's Law:** When a measure becomes a target, it ceases to be a good measure.
- **Unpredictability:**
  - **Lyapunov instability:** Chaos (e.g., weather).
  - **Feedback loops:** Predictions affect outcome (e.g., finance).
  - **NP-hard problems:** Cryptography.

Bibliography and Suggested Readings

- Hastie, Tibshirani, and Friedman: *The Elements of Statistical Learning*
- Sanchez and Marzban: *All Models Are Wrong: Concepts of Statistical Learning*
- Burkov: *The Hundred-Page Machine Learning Book*
- *Introduction to Machine Learning*, University of Toronto (CSC311)
- *Introduction to Machine Learning*, Stanford University (EE104)
- *Introduction to Machine Learning and Statistical Pattern Classification*, University of Wisconsin–Madison (STAT 451)

Course Outline

- **Datasets and Preparation**
  - Feature Engineering, Data Cleaning, Imbalanced Data
  - Train/Test Splits, Cross-Validation
- **Supervised Learning**
  - Regression
  - Classification
  - Optimization algorithms
- **Unsupervised Learning**
  - Dimensionality reduction
  - Clustering

Chapter 2: Datasets and Data Preparation

- Importance of dataset quality
- How to convert to numerical data
- Feature engineering
- Splitting data sets for model evaluation

The Importance of Data in Machine Learning

Typical challenges for applications (Kaggle 2017):

Image source [6]

Dataset quality

Examples from which the system learns: For each input x, there is an output (or label) y. Both input and output are usually vectors (or matrices); the output can be a single number as well.
The data set should be:

- representative of the problem, diverse
- balanced (distribution over the input and output data, no duplicates)
- outlier-free
- normalized (input and output values between zero and one)
- accurate
- sufficiently large
- no missing data

Sometimes it is possible to **simulate** data, making it easier to increase the size and quality of the data set.

Handling Categorical Data: One-hot-encoding

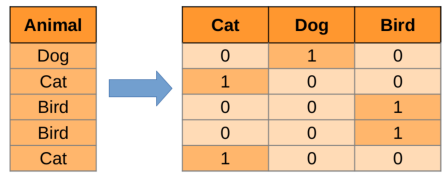
- **Definition**: Converts categorical class labels into binary vectors.
- **Example**:

$$\text{Classes} : \{\text{Cat}, \text{Dog}, \text{Bird}\}$$

$$\text{Cat} \rightarrow [1, 0, 0]$$

$$\text{Dog} \rightarrow [0, 1, 0]$$

$$\text{Bird} \rightarrow [0, 0, 1]$$



Animal
Dog
Cat
Bird
Bird
Cat

Cat	Dog	Bird
0	1	0
1	0	0
0	0	1
0	0	1
1	0	0

- **Purpose**: categorical data

Ordinal Data

What is Ordinal Data?

- Categorical data with a meaningful order or ranking between categories.
- The intervals between categories may not be consistent or meaningful.

Examples:

- Number of Michelin stars for a restaurant
- Ratings: Poor $<$ Fair $<$ Good $<$ Excellent.
- Income brackets: Low $<$ Medium $<$ High.

Handling Ordinal Data:

- Assign numerical values to categories (e.g., Poor $= 1$, Fair $= 2$, etc.).
- Use ordinal encoding to preserve the order.
- Avoid one-hot encoding, as it loses the ordinal relationship.

Use only if the intervals between categories carry consistent meaning, otherwise use one-hot-encoding.

Binning: Converting Numerical to Categorical Data

What is Binning?

- Binning (or bucketing) is the process of converting a continuous feature into categorical bins or buckets.
- Each bin represents a range of values for the continuous feature.

Example:

- Instead of representing age as a continuous feature, we can create discrete bins:
  - $0 \leq \text{age} \leq 5$: Bin 1
  - $6 \leq \text{age} \leq 10$: Bin 2
  - $11 \leq \text{age} \leq 15$: Bin 3

Converting Images to Numerical Data

Black-and-White Image:

- Represented as a 2D matrix.
- Each pixel's intensity is a numerical value (e.g., 0 for black, 255 for white in 8-bit grayscale).

$$\begin{bmatrix} 0 & 128 & 255 \\ 64 & 192 & 128 \\ 255 & 64 & 0 \end{bmatrix}$$

Color Image:

- Represented as a 3D matrix (tensor).
- Three color channels: Red, Green, Blue (RGB).

$$\begin{bmatrix} \text{Red} & \text{Green} & \text{Blue} \\ \begin{bmatrix} 5 & 12 & 14 \\ \vdots & \vdots & \vdots \end{bmatrix} & \begin{bmatrix} 108 & 7 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix} & \begin{bmatrix} 13 & 3 & 5 \\ \vdots & \vdots & \vdots \end{bmatrix} \end{bmatrix}$$

Images are simply numerical data, enabling machine learning and image processing.

Converting Text to Features

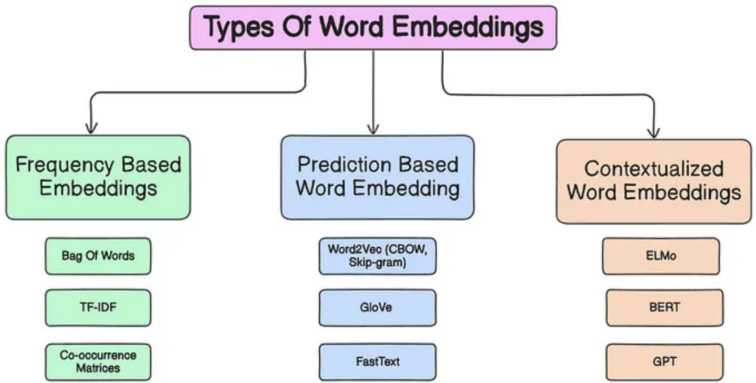


Image source [7]

Tokenizer: Byte Pair Encoding (BPE)

How it works:

1. Start with characters as base vocabulary.
2. Iteratively merge the most frequent adjacent pairs to form subwords
3. Stop when the desired vocabulary size is reached

Why use BPE?

- Handles out-of-vocabulary words by breaking them into subwords
- Balances between word-level and character-level tokenization
- Ability to predefine vocabulary size

Iteration	Sequence	Vocabulary
0	a b a b c a b c	{a, b, c}
1	ab ab c ab c	{a, b, c, ab}
2	ab abc abc	{a, b, c, ab, abc}
3	ababc abc	{a, b, c, ab, abc, ababc}
4	ababcabc	{a, b, c, ab, abc, ababc, ababcabc}

Image source [8]

Bag of Words (BoW) Model

Text as a collection of words, disregarding grammar, order, or context.

Steps to Create a BoW:

1. Tokenize the text into words.
2. Create a vocabulary of unique words across all documents.
3. Count the frequency of each word in each document.

Example:

- Documents:
  - Document 1: "Cats are awesome."
  - Document 2: "Dogs are awesome too."
- Vocabulary: {cats, are, awesome, dogs, too}
- Feature Matrix:

	cats	are	awesome	dogs	too
Doc 1	1	1	1	0	0
Doc 2	0	1	1	1	1

TF-IDF: Term Frequency-Inverse Document Frequency

TF-IDF adjusts the raw word count by considering the importance of words in the context of the entire corpus.

- **Term Frequency (TF):** Measures how frequently a word appears in a document.
- **Inverse Document Frequency (IDF):** Measures how important a word is by considering how often it appears across all documents.

Formula:

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

$$\text{TF} = \frac{\text{No. of repetitions of a word in a document}}{\text{Total no. of words in the document}}$$

$$\text{IDF} = \log\left(\frac{\text{Total no. of documents}}{\text{No. of documents containing the word}}\right)$$

Word2Vec

Word2Vec is a neural network-based word embedding technique that represents words as dense vectors in a continuous vector space.

Neural Network Architecture:

- A shallow neural network with one hidden layer.
- Input: One-hot encoded representation of the word.
- Output: Probability distribution over the vocabulary (via Softmax).
- Hidden layer weights are the word embeddings.

Techniques: Example phrase: "The cat sits on the mat."

- **Skip-Gram:**
  - Predicts context words given a target word.
  - Example: From "cat," predict "the" and "sits."
- **Continuous Bag of Words (CBOW):**
  - Predicts the target word given surrounding context words.
  - Example: From "the" and "sits," predict "cat."

Key Feature: Captures semantic similarity (e.g., "king" - "man" + "woman" = "queen").

Feature Engineering Overview

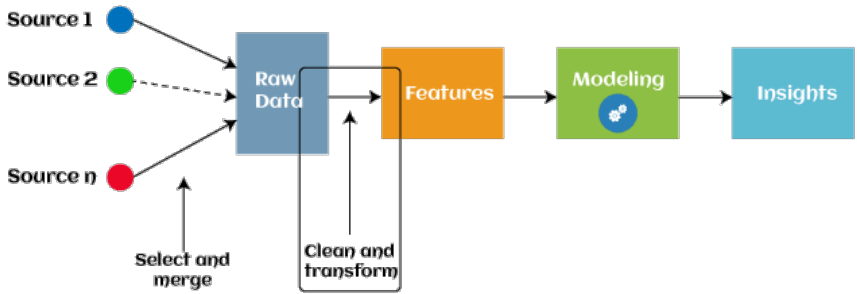


Image source [9]

Feature Engineering

Types of Feature Transforms:

- **Modify individual features:** Replace x_i with transformed feature x_i^{new}.
  - Example: Standardize $x_i^{\text{new}} = \frac{x_i - \mu_i}{\sigma_i}$.
- **Create multiple features from one:** Replace x_i with $(x_i, x_i^2, \ldots, x_i^q)$.
- **Combine multiple features:** Create new features using combinations.
  - Example: $x_i^{\text{new}} = x_k x_l$.

Use Intuition and Verify

1. Train and validate the model with original features.
2. Train and validate the model with engineered features.
3. Compare performances. If the new features improve the model, feature engineering was successful.

Normalization

- **Definition:**
  - Normalization is the process of converting a feature's range of values into a standard range, typically [0, 1].
- **Formula:**

$$\bar{x}^{(j)} = \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}}$$

  where:
  - $\min^{(j)}$: Minimum value of feature j in the dataset.
  - $\max^{(j)}$: Maximum value of feature j in the dataset.
- **Example:**
  - If a feature has a range [350, 1450]:
    - Subtract 350 from all values.
    - Divide by 1100 to scale to [0, 1].
- **Why Normalize?**
  - Ensures features have comparable ranges to avoid domination by larger-scaled features.
  - Prevents numerical overflow or underflow.

Standardization

- **Definition:**
  - Standardization (or z-score normalization) rescales feature values to have a mean (μ) of 0 and a standard deviation (σ) of 1.
- **Formula:**

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}}$$

  where:
  - $\mu^{(j)}$: Mean value of feature j.
  - $\sigma^{(j)}$: Standard deviation of feature j.
- **When to Use Standardization:**
  - Features are distributed close to a normal (bell-curve) distribution.
  - Features may have extreme values (outliers).

Log Transform

Definition:

- Apply the transformation:

$$\phi(u) = \log(u) \quad \text{or} \quad \phi(u) = \log(1 + u) \, (\text{if } u = 0)$$

- Standardize the transformed values for further processing.

When to Use:

- u is positive and spans a wide scale (e.g., web visits, ad views, company capitalization).
- You care about relative differences rather than absolute differences.
- To handle percentage or fractional errors in labels.
- Converts skewed distributions into approximately normal distributions.

Output Transformation:

Recover original output as:

$$\hat{u} = \exp(\hat{\phi})$$

Data Imputation Techniques

- **Replace with Average Value:**
  - Replace missing value with the average (mean) of the feature:

$$\hat{x}^{(j)} \leftarrow \frac{1}{N} \sum_{i=1}^{N} x_i^{(j)}.$$

- **Replace with Out-of-Range Value:**
  - Use a value outside the normal range (e.g., if range is [0, 1], use 2 or -1).
  - Alternatively, use a mid-range value (e.g., set missing values in [-1, 1] to 0).
- **Predict Missing Values:**
  - Treat the missing value as the target in a regression problem.
  - Use remaining features to predict the missing value.
- **Binary Indicator Features:**
  - Add a binary feature indicating whether the value is missing (1: present, 0: missing).
  - Replace the missing value with 0 or another placeholder.

No single technique works best; try several methods and evaluate performance.

Data augmentation

Artificially expand the training dataset by applying transformations on original data

- **Combat Overfitting:** avoids memorizing specific data instances → better generalization
- **Simulate real-world:** prepares models for varied data
- **Compensate for Small Dataset:** adds artificial data

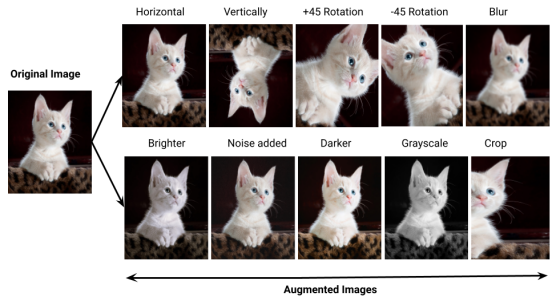


Image source [10]

Handling Imbalanced Datasets

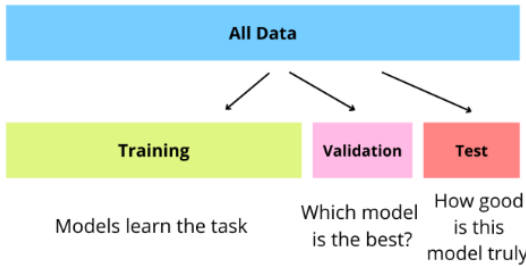
- Minority class is underrepresented in training data.
- Models may favor the majority class, leading to misclassification of the minority class.

Solutions:

- **Cost-sensitive learning:**
  - Assign higher cost to misclassifying minority class examples.
  - Use algorithms that allow class weighting.
- **Resampling techniques:**
  - **Oversampling:** Duplicate or create synthetic examples for the minority class.
  - **Undersampling:** Remove examples from the majority class.
- **Algorithm selection:**
  - Use decision trees which handle imbalanced datasets better.

Training, Validation, and Test Sets

Dividing a dataset into subsets to train, tune, and evaluate the deep learning model



Common split ratios are 60-20-20 or 70-15-15 for training-validation-test.

Data leakage happens when information from the validation / test sets is unintentionally used during training. $\rightarrow$ Overly optimistic performance estimate.

Image source [11]

k-Fold Cross-Validation

- The dataset is divided into k equally sized folds.
- For each iteration:
  - Use $k-1$ folds for training.
  - Use the remaining fold for validation.
- Repeat k times, each fold serving as validation once.
- Average the performance metrics over all k iterations.

Advantages:

- Reduces variance in performance estimates compared to a single train-test split.
- Uses the entire dataset for both training and validation across iterations.

Common Variants:

- k-Fold ($k = 5$ or 10 is common).
- Leave-One-Out Cross-Validation ($k = n$, where n is the number of samples).