

# Chapter 1

## Machine Learning Fundamentals

# Contents

<b>1</b>	<b>Machine Learning Fundamentals</b>	<b>1</b>
1.1	Machine Learning	3
1.2	Defining Learning Tasks	3
1.3	Why we use ML?	4
1.4	Machine Learning Applications	4
1.5	Datasets and Features	4
1.6	Feature Scaling	5
1.7	Types of Datasets	5
1.8	The Task, $T$	6
1.9	The Experience, $E$	6
1.9.1	Supervised Learning	6
1.9.2	Unsupervised learning	7
1.9.3	Reinforcement learning	7
1.9.4	Transfer Learning	7
1.10	Performance Measure, $P$	7
1.11	No Free Lunch Theorem	7
1.12	Training and Testing	7
1.12.1	Independent and Identically Distributed (IID) assumptions	8
1.13	Fitting	8
1.13.1	Resolving Underfitting	8
1.13.2	Resolving Overfitting	8
1.14	Cross-Validation	9
1.14.1	$k$ -fold cross-validation	9
1.14.2	Leave-one-out cross validation (LOOCV)	9
1.15	Parametric Learning	10
1.16	Non-parametric Learning	10
1.17	Model Selection	11
1.18	AIC/BIC	11
1.19	Classification	12
1.20	Classification Applications	15

## 1.1 Machine Learning

“**Learning** is any process by which a system improves **performance** from **experience**.” – Herbert Simon

**Machine Learning** is the **study of algorithms** that –

- improve their performance  $P$
- at some task  $T$
- with experience  $E$ .

A well-defined **learning task** is given by  $\langle P, T, E \rangle$ .

## 1.2 Defining Learning Tasks

$T$  : Playing checkers.

$P$  : Percentage of games won against an arbitrary opponent.

$E$  : Playing practice games against itself.

$T$  : Recognizing hand-written words.

$P$  : Percentage of words correctly classified.

$E$  : Database of human-labeled images of handwritten words.

$T$  : Driving on four-lane highways using vision sensors.

$P$  : Average distance traveled before a human-judged error.

$E$  : A sequence of images and steering commands recorded while observing a human driver.

$T$  : Categorize email messages as spam or legitimate.

$P$  : Percentage of email messages correctly classified.

$E$  : Database of emails, some with human-given labels.

## 1.3 Why we use ML?

- Human expertise does not exist (navigating on Mars).
- Humans can't explain their expertise (speech recognition).
- Models must be customized (personalized medicine).
- Models are based on huge amounts of data (genomics).

## 1.4 Machine Learning Applications

- Recognizing patterns:
  - Facial identities or facial expressions.
  - Handwritten or spoken words.
  - Medical images.
- Generating patterns:
  - Generating images or motion sequences.
- Recognizing anomalies:
  - Unusual credit card transactions.
  - Unusual patterns of sensor readings in a nuclear power plant.
- Prediction:
  - Future stock prices or currency exchange rates.

## 1.5 Datasets and Features

- A **dataset** is a set of data grouped into a **collection** with which developers can work to meet their goals. In a dataset, **rows** represent the number of data points and **columns** represent the features of the dataset.
- The **features of a dataset** are the most **critical aspect** of the dataset, as based on the features of each available data point, will there be any possibility of **deploying models** to find the **output to predict** the features of any new data point that may be added to the dataset.

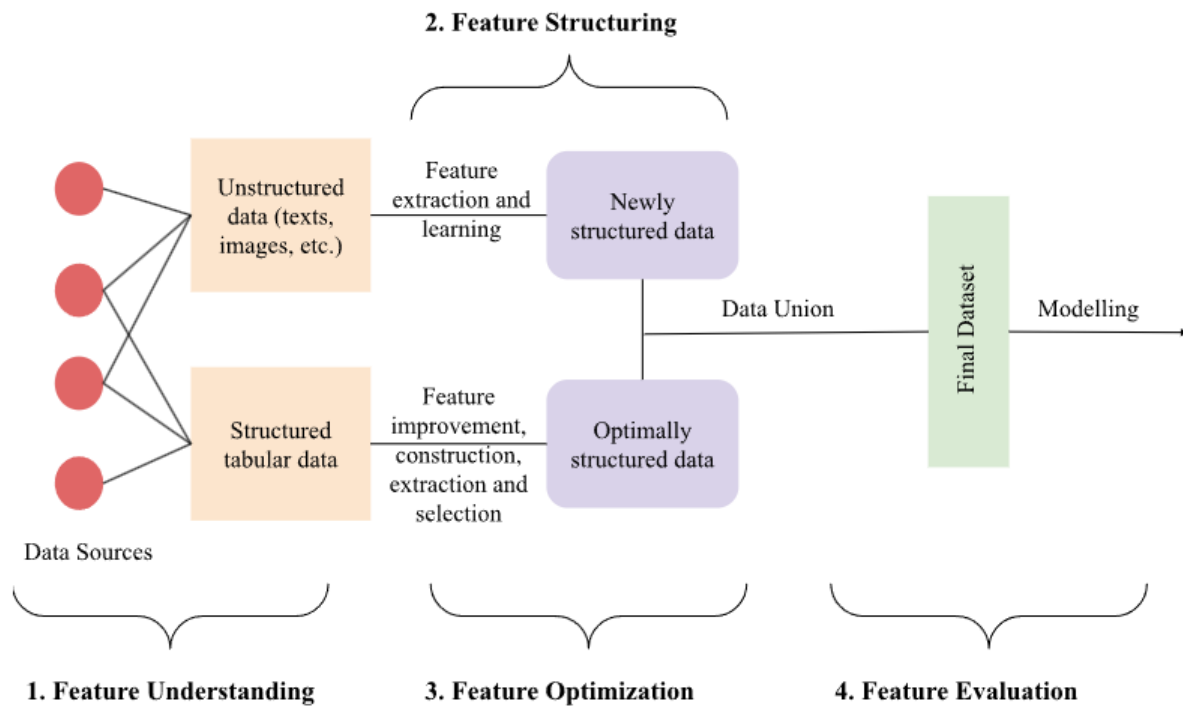


Figure 1.1: Datasets and Features

## 1.6 Feature Scaling

Scale the data to a fixed range  $[0, 1]$ .

- **Normalization:** Rescale the data  $x$  using the mean ( $\mu$ ) and the standard deviation ( $\sigma$ ) of the data.

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (1.1)$$

- **Min-Max Scaling:**

$$x_{minmax} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1.2)$$

## 1.7 Types of Datasets

- Numerical Dataset.
- Categorical Dataset.
- Web Dataset.
- Time series Dataset.

- Image Dataset.
- Ordered Dataset.
- Bivariate Dataset.
- Multivariate Dataset.

## 1.8 The Task, $T$

Tasks are usually described in terms of how the machine learning should process an example:  $x \in \mathbb{R}^n$  where each entry  $x_i$  is a **feature**.

**Classification:** Learn  $f : \mathbb{R}^n \rightarrow \{1, \dots, k\}$

- $y = f(x)$ : assigns input to the category with output  $y$ .
- Example: Object recognition

**Regression:** Learn  $f : \mathbb{R}^n \rightarrow \mathbb{R}$

- Example: Weather prediction, real estate price prediction.

**Transcription:** Unstructured representation to discrete textual form.

- Example: Optical character recognition, speech recognition.

**Machine translation:** Sequence to sequence.

- Example: Translate English to French.

**Synthesis and sampling:** Generate examples that are similar to those in the training data.

- Example: Generate textures for video games, speech synthesis.

## 1.9 The Experience, $E$

### 1.9.1 Supervised Learning

- Experience is a **labeled dataset** (or data points).
- Each **datapoint** has a **label** or target.
- Given  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , learn a **function**  $f(x)$  to predict  $y$  given  $x$ 
  - $y$  is real-valued == **regression**
  - $y$  is categorical == **classification**

### 1.9.2 Unsupervised learning

- Experience is an [unlabeled dataset](#).
- Clustering, learning probability distribution, denoising, etc.

### 1.9.3 Reinforcement learning

- Experience is the interaction with an [environment](#).

### 1.9.4 Transfer Learning

A model trained off of dataset  $D_1$  can be retrained with dataset  $D_2$  to add more knowledge to the model. This means that a model does not have to be trained from scratch.

## 1.10 Performance Measure, $P$

**Accuracy:** The [proportion](#) of examples for which the [model produces](#) the [correct output](#).

**Error rate:** The [proportion](#) of examples for which the [model produces](#) an [incorrect output](#).

**Loss function:** Quantifies the [difference](#) between the [predicted outputs](#) of a machine learning algorithm and the [actual target values](#).

**Generalization** Ability to [perform well](#) on previously [unobserved data](#); e.g., [evaluate](#) the [performance](#) using a [test set](#).

## 1.11 No Free Lunch Theorem

- “[No Free Lunch Theorem](#)”: without having substantive information about the modeling problem, there is [no single model](#) that will always [do better than any other model](#).
- The [goal](#) of machine learning research is not to seek a universal learning algorithm or the absolute best learning algorithm.
- Instead, our goal is to understand what kinds of distributions are relevant to the real world and what kinds of machine learning algorithms perform well on data drawn distributions we care about.

## 1.12 Training and Testing

- [Training data](#): used to [train](#) the machine learning model.
- [Testing data](#): used to determine the performance of the trained model.

### 1.12.1 Independent and Identically Distributed (IID) assumptions

- Examples in each dataset are independent from each other
- Training and testing set are identically distributed; i.e., drawn from the same probability distribution as each other.

## 1.13 Fitting

- **Underfitting**: when the model is **unable** to **obtain** a sufficiently low training error value.
- **Overfitting**: when the **gap** between the **training error** and **test error** is too large.
- **Hypothesis**: the machine's presumption regarding the connection between the input features and the output.

Consider a **hypothesis**  $h$  and its...

- Error rate over training data:  $error_{train}(h)$
- True error rate over all data:  $error_{true}(h)$

Hypothesis  $h$  **overfits** the training error if there is an alternative hypothesis  $h'$  such that

$$\begin{aligned} error_{train}(h) &< error_{train}(h') \\ error_{true}(h) &> error_{true}(h') \end{aligned} \tag{1.3}$$

### 1.13.1 Resolving Underfitting

- **Increasing** model **complexity**.
- Using a different ML **algorithm**.
- Increasing the amount of **training data**.
- **Ensemble methods** to combine multiple methods to create better outputs.
- **Feature engineering** for **creating new model features** from the existing ones that may be more relevant to the learning task.

### 1.13.2 Resolving Overfitting

- **Cross-Validation**: a technique for evaluating ML models by **training several ML models** on **subsets** of the available input data and evaluating them on another subset of the data.
- **Regularization**: a technique where a **penalty term** is added to the **loss function**, discouraging the model from assigning too much importance to individual features.



- **Early-stopping**: stops training when a monitored **metric** has **stopped improving**.
- **Bagging**: **learning multiple models** in parallel and applying **majority voting** to choose the final candidate model.

## 1.14 Cross-Validation

### 1.14.1 $k$ -fold cross-validation

- Divide data into  $k$  folds.
- Train on  $k - 1$  folds, use the  $k^{\text{th}}$  to **measure error**.
- Repeat  $k$  times; use **average error** to measure generalization accuracy.
- Statistically valid and gives good accuracy estimates.

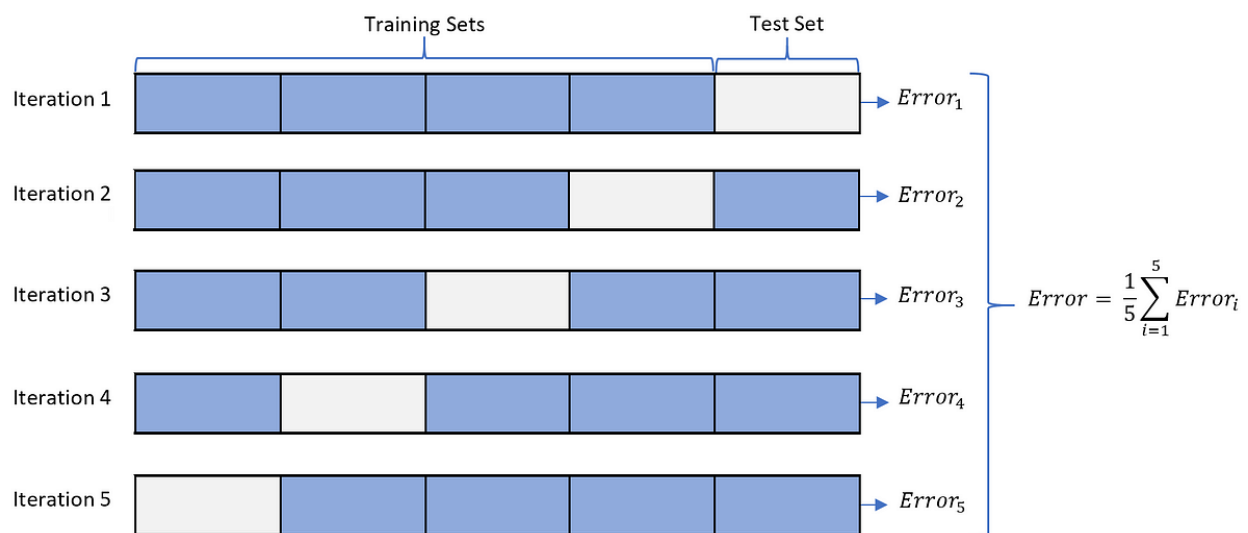


Figure 1.2:  $k$ -fold Cross-Validation

### 1.14.2 Leave-one-out cross validation (LOOCV)

- $k$ -fold cross validation with  $k = N$ , where  $N$  is the number of **data points**.
- Quite accurate, but also **expensive**, since it requires **building  $N$  models**.

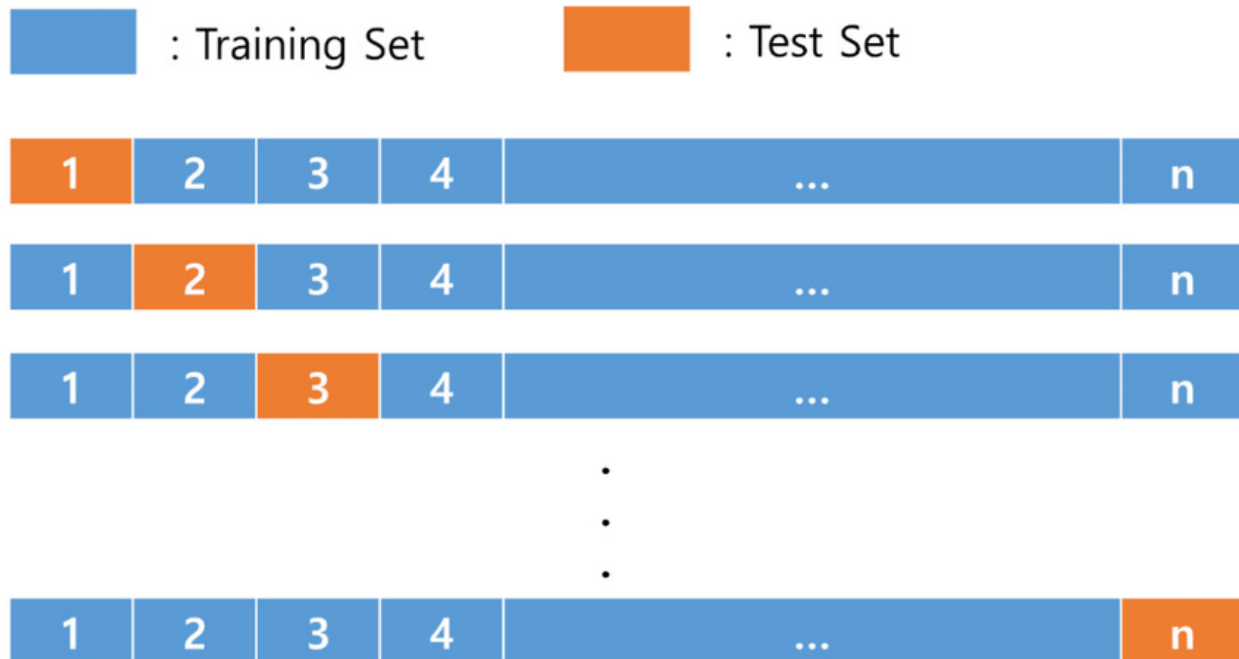


Figure 1.3: LOOCV

## 1.15 Parametric Learning

**Parametric learning algorithms:** make **strong assumptions** about the form of the **mapping function** between the input features and output.

- For example, logistic regression, linear regression, perceptron, naïve bayes, neural network.
- **Benefits** of such models are
  - (a) easier to understand and interpret results
  - (b) very fast to learn from data
  - (c) do not require as much training data
  - (d) can work even if they do not fit the data perfectly.
- However, by pre-emptively choosing a functional form, these methods are highly constrained to the specified form.

## 1.16 Non-parametric Learning

**Non-parametric learning algorithms:** **do not make assumptions** about the form of the **mapping function** between the input features and output.

- For example, SVM,  $k$ -NN,  $k$ -means, decision tree.

- **Benefits** include
  - (a) being capable of fitting a large number of functional forms,
  - (b) there are no assumptions about the underlying... ,
  - (c) can result in higher performance models for prediction.
- However,
  - (a) it requires a lot more training data,
  - (b) takes longer to train,
  - (c) prone to overfitting.

## 1.17 Model Selection

- **Adopting** the **best algorithm** and model for a specific dataset by **assessing** and **comparing different models** to identify the one with the **best results**.

## 1.18 AIC/BIC

**Akaike Information Criterion (AIC)** and **Bayesian Information Criterion (BIC)**: compares different models to choose one that **best fits the data**.

- The goal of both AIC and BIC is to **balance** the **goodness-of-fit** of the model with its **complexity**, in order to avoid overfitting or underfitting.
- Both **AIC** and **BIC** **penalize models** with **large number of parameters** relative to the **size** of the **data**, but BIC penalizes more severely.

$$\begin{aligned}\min AIC &= \min \{2m - 2 \log(L)\} \\ \min BIC &= \min \{m \log(n) - 2 \log(L)\}\end{aligned}\tag{1.4}$$

where  $m$  is the **number** of **model parameters**,  $n$  is the **number** of **data points**, and  $L$  is the **maximum likelihood** of the model.

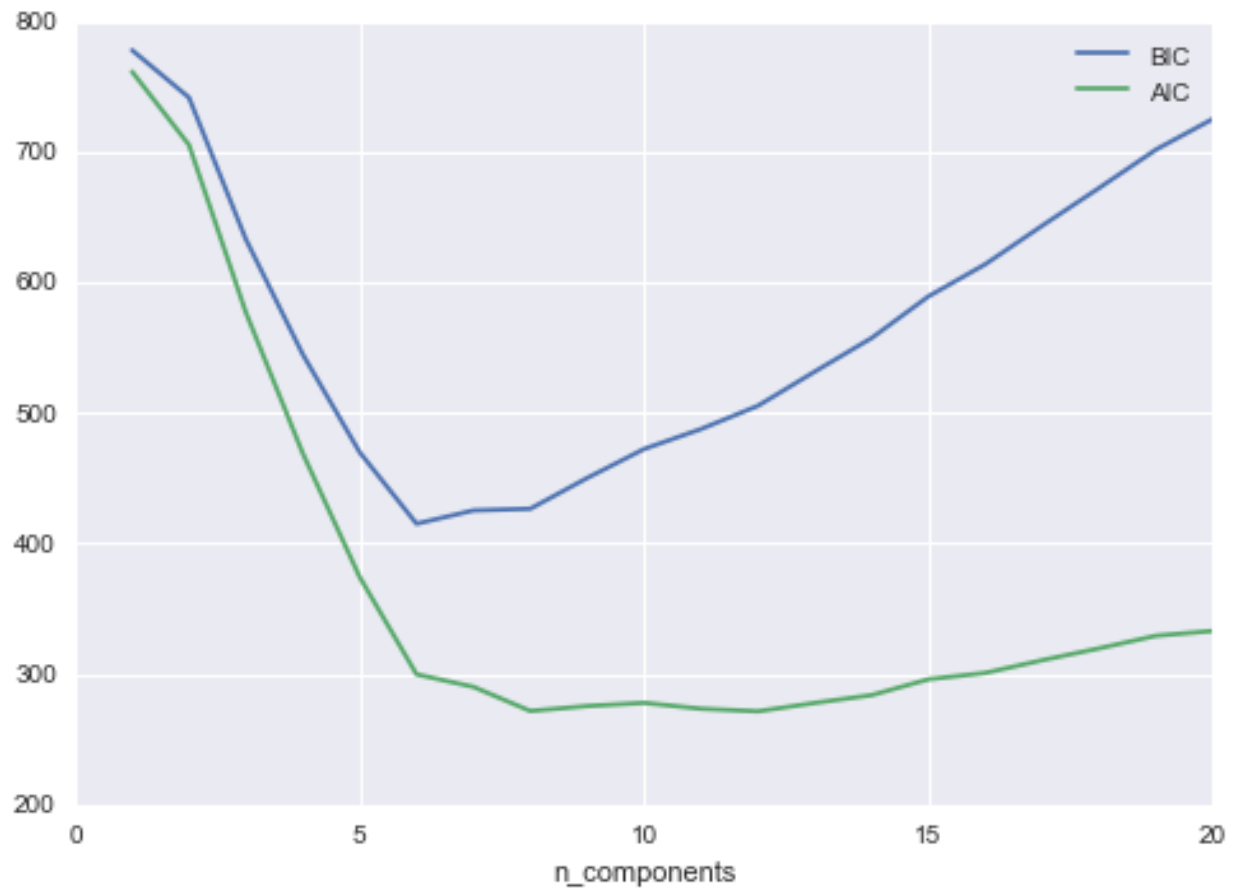


Figure 1.4: AIC/BIC

## 1.19 Classification

**Predictive modeling** problem where a **class label** is **predicted** for a given example of input data.

Types of classification problems:

- Binary classification.
- Multi-class classification.
- Multi-label classification.
- Imbalanced classification.

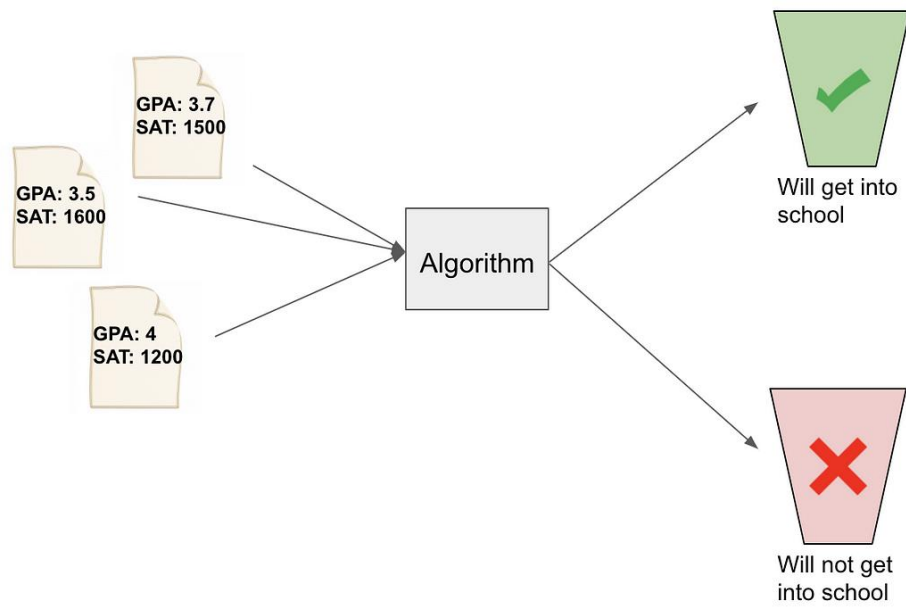


Figure 1.5: Binary Classification

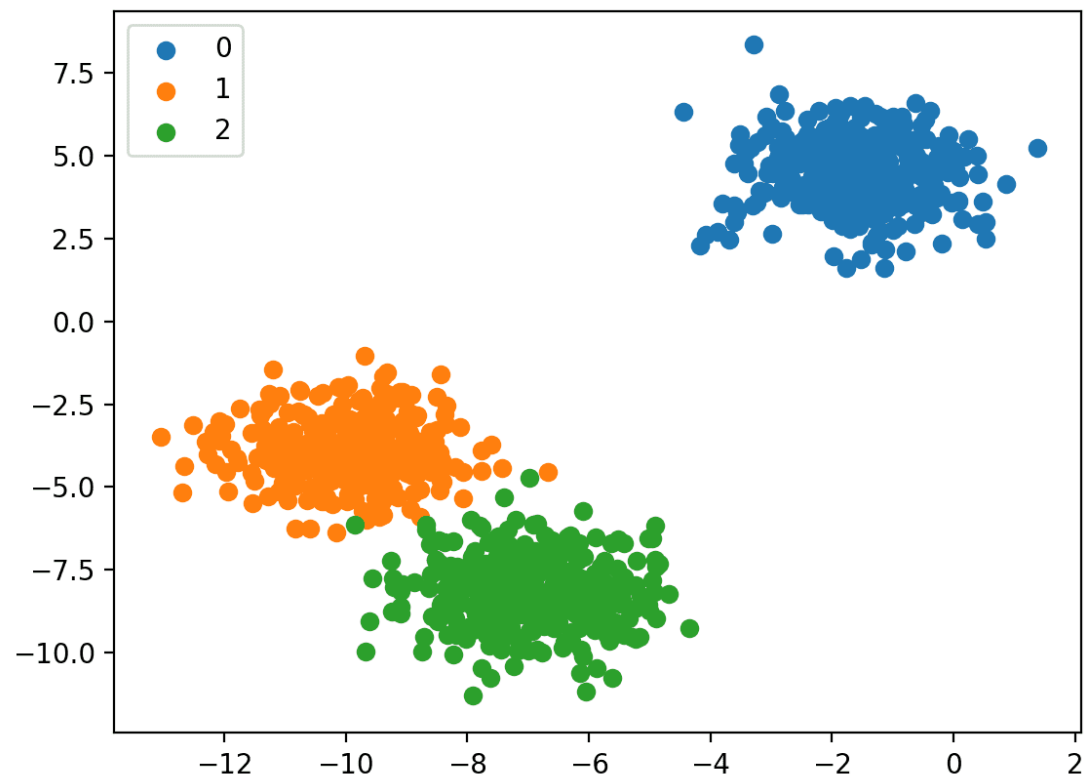


Figure 1.6: Multi-class Classification

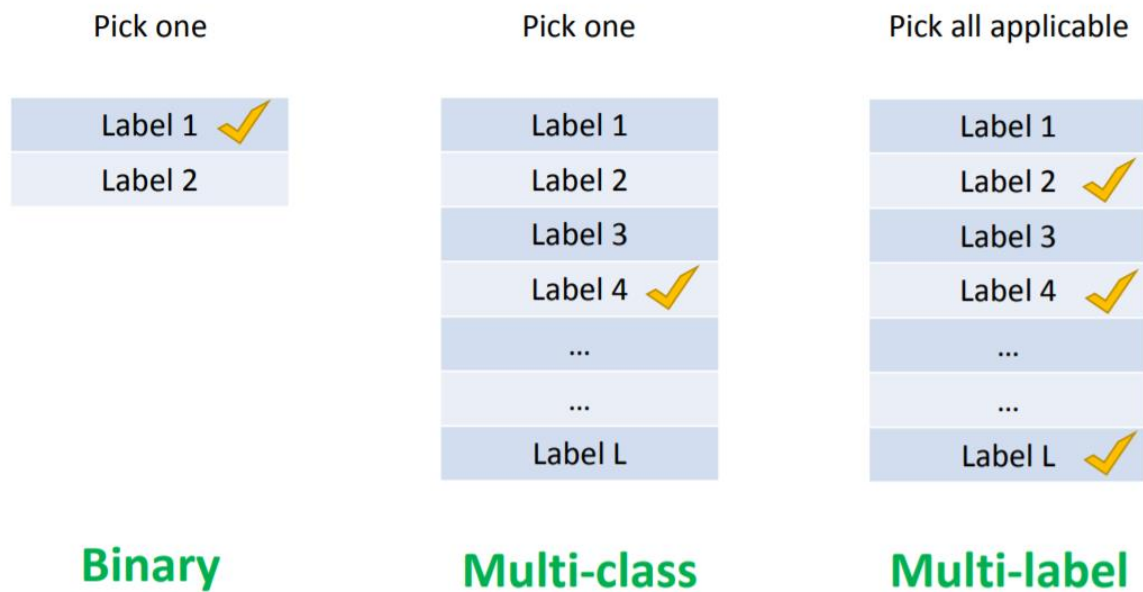


Figure 1.7: Multi-label Classification

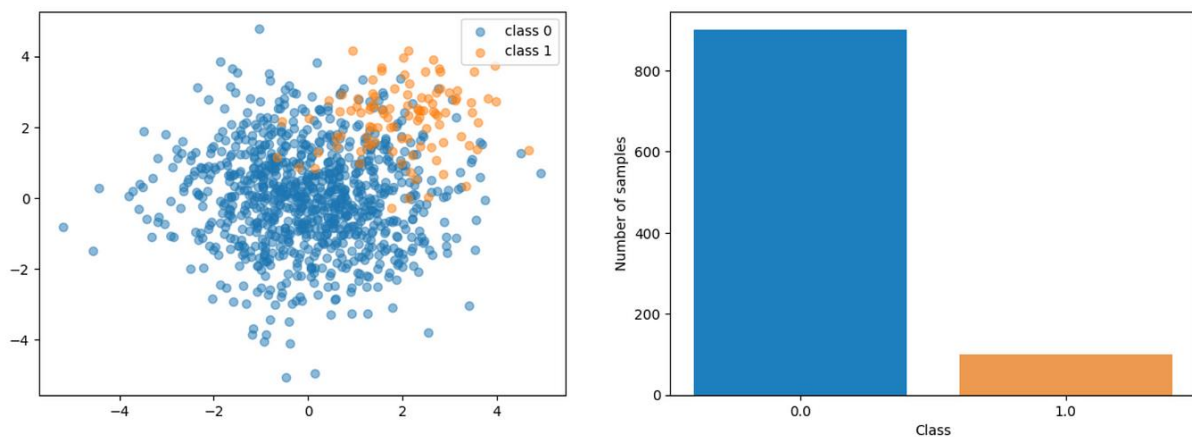


Figure 1.8: Imbalanced Classification

## 1.20 Classification Applications

- Medical diagnosis
  - Features: age, gender, history, symptoms, test results.
  - Label: disease.
- Email spam detection
  - Features: sender-domain, length, images, keywords.

- Label: spam or not-spam.
- Credit card fraud detection
  - Features: user, location, item, price.
  - Label: fraud or legitimate.

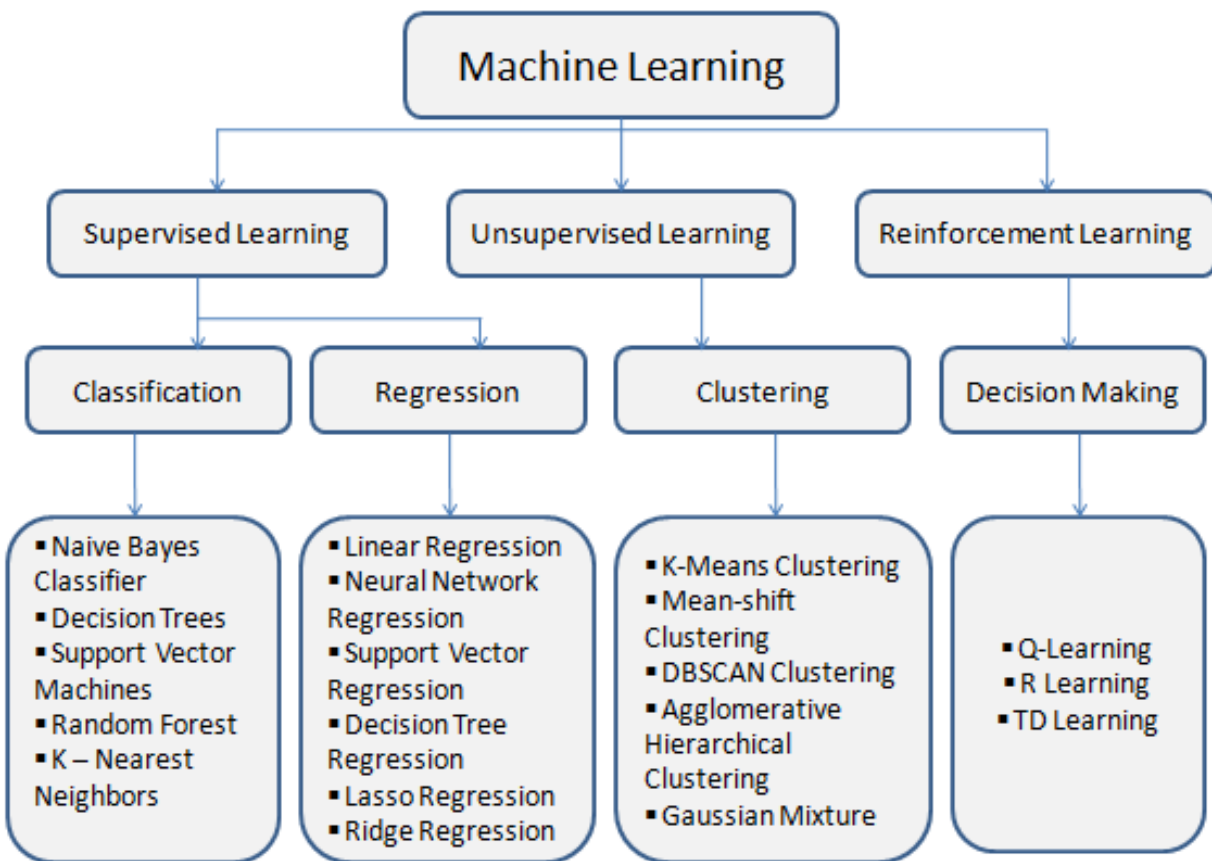


Figure 1.9: Machine Learning Algorithm Tree