# Modeling

**Modeling** is the process of representing a real-world problem in a way that a computer can understand and solve it, in order to **find information or make predictions** in data.

This models are created using **Machine Learning** algorithms. In traditional programming, we give to the computer an input and a program, and it returns an output. In Machine Learning, we give to the computer an **input and an output**, and it returns a program.

- **Predictive Models**: predict the value of a variable of interest based on other variables; **Classifiers** and **Predictors**;
- **Descriptive Models**: describe the relationship between variables in data in a way that is interpretable by humans; **Patterns/Rules** and **Data Partition**;

We can distinguish three **learning techniques**:

- **Supervised Learning**: the model is trained using a set of **labeled** examples, and then it is used to predict the label of new examples;
- **Semi-supervised Learning**: the model is trained using a set of **labeled** and **unlabeled** examples, and then it is used to predict the label of new examples;
- **Unsupervised Learning**: the model is trained using a set of **unlabeled** examples, and then it is used to find patterns in new examples;

According to this categorization, we can present a very simple taxonomy for the most famous **machine learning techniques**:

- **Supervised/Semi-supervised Learning**:
  - **Classification**;
  - **Forecasting**;
- **Unsupervised Learning**:
  - **Clustering**;
  - **Pattern Mining**.

**Anomaly Detection** is a fifth task that can not be classified in this taxonomy, because it can be used in both supervised and unsupervised learning.

---

---

## Classification

**Concept** is something conceived in the mind, an **abstract/generic idea** generalized from particular instances.

**Classification** is the task of automatically learning concepts.

- A **class** is a set of objects that share common properties;
  - This is our **concept to be learned** - the **target**;
- The **dataset** is described by a set of **variables**;
- A **labeled dataset** is a dataset where each object is associated with a class, to be used in **supervised learning**; This dataset can be seen as a matrix:

| $A_1$ | $A_2$ | ... | $A_d$ | $C$ |
|---|---|---|---|---|
| $x_{11}$ | $x_{12}$ | ... | $x_{1d}$ | $c_1$ |
| $x_{21}$ | $x_{22}$ | ... | $x_{2d}$ | $c_2$ |
| ... | ... | ... | ... | ... |
| $x_{n1}$ | $x_{n2}$ | ... | $x_{nd}$ | $c_n$ |

The **model** or **classifier** is a function that maps the **input** to the **output**:

$$f : A_1 \times A_2 \times ... \times A_d \to C$$

- To **learn** a model is to **estimate** the function $f$;
- This is done by using **ML algorithms**, given a labeled dataset - **training set**;
- This means that the algorithm tries to find the function that **minimizes the error** between the **predicted** and the **actual** values.

---

### Classification Tribes

- **Analogizers** - classify an object by comparing it to **similar known objects** - learning by **analogy**;
  - The most famous algorithms are **k-Nearest Neighbors** and **Support Vector Machines**;
- **Bayesians** - classify an object by computing the **probability** of each class given the object - learning by **probabilistic inference**;
  - The most famous algorithms are **Naive Bayes** and **Bayesian Networks**;
- **Symbolists** - classify an object by inferring a set of **logical rules** - learning by **logic inference**;
  - The most famous algorithms are **Decision Trees** and **Rule-based Classifiers**;
- **Connectionists** - classify an object by using a **neural network** - learning by **optimization**;
  - The most famous algorithms are **Multilayer Perceptrons** and **Deep Learning**;
- **Evolutionaries** - classify an object by using an **evolutionary algorithm** - learning by **evolution**;
  - The most famous algorithms are **Genetic Algorithms** and **Genetic Programming**.

---

---

## Evaluation Metrics

- **Objective**: evaluate the **performance of a model**;
- **Criteria**:
  - **Accuracy**;
  - **Simplicity** and **Interpretability**;

  **Occam's Razor**: the simplest explanation is usually the correct one.
- The goal is to **minimize the error rate - error minimization**;
- **Generalization Error** is the error rate of the model in **unseen data** and is expected to decrease as the model is trained with more data;

- However, if the model is **overfitted**, the generalization error will increase;
- This component can be decomposed into **bias** and **variance**, that are **inversely proportional**;
- **Bias** is the **difference** between the **predicted** and the **actual** values;
- **Variance** is the **variability** of the **predicted** values;

- **Underfitting** is when the model is **too simple** and **does not capture the data - High bias and low variance**;
- **Overfitting** is when the model is **too complex** and **captures the noise - Low bias and high variance**;
  - Model starts to **memorize the training data** instead of **learning the concept**;
  - Training error decreases;
  - Test error increases.

**Confusion Matrix** is a matrix that shows the **number of correct and incorrect predictions** for each class.

| | Actual Positives | Actual Negatives |
|---|---|---|
| **Predicted Positives** | True Positives | False Positives |
| **Predicted Negatives** | False Negatives | True Negatives |

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{All}}$$

$$\text{error rate} = 1 - \text{accuracy} = \frac{\text{FP} + \text{FN}}{\text{All}}$$

Accuracy and error rate are **global metrics**. We can also use **local metrics**:

**Recall/Coverage** is the **proportion of actual positives** that was **correctly identified**. Also known as **TP rate**, **sensitivity**, hit rate, or true positive rate:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

**Precision** is the **proportion of predicted positives** that was **correctly identified**. Also known as **positive predictive value**:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

**F-measure** is the **harmonic mean** of **recall** and **precision**:

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$$

**Specificity** is the opposite of recall, and is the **proportion of actual negatives** that was **correctly identified**. Also known as **selectivity**, **true negative rate**, or **TN rate**:

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

$$\text{FP rate} = 1 - \text{specificity}$$

**ROC (Receiver Operating Characteristic) charts** are used to **evaluate the performance** of a classifier, by plotting the **TP rate** against the **FP rate**.

- The **area under the curve** (**AUC**) is a **global metric** that measures the **overall performance** of the classifier - **higher is better**;

---

# Training Strategies

There are several strategies to train a model:

- For several thousand records, we can use **holdout**;
  - The dataset is **randomly divided** into two subsets: **training set** (70%) and **test set** (30%);
  - The model is trained using the training set, and then it is evaluated using the test set;
  - **Validation dataset** is a third dataset used to **tune the parameters** of the model;
- For few thousand records, we can use **cross-validation**;
  - The dataset is **randomly divided** into $k$ subsets of **equal size**;
  - The model is trained using $k-1$ subsets, and then it is evaluated using the remaining subset;
  - This process is repeated $k$ times, and the results are averaged;
- For few records, we can use **leave-one-out**;
  - **One record is used for testing**, and the others are used for training.

---

---

## Forecasting

**Forecasting** is the task of automatically learning a function that **predicts the value of a variable of interest** based on other variables.

- Usually, the data to be forecasted is **time-dependent**, and are called **time series**;
- Against classification, the **target** is a **continuous variable**, and not a class - the result information is called **predictor**;
- After training the predictor, we can apply it to predict the value of the target in **future time steps**;
- Considering a function $f$ that maps the **input** to the **output**, and a function $\hat{f}$ that maps the **input** to the **predicted output**;
  - The best estimation of $\hat{f}$ is the one **closest** to $f$ - **minimizes the square error**;
  - $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$;

There are several different families of forecasting approaches:

- **Regression -** the predictor is a **linear function** of the input variables;
  - The most famous algorithms are **Linear Regression** and **Logistic Regression**;
- **Extrapolation -** the predictor is a **polynomial function** of the input variables;
  - The most famous algorithms are **Polynomial Regression** and **Support Vector Regression**;
- **Markov Models -** the predictor is a **probabilistic function** of the input variables;
  - The most famous algorithms are **Hidden Markov Models** and **Markov Chains**;
- **Neural Networks -** the predictor is a **neural network**;
  - The most famous algorithms are **Multilayer Perceptrons** and **Deep Learning**.