module1 · classification

introduction to 📈 classification

Machine Learning

Machine Learning is a field that grew out of artificial intelligence within computer science. The practice teaches a computer through examples.

Machine Learning is most closely related to statistics.

Classification

Classification is the process of labeling examples of a dataset into defined classes or categories.

A **Training Set** of data exists to learn what the model is trying to predict.

A **Test Set** of data is assigned as images not in the training set purposed to evaluate performance.

In the case of images, each observation is represented by a set of numbers (features) as a vector.

The classification assigned to an observation is referred to as the **label** (often '1' or '0').

Machine Learning requires that the data is initially represented in the correct manner.

Defining **Classification** formally:

A given training set (x_i, y_i) for $1 = n$, a classification model f is created to predict label y for a new x.

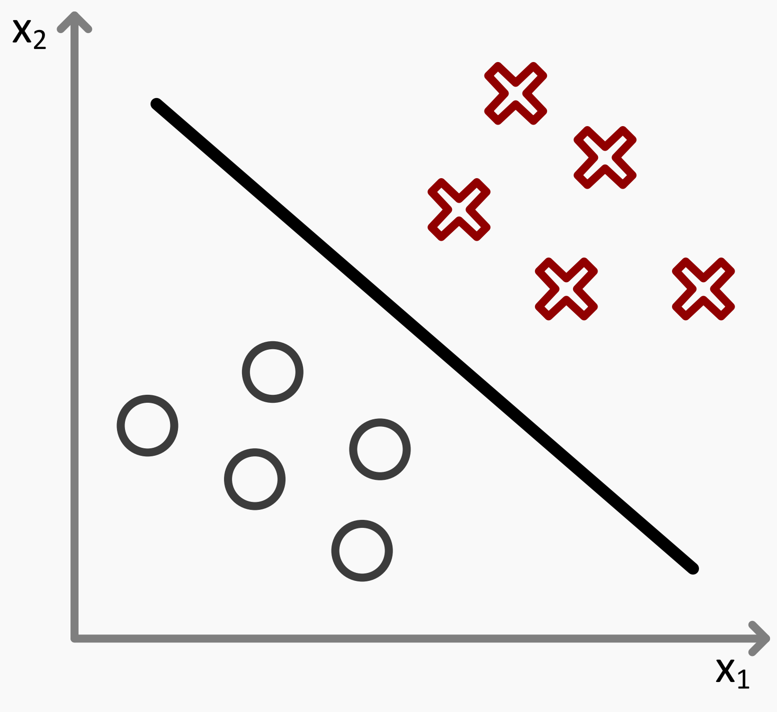
The machine learning algorithm will create the function f.

The predicted y for a new x is simple the sign of $f(x)$.

Classification is designed for Yes/No question → **Binary Classification**

Common Classification Algorithms:

- Logistic Regression (with L1 or L2 regularization)
- Decision Trees/Classification Trees/CART/C4.5/C5.0
- AdaBoost (Boosted Decision Trees)
- Support Vector Machines
- Random Forests
- Neural Networks



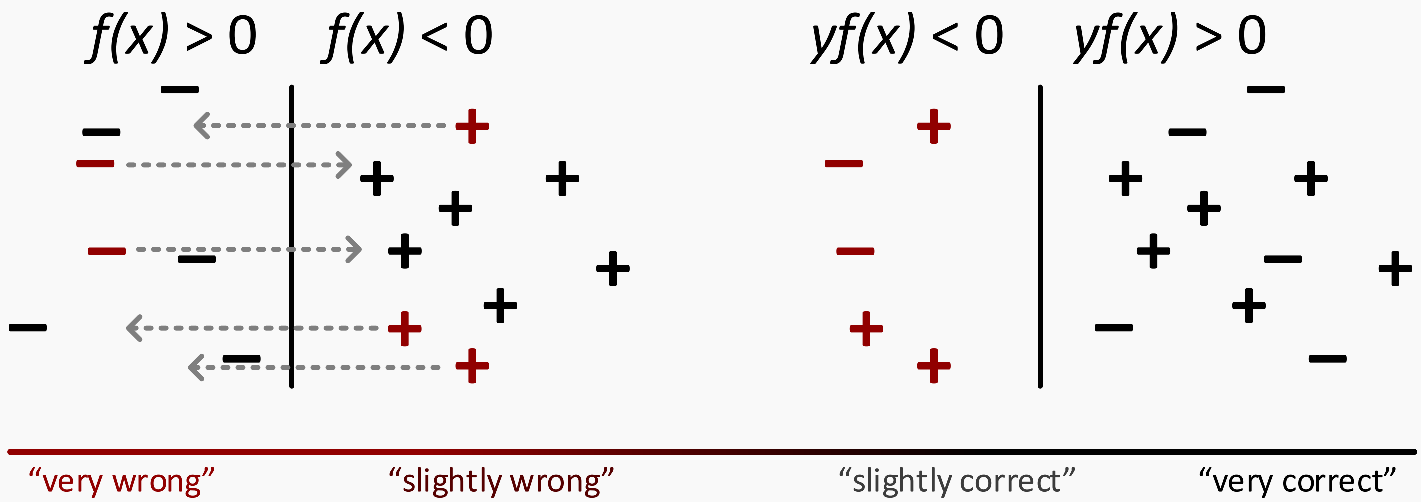
loss functions for classification

Classification error is measured as:

Fraction of times $sign(f(x_i))$ is not y_i:

$$y_i = \frac{1}{n}\sum_{i=1}^{n}[y_i \neq sign(f(x_i))]$$

The expression is geometrically illustrated as follows:



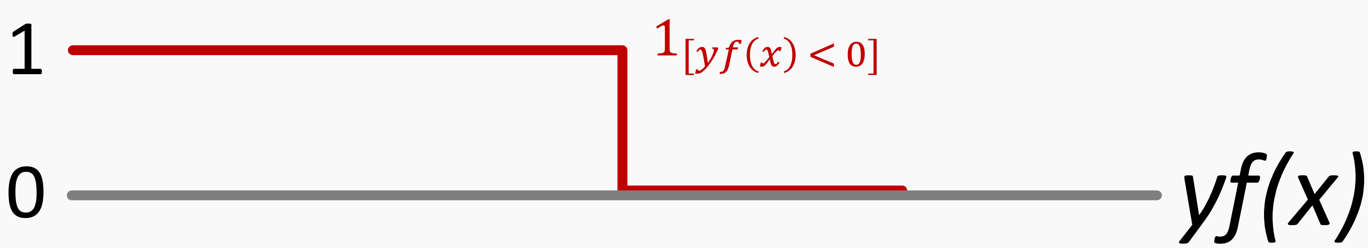
The **Decision Boundary** is the center line dividing examples in each illustration above. In the **left** image, the left cluster represents negative **(-)** function values and the right represents positive function values **(+)**. The examples in **red** represent **misclassified examples**. The **right** image is the effect once all examples are shifted across the decision boundary to cluster them based on correct/incorrect classification predictions opposed to the original values. The new representation in the **right** illustration changes the function from **(+/-)** predictions on each side of the decision boundary to functions where:

- Both $\boldsymbol{y}(\boldsymbol{x})$ **and** $\boldsymbol{f}(\boldsymbol{x})$ are positive **(+)** on the **left** side of the decision boundary;
- Both $\boldsymbol{y}(\boldsymbol{x})$ **and** $\boldsymbol{f}(\boldsymbol{x})$ are negative **(-)** on the **right** side of the decision boundary

Therefore, the right side of the decision boundary represents cases where the sign of $\boldsymbol{f} \neq \boldsymbol{y}$ and will thus be penalized heavily for their values.

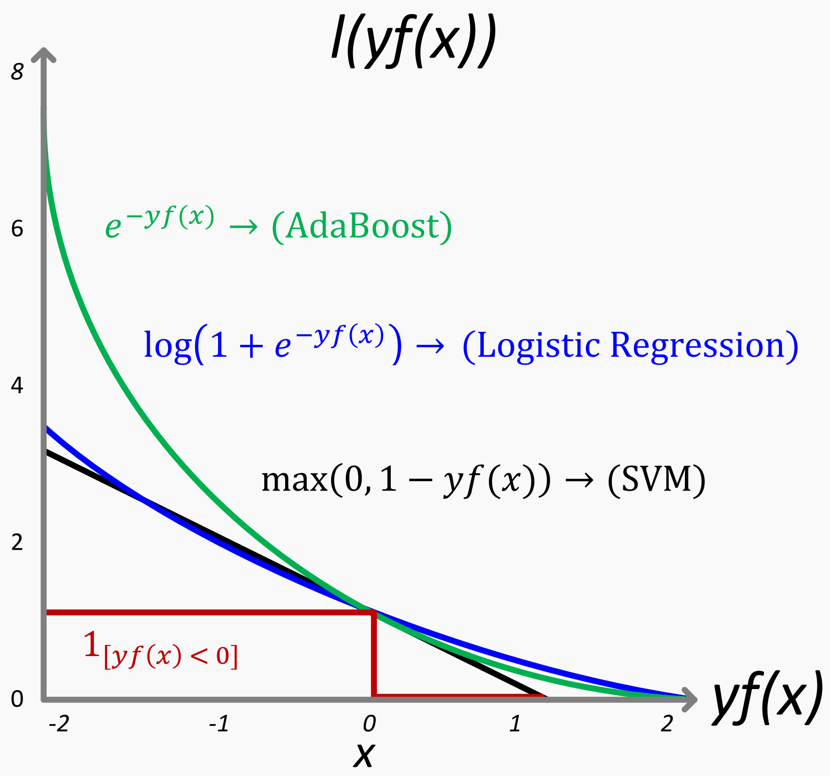
Logical examples of the $yf(x) < 0, yf(x) > 0$:

- If $y > 0$ and $f > 0$, the classification is **correct** and the loss function does **not** penalize.
- If $y < 0$ and $f < 0$, the classification is **correct**, and the loss function does **not** penalize.
- If $y > 0$ and $f < 0$, the classification is **incorrect**, and the loss function **does** penalize.
- If $y < 0$ and $f > 0$, the classification is **incorrect**, and the loss function **does** penalize



Loss Function Intuition:

Fraction of times $sign(f(x_i))$ is not y_i:



$$y_i = \frac{1}{n}\sum_{i=1}^{n}[y_i \neq sign(f(x_i))]$$

$$y_i = \frac{1}{n}\sum_{i=1}^{n}[y_i f(x_i) < 0]$$

$$y_i \leq \frac{1}{n}\sum_{i=1}^{n}\ell(y_i f(x_i))$$

Applying an algorithm attempts to minimize the Loss Function:

$$\min_{\text{models } f} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i f(x_i))$$

However, the above model fails to **generalize** new examples which key in machine learning.