Regression and Classification

Fabien Baradel
PhD Student - INSA Lyon
fabienbaradel.github.io

Linear Regression

Recap

Linear Regression

Data

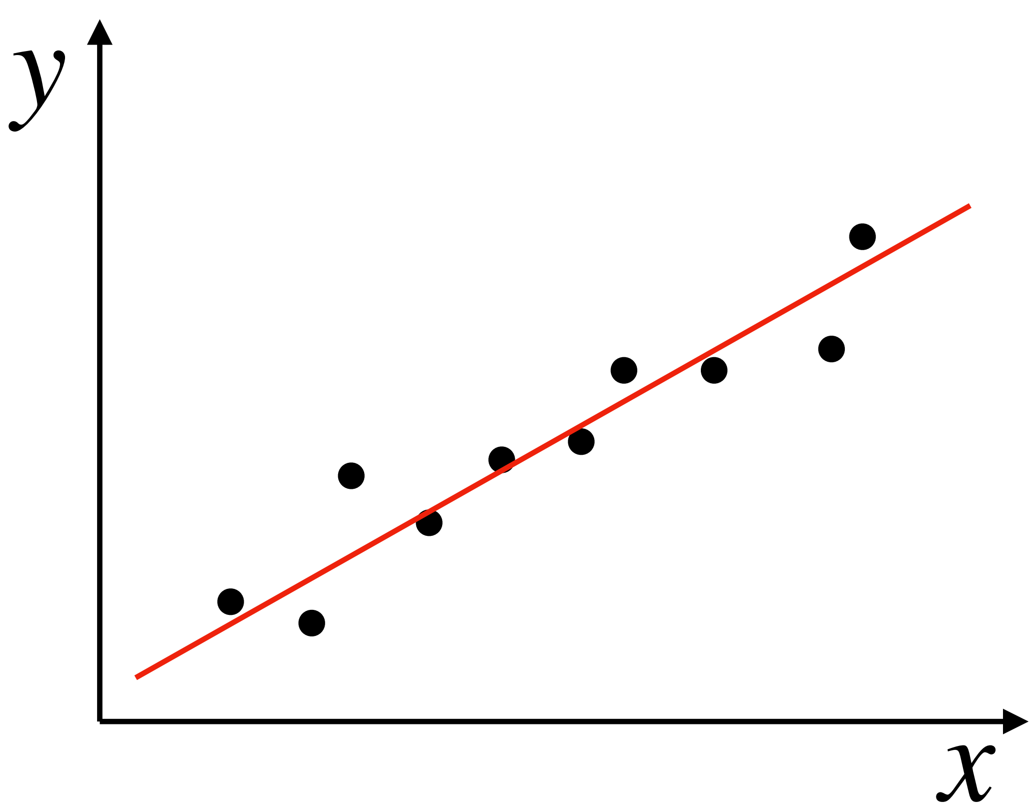
$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

Model

$$f_{w,b}(x) = wx + b$$

Loss function

$$J(w, b) = \frac{1}{N} \sum_{i=1}^{N} (f_{w,b}(x_i) - y_i)^2$$



Optimization by SGD

$$\min_{w,b} J(w, b)$$

Prediction

$$f_{\hat{w},\hat{b}}(x) = \hat{w}x + \hat{b}$$

Multivariate Linear Regression

Data

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$$

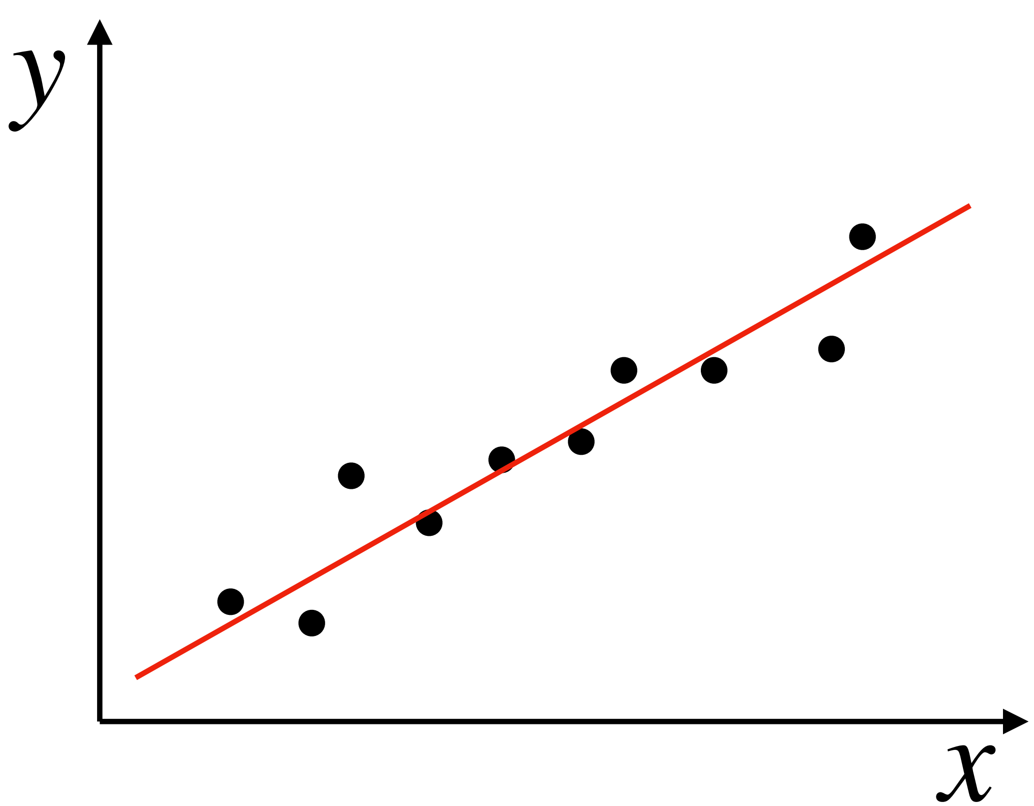
$$\mathbf{x}_i = (1, x_i^1, \ldots, x_i^p)$$

Model

$$f_{\mathbf{w},b}(\mathbf{x}) = b + w^1 x^1 + \ldots + w^p x^p$$

Loss function

$$J(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^{N} (f_{\mathbf{w},b}(\mathbf{x}_i) - y_i)^2$$



Optimization by SGD

$$\min_{\mathbf{w},b} J(\mathbf{w}, b)$$

Prediction

$$f_{\hat{\mathbf{w}},\hat{b}}(x) = \hat{\mathbf{w}} x + \hat{b}$$

Supervised Learning

Overfitting

Define a ML problem

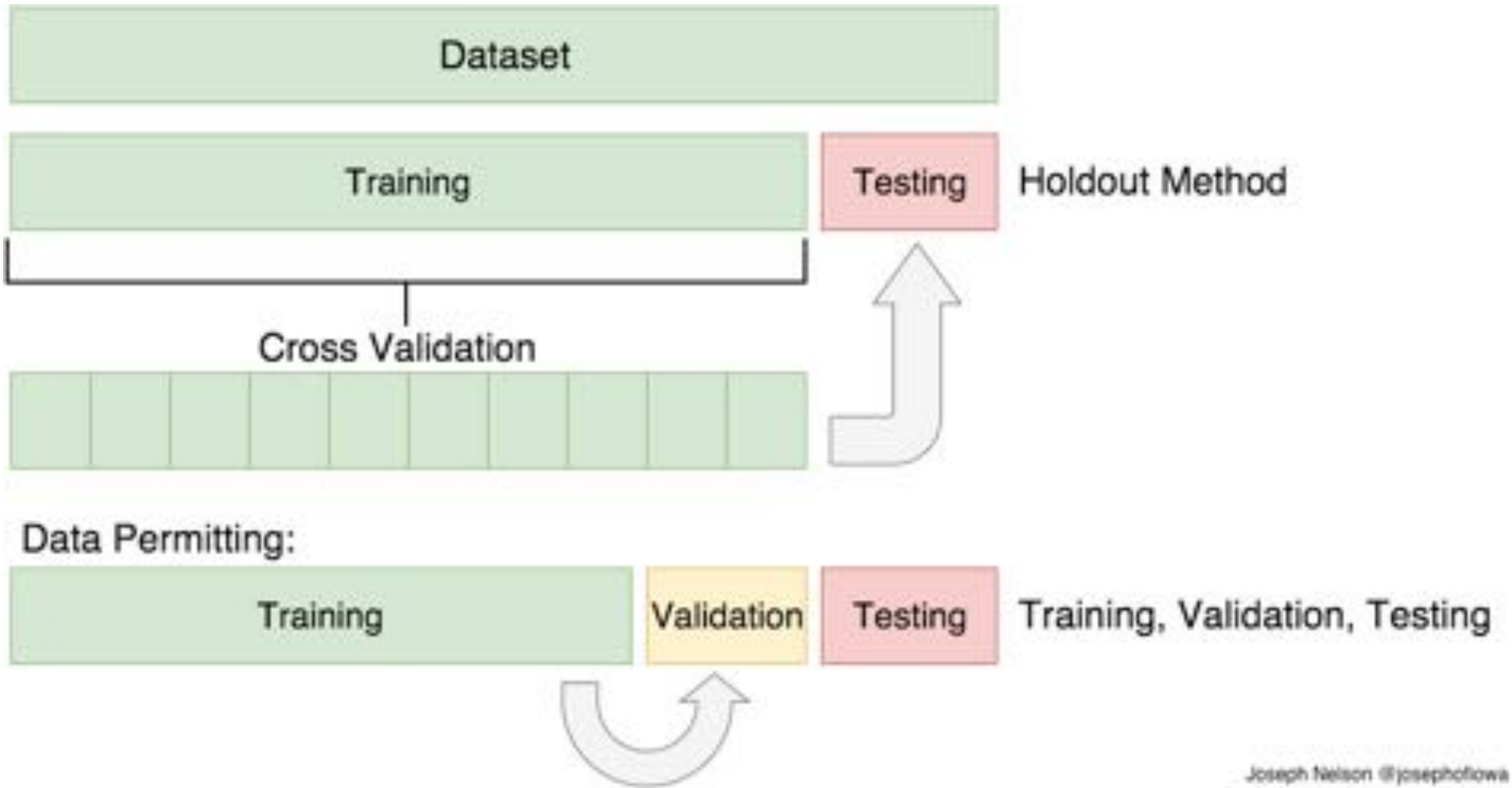
GOAL: Generalization !

1) Training set
2) Validation set
3) Test set

Shuffle dataset
70 % - 15 % - 15 %
Hyper parameters on val while training on train

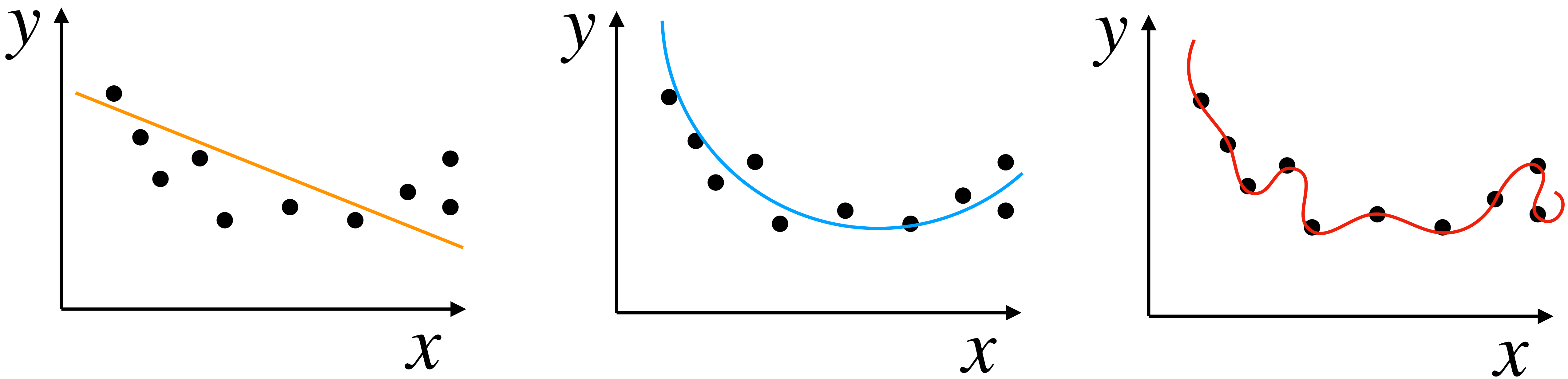
k-cross fold validation
And at the end only get the performance on the TEST set

Three sets



https://towardsdatascience.com/train-test-split-and-cross-validation-in-python-80b61beca4b6

Underfitting and Overfitting



Underfitting   Good fit   Overfitting

Regularization

Build less complex model

L2 normalization

$$\min_{w,b} \frac{1}{N} \sum_{i=1}^{N} ((wx_i + b) - y_i)^2 + C\|w\|^2$$

hyperparameter to tune

L1 normalization

$$\min_{w,b} \frac{1}{N} \sum_{i=1}^{N} ((wx_i + b) - y_i)^2 + C|w|$$

L1 + L2 = Elastic Net

Supervised Learning

Tips and Tricks

Numerical and Categorical Variable

One-hot-encoding:
From categorical to numerical

$$
\begin{aligned}
red &= [1, 0, 0] \\
yellow &= [0, 1, 0] \\
green &= [0, 0, 1]
\end{aligned}
$$

Binning:
From numerical to categorical

Normalization - Standardization

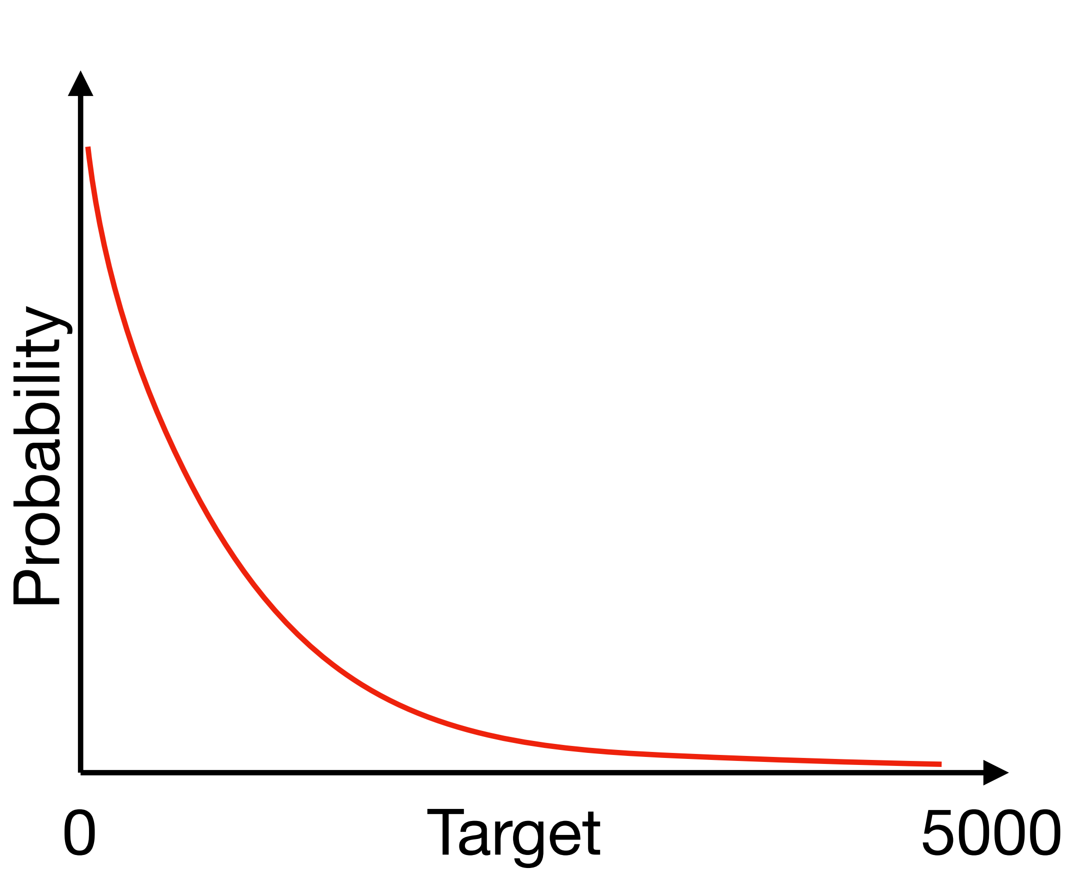
Normalization (0-1)

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

Z-score

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

Target transformation



Log rescaling

Interactions between variables

Inductive bias with expert knowledge

Indicator Variables:
Threshold (e.g. age >=21)

Interaction Features:
Sum
Difference
Product
Quotient

Exercises

Regression on real datasets

Supervised Learning

Classification

Logistic Regression

$$D = \{(x_i, y_i)\}_{i=1}^{N} \qquad y \in \{0,1\}$$

Mapping we want to learn

$$y = f^*(x)$$

Our modeling

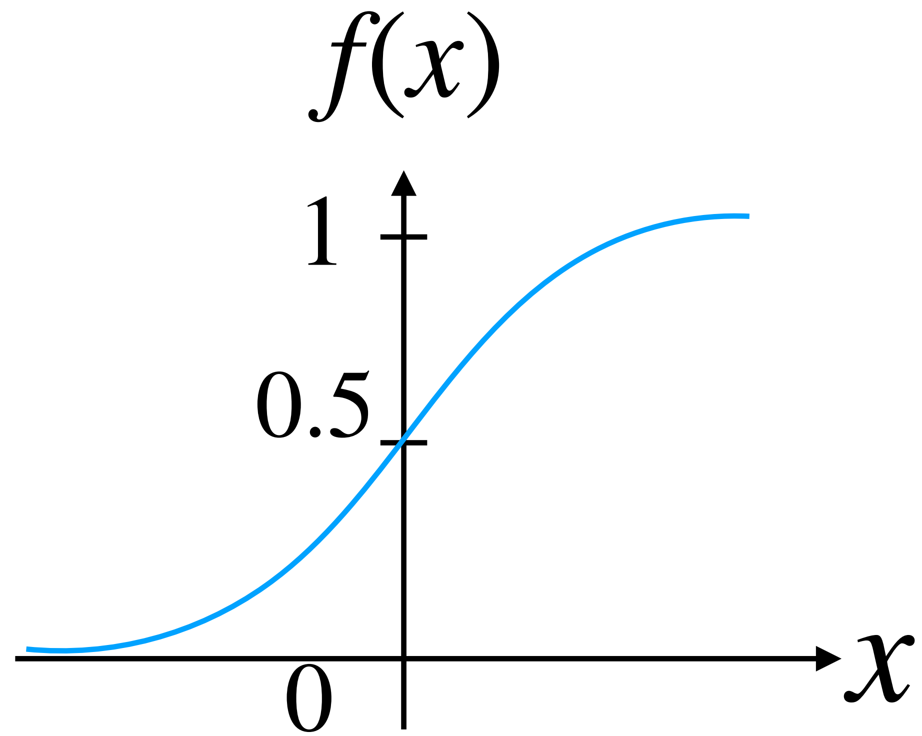
$$y = f_{w,b}(x)$$

How to predict 0 or 1 ?

Logistic Regression

Sigmoid function

$$f(x) = \frac{1}{1 + e^{-x}}$$



Model $$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}}$$

0.5 is the threshold value !

Loss function

Model

$$f_{w,b}(x) = \frac{1}{1 + e^{-(wx+b)}} = p_{w,b}(x)$$

Loss function

$$J(w, b) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p_i) + (1 - y_i)\log(1 - p_i)$$

Minimization of the negative maximum likelihood

Fully differentiable and parameters can be estimated by SGD

?

Confusion Matrix

		Actual Class	
		1	0
Predicted Class	1	True Positive (TP)	False Negative (FN)
	0	False Positive (FP)	True Negative (TN)

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{F-mesure} = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Exercises

Classification on real datasets