BIG DIVE

DATA SCIENCE & ANALYTICS

Machine Learning

André Panisson

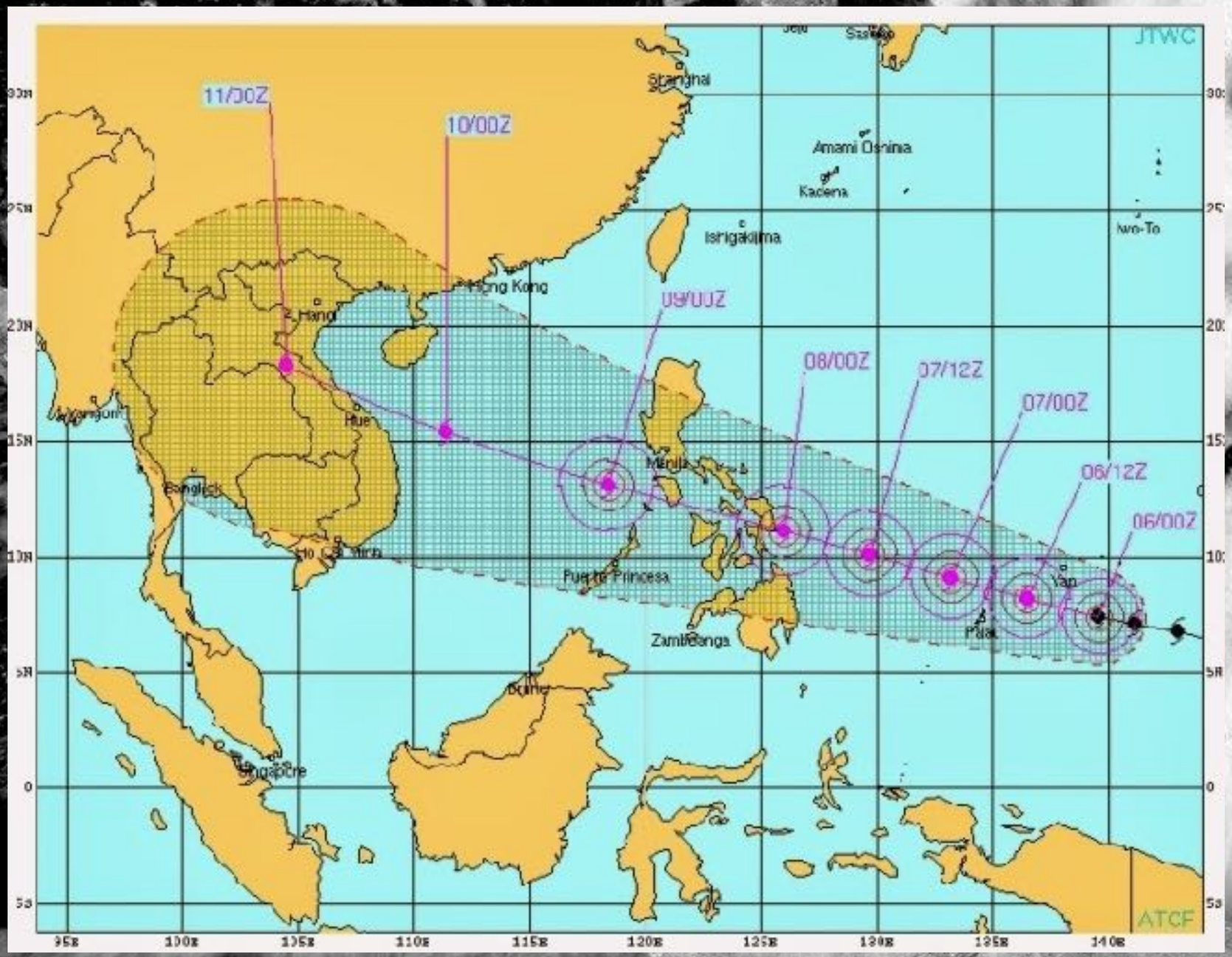
Organized by top ix torino piemonte internet exchange

Designed for INTESA SANPAOLO

In collaboration with arizon technology consulting, ISI Foundation, TODO

what is a “model” ?

- mathematical model
- statistical model
- generative model
- machine learning model
- descriptive model
- dynamical model
- agent-based model
- predictive model (of the future)
- predictive model (of unknown features)


JTWC
11/00Z
10/00Z
09/00Z
08/00Z
07/12Z
07/00Z
06/12Z
06/00Z
Shanghai
Amami Oshima
Kadena
Ishigakijima
Iwo-To
Hong Kong
Hanoi
Yangon
Hue
Bangkok
Manila
Puerto Princessa
Zamboanga
Palau
Yap
Brunei
Singapore
ATCF
95E
100E
105E
110E
115E
120E
125E
130E
135E
140E

THE LEARNING PROBLEM

Metaphor: Credit approval

Applicant information:

age	23 years
gender	male
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000
...	...

Approve credit?

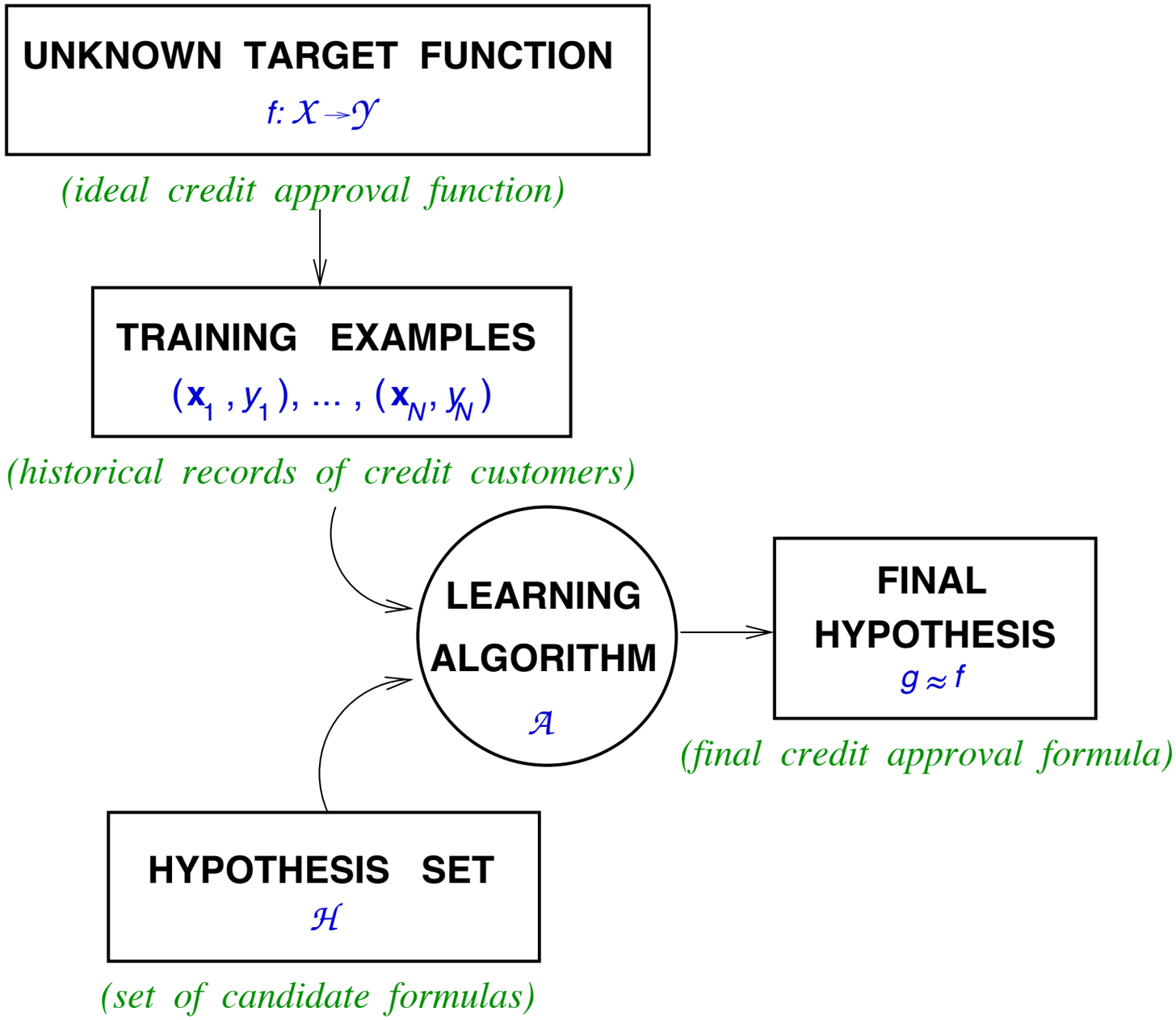
Components of learning

Formalization:

- Input: $\mathbf{x}$ (*customer application*)
- Output: y (*good/bad customer?*)
- Target function: $f : \mathcal{X} \to \mathcal{Y}$ (*ideal credit approval formula*)
- Data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$ (*historical records*)

$\downarrow \quad \downarrow \quad \downarrow$

- Hypothesis: $g : \mathcal{X} \to \mathcal{Y}$ (*formula to be used*)


UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \to \mathcal{Y}$
(ideal credit approval function)
TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$g \approx f$
(final credit approval formula)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate formulas)

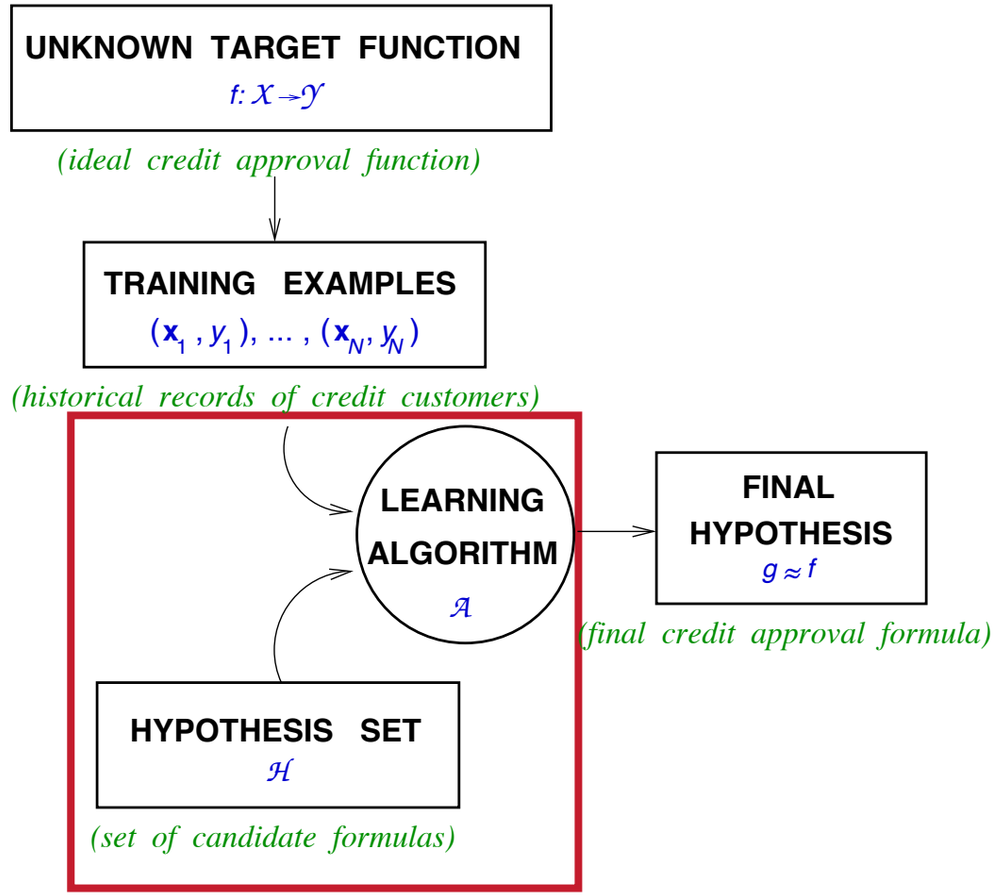
The 2 components of the learning problem:

- The Hypothesis Set

$$\mathcal{H} = \{h\} \quad g \in \mathcal{H}$$

- The Learning Algorithm $\mathcal{A}$

Together, they are referred as the **Learning Model**



A simple hypothesis set - the 'perceptron'

For input $\mathbf{x} = (x_1, \cdots, x_d)$ 'attributes of a customer'

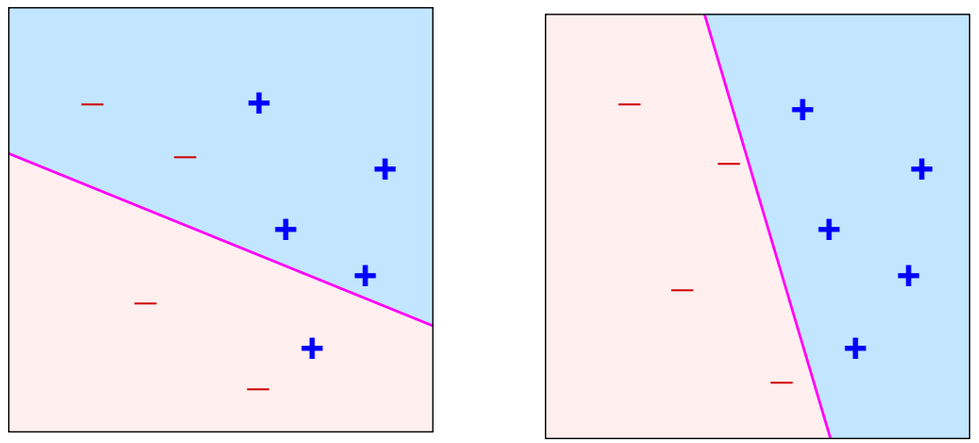
$$\text{Approve credit if} \quad \sum_{i=1}^{d} w_i x_i > \text{threshold},$$

$$\text{Deny credit if} \quad \sum_{i=1}^{d} w_i x_i < \text{threshold}.$$

This linear formula $h \in \mathcal{H}$ can be written as

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right)$$

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i\, x_i\right) + \quad w_0 \quad\right)$$



Introduce an artificial coordinate $x_0 = 1$:

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i\, x_i\right)$$

In vector form, the perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^{\mathsf{T}}\mathbf{x})$$

PLA - The Perceptron Learning Algorithm

The perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\mathsf{T}\mathbf{x})$$

Given the training set:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

pick a misclassified point:

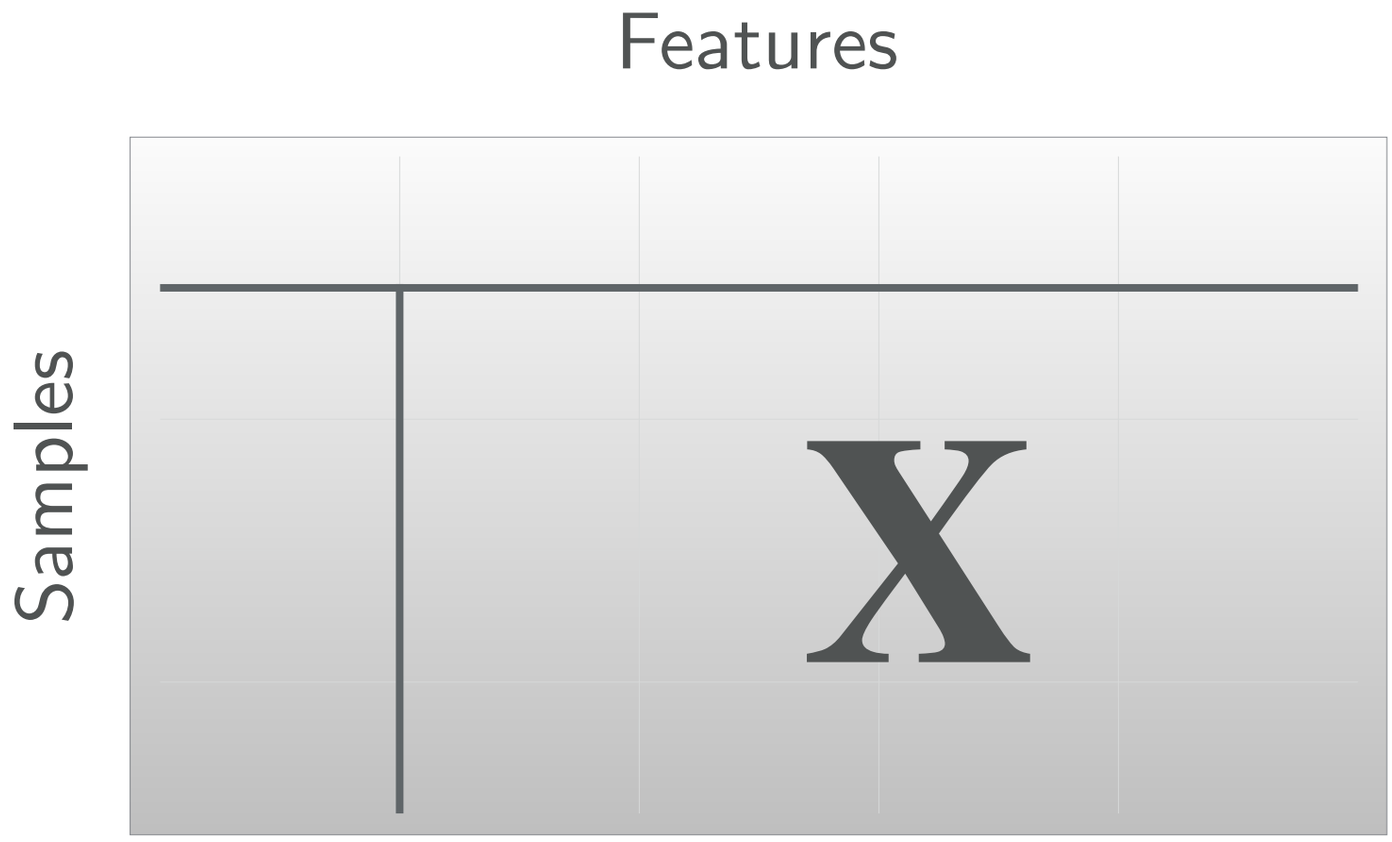
$$\text{sign}(\mathbf{w}^\mathsf{T}\mathbf{x}_n) \neq y_n$$

and update the weight vector:

$$\boxed{\mathbf{w} \leftarrow \mathbf{w} + y_n\mathbf{x}_n}$$

BASIC MACHINE LEARNING CONCEPTS

DATASET



A dataset is composed of

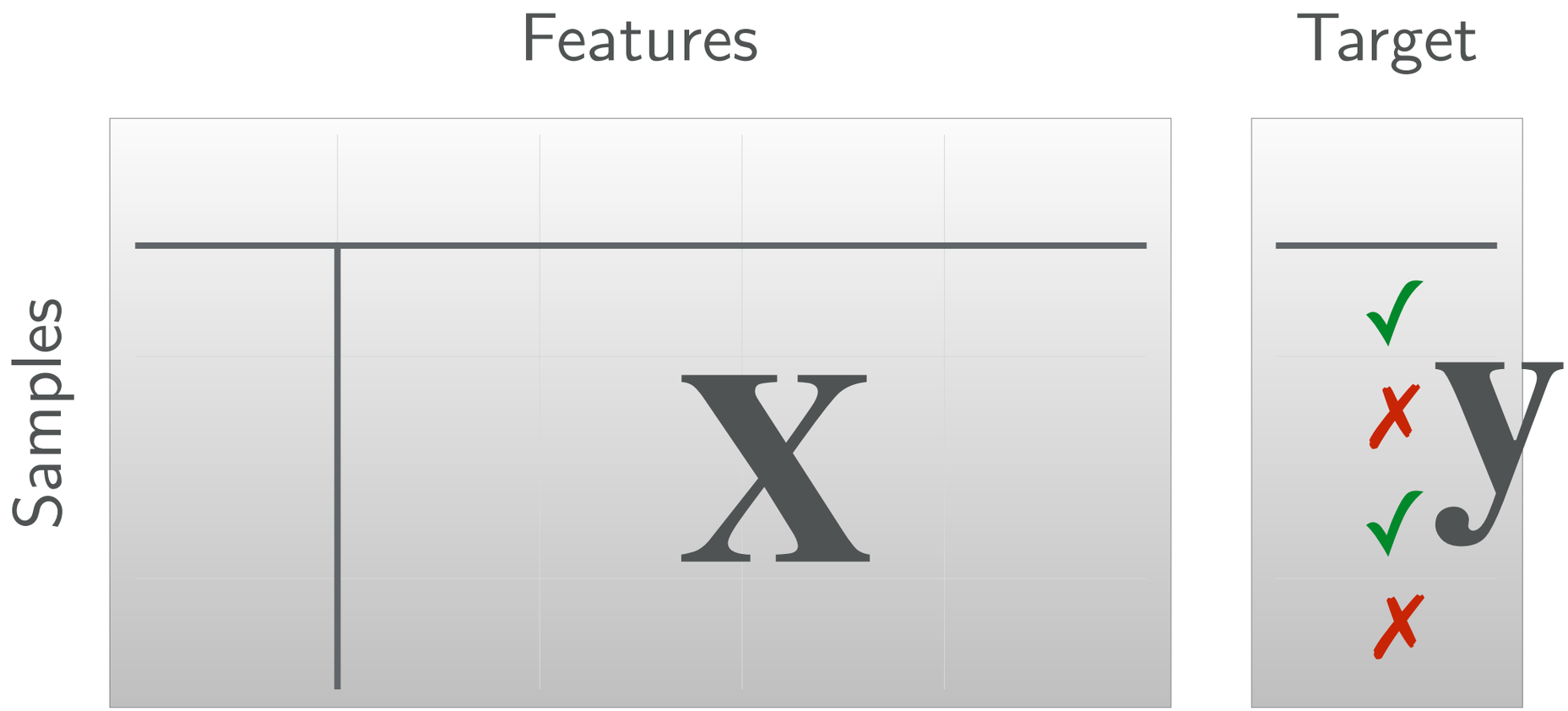
N Samples

D Features

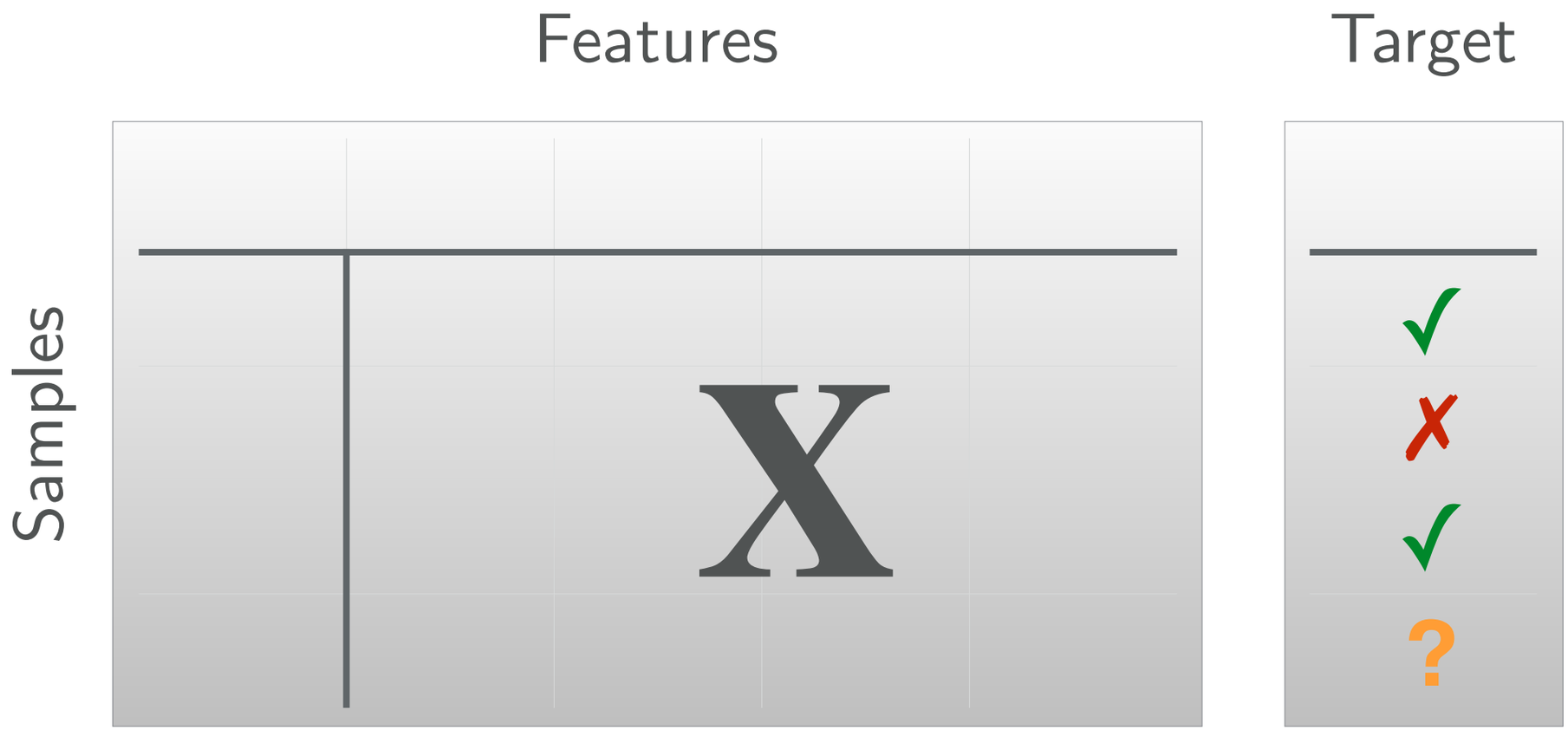
X is also known as the **feature matrix**

D is also known as the **dimensionality** of the dataset

DATASET



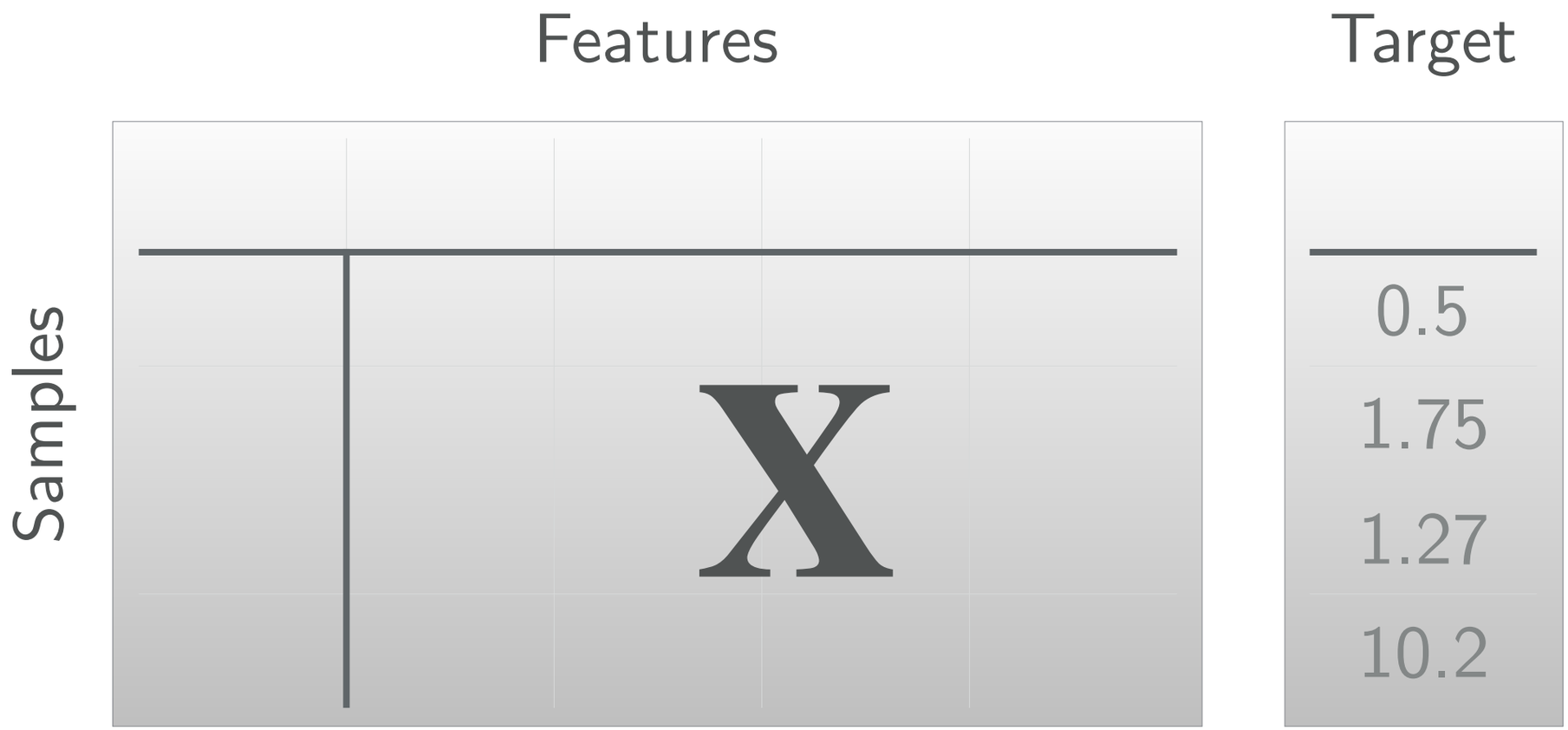
DATASET



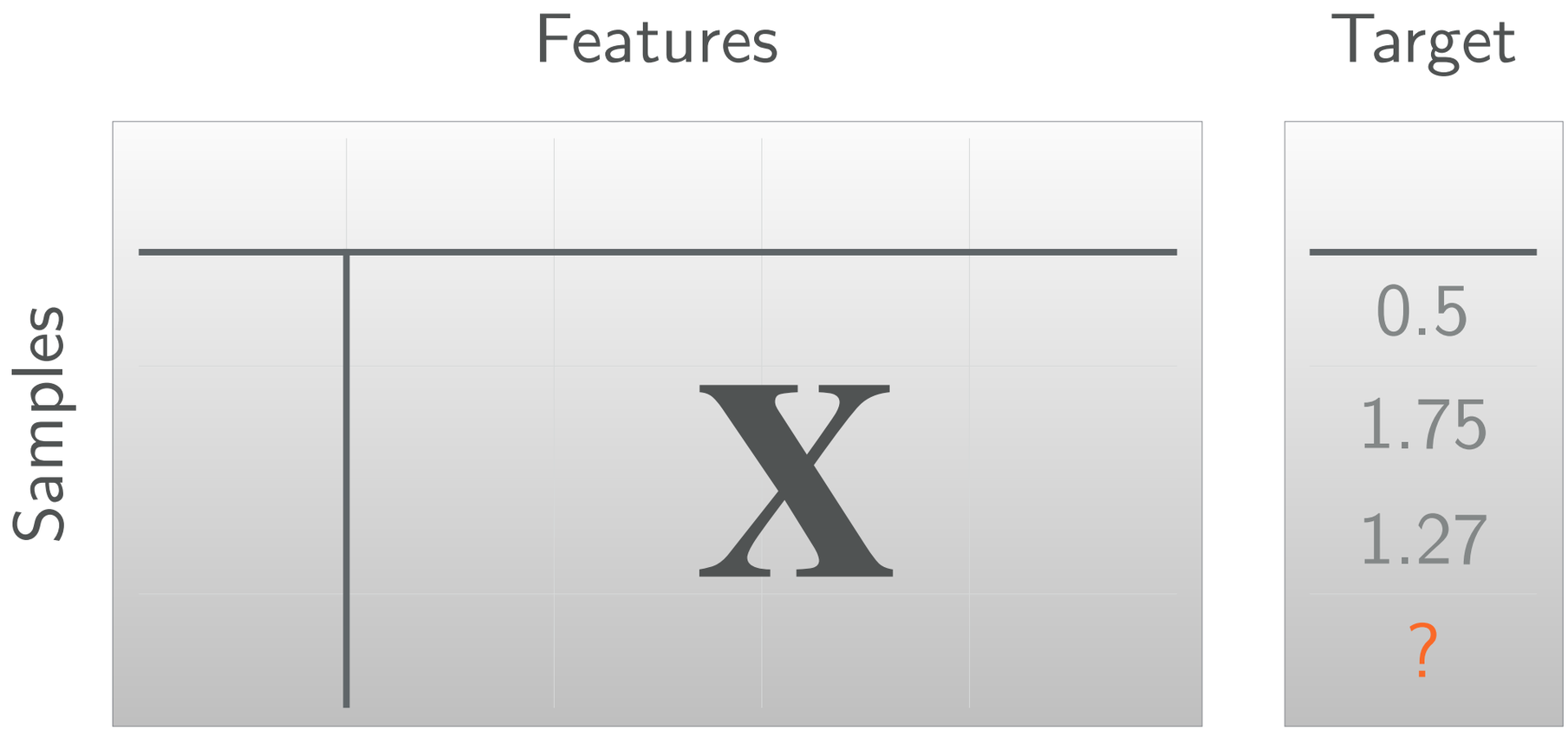
The target is a set of classes:

Supervised Learning, Classification Problem

DATASET



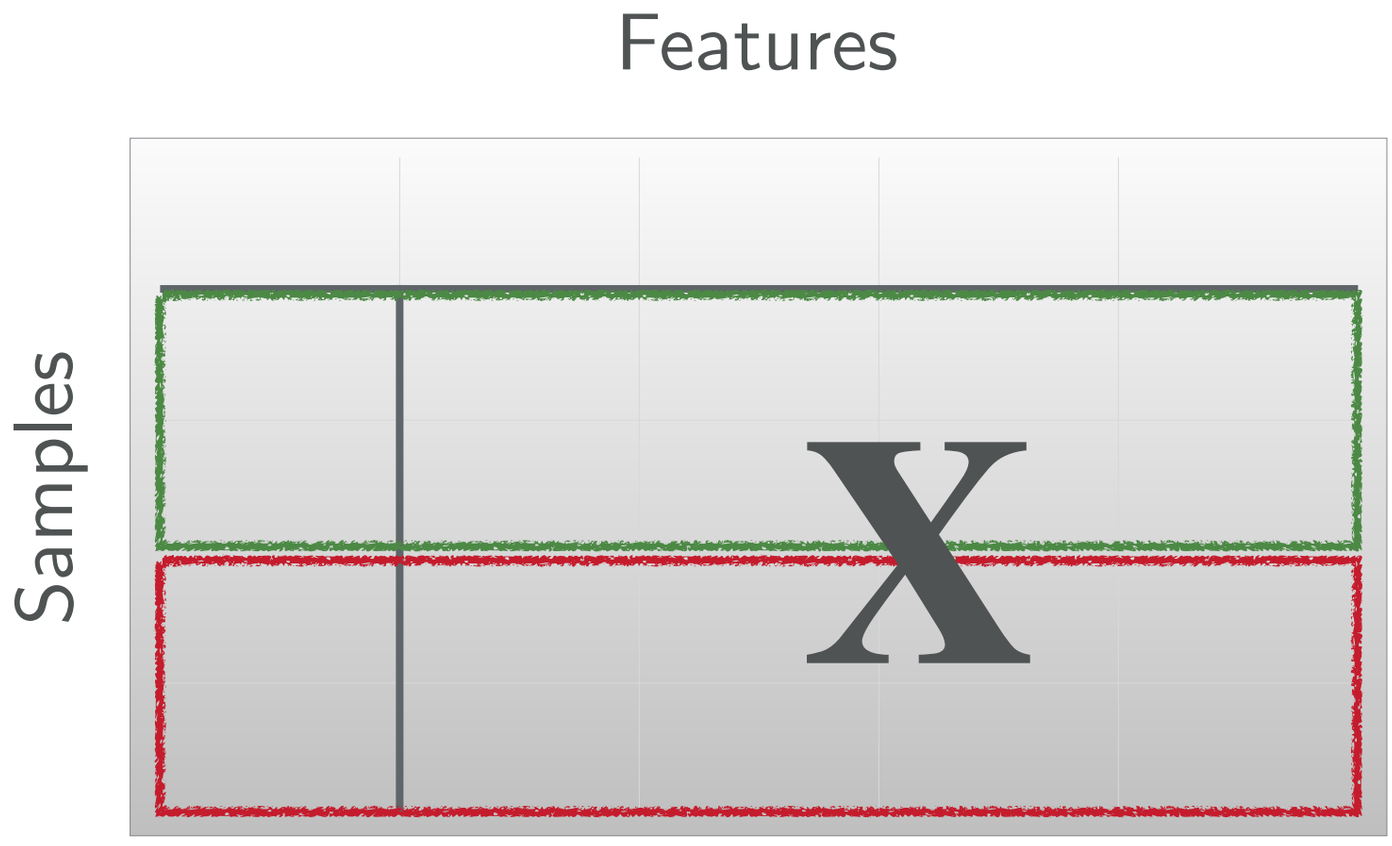
DATASET



The target is made of real numbers:

Supervised Learning, Regression Problem

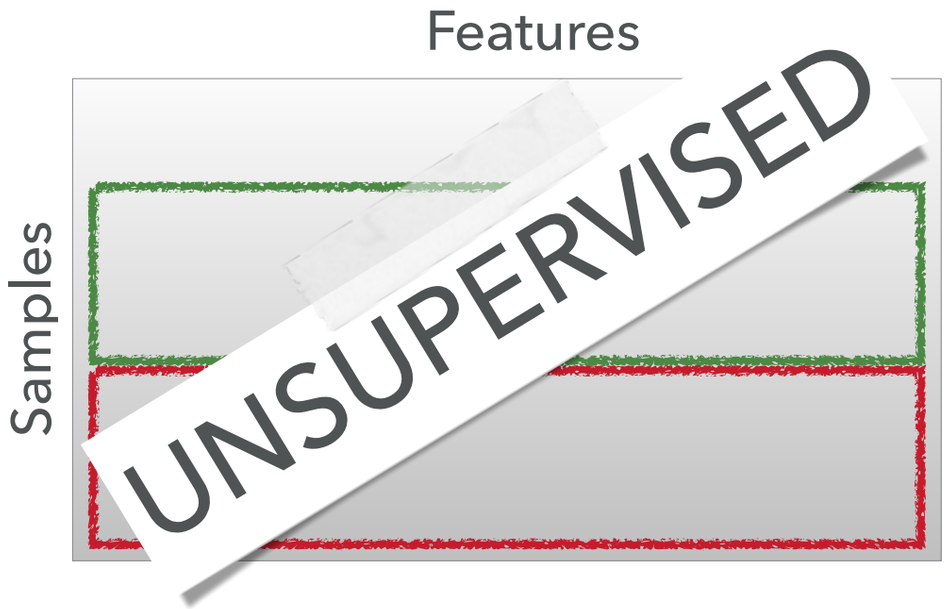
DATASET



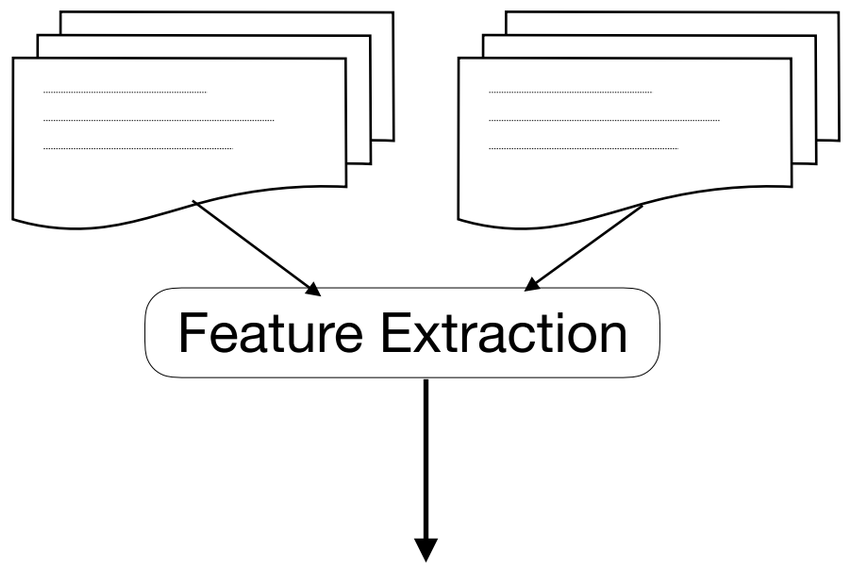
There is no target:

Unsupervised Learning, Clustering Problem

SUPERVISED VS UNSUPERVISED LEARNING



FEATURE ENGINEERING



FEATURE ENGINEERING

$$X$$

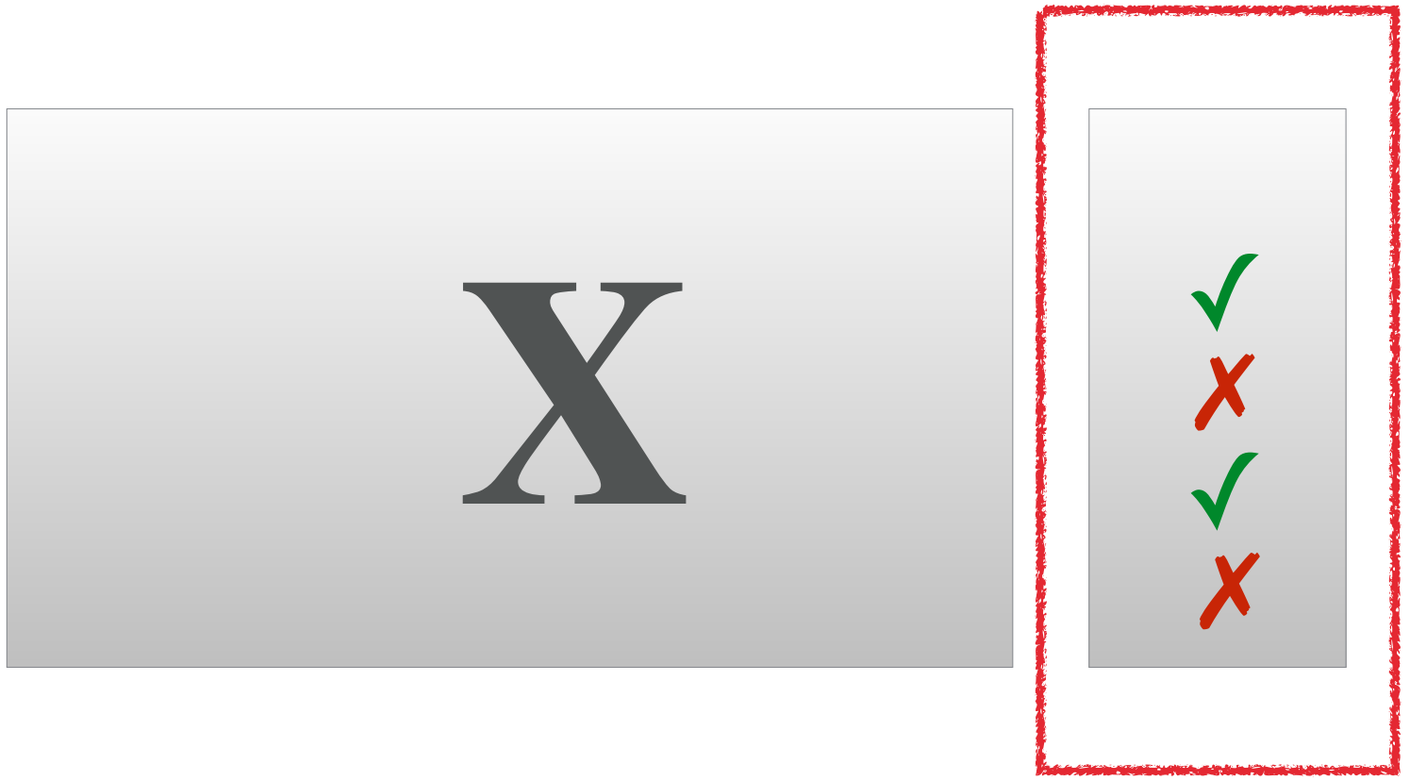
↓

Dimensionality Reduction (Unsupervised Learning)

↓

$$X'$$

DATA QUALITY

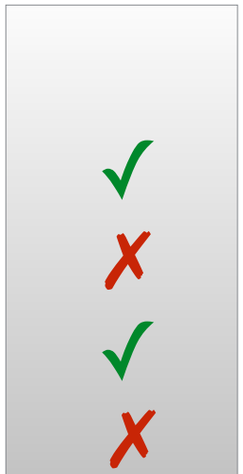


When learning from examples, any algorithm is very sensitive to errors in the target.

Quality of target is essential, otherwise, **garbage in - garbage out**

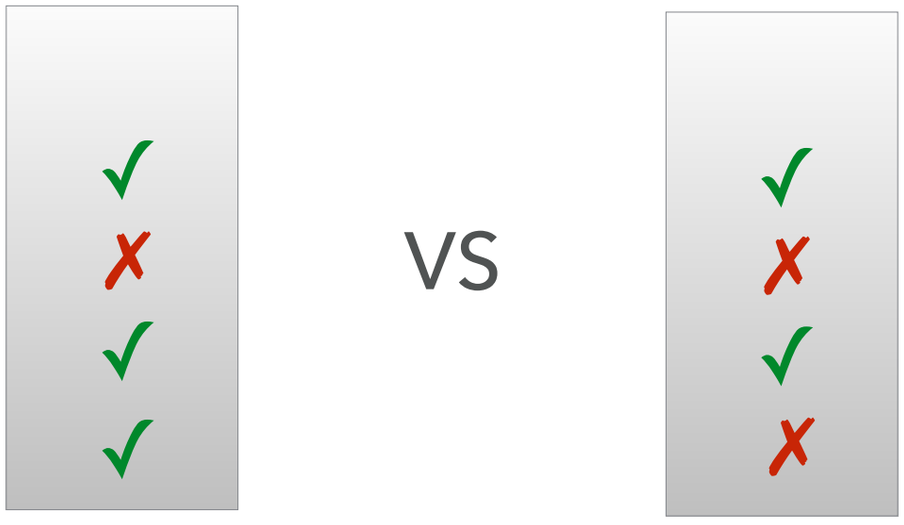
ERRORS AND SCORES

ERROR FUNCTION VS SCORE FUNCTION



ERROR FUNCTION VS SCORE FUNCTION

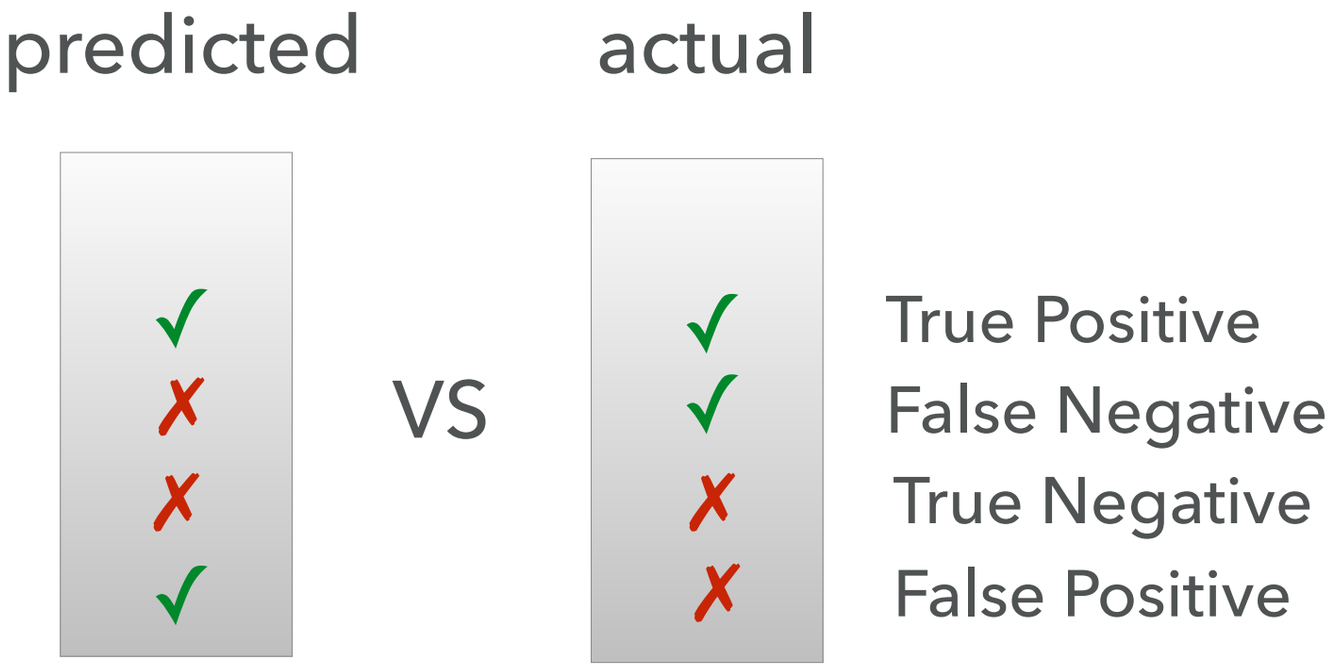
ERROR FUNCTION VS SCORE FUNCTION



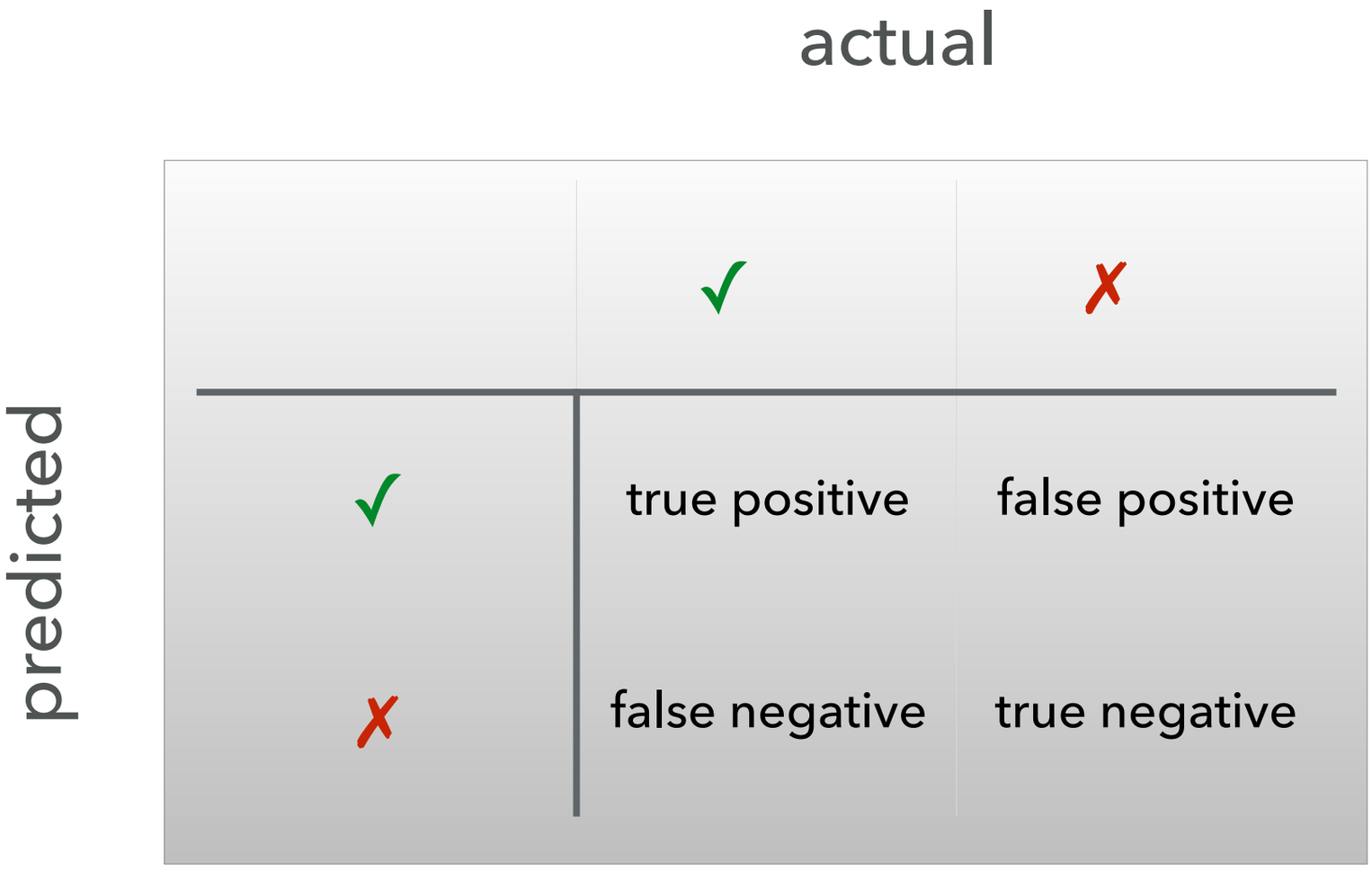
Mean Absolute Error: 0.25

Accuracy Score: 0.75

SCORE FUNCTIONS (CLASSIFICATION)



CONFUSION MATRIX



ACCURACY

actual

predicted	✓	✗
✓	TP	FP
✗	FN	TN

$$\frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Warning: accuracy is sensitive to highly unbalanced classes

PRECISION

actual

predicted	✓	✗
✓	TP	FP
✗	FN	TN

$$TP \;/\; (TP + FP)$$

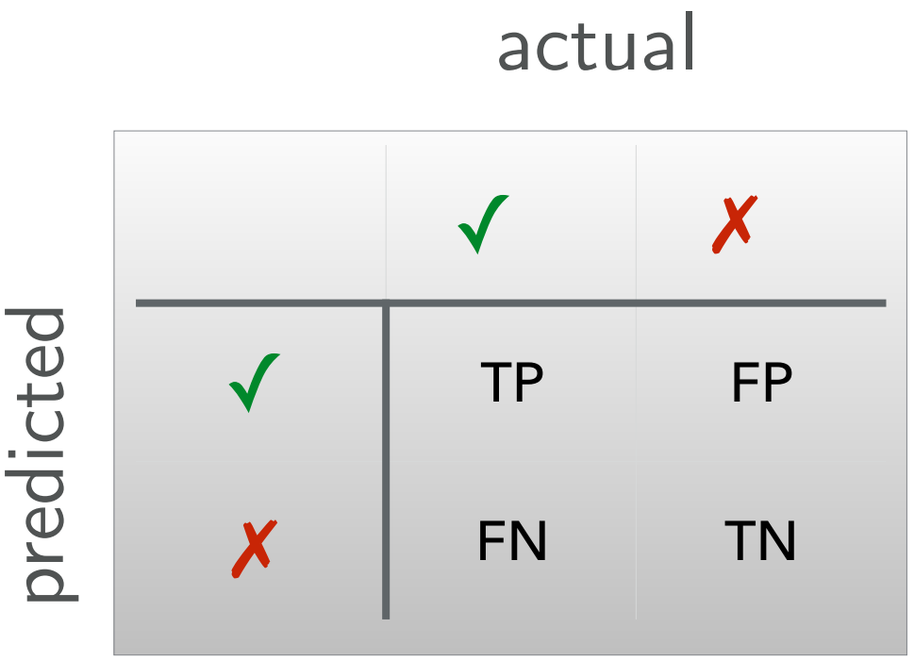
RECALL

actual

predicted \ actual	✓	✗
✓	TP	FP
✗	FN	TN

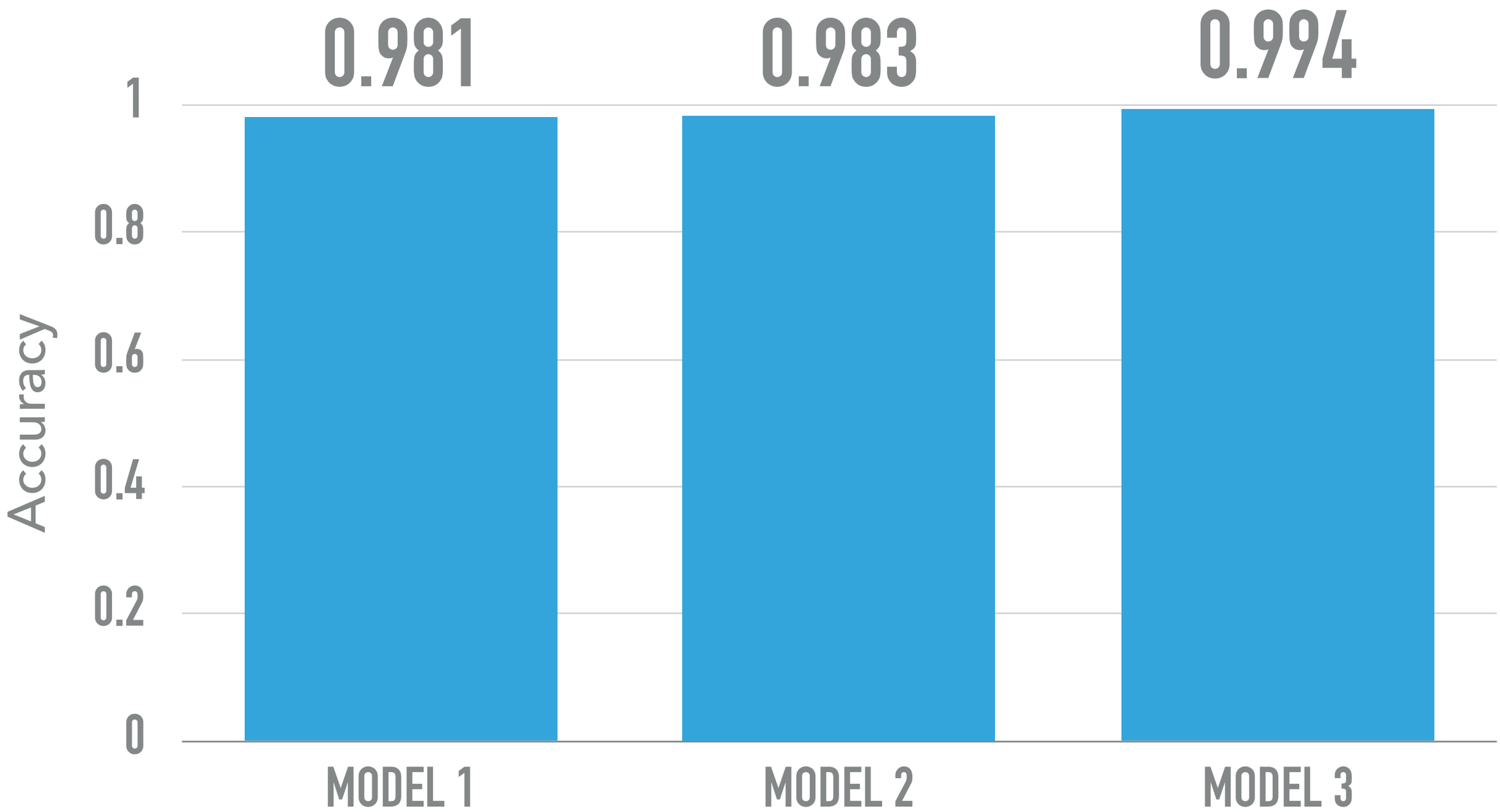
$$TP \ / \ (TP + FN)$$

F1 SCORE

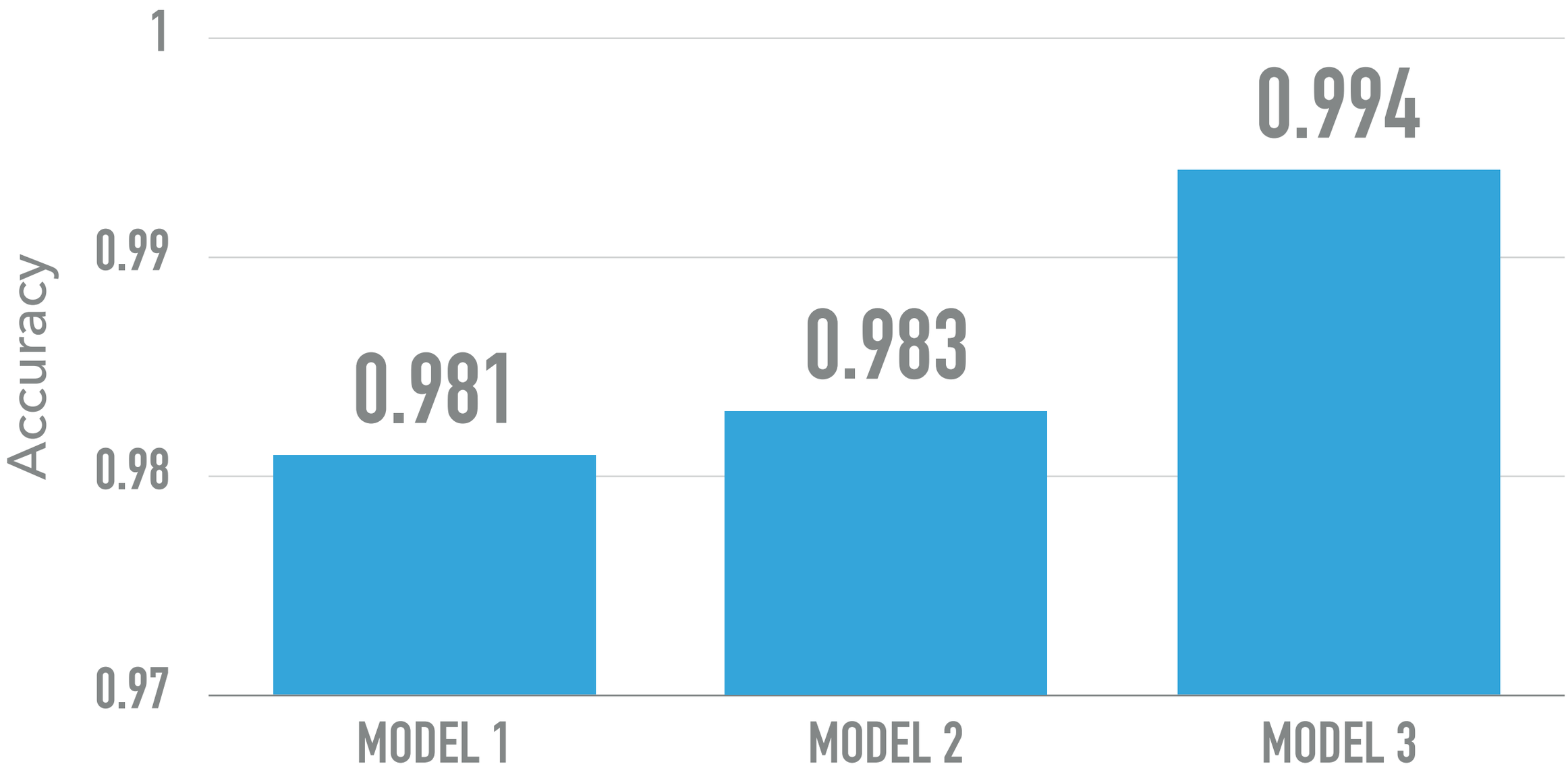


$$\frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

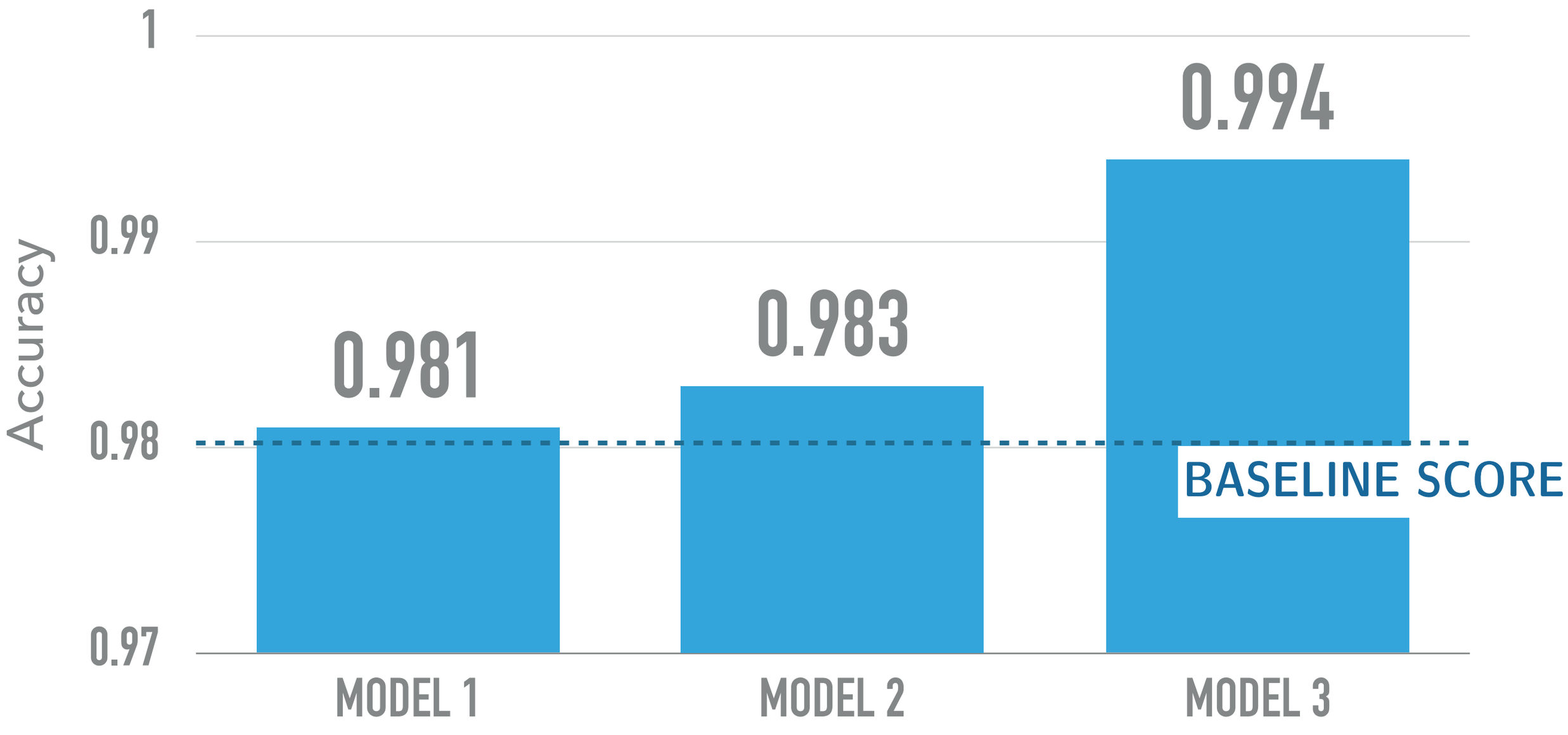
FRAUD DETECTION SCORES



FRAUD DETECTION SCORES

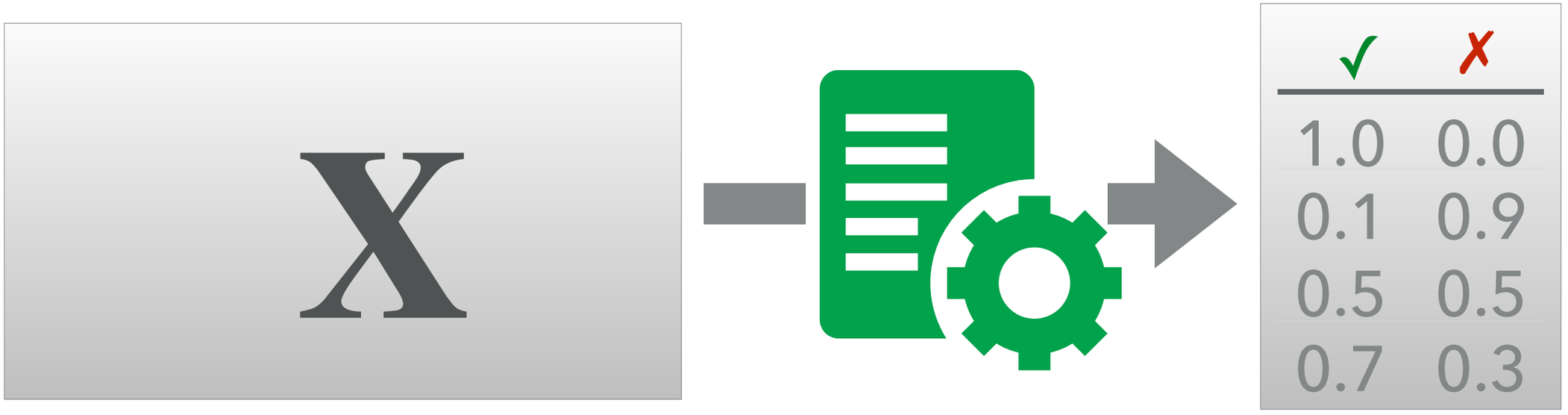


FRAUD DETECTION SCORES



Class unbalance: only 2% of target is positive

CLASS PROBABILITY



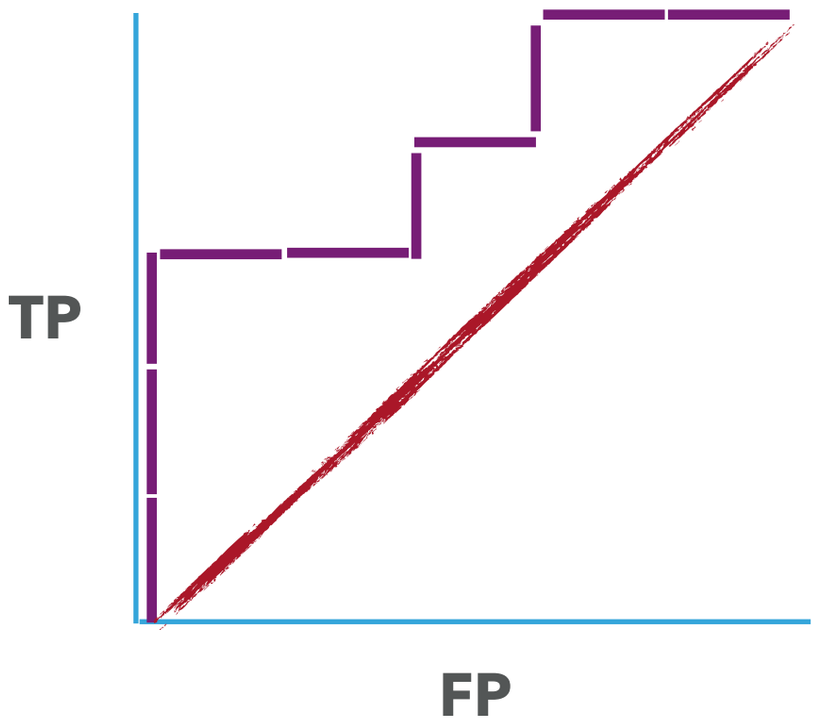
Many classification algorithms give not only the predicted class, but also the estimated probability of each class

ROC AUC

RECEIVING OPERATOR CHARACTERISTICS - AREA UNDER THE CURVE

Y	0	0	1	0	1	1	1	1	1	0
P	**0.2**	**0.1**	**0.4**	**0.5**	**0.9**	**0.3**	**0.7**	**0.8**	**0.5**	**0.4**

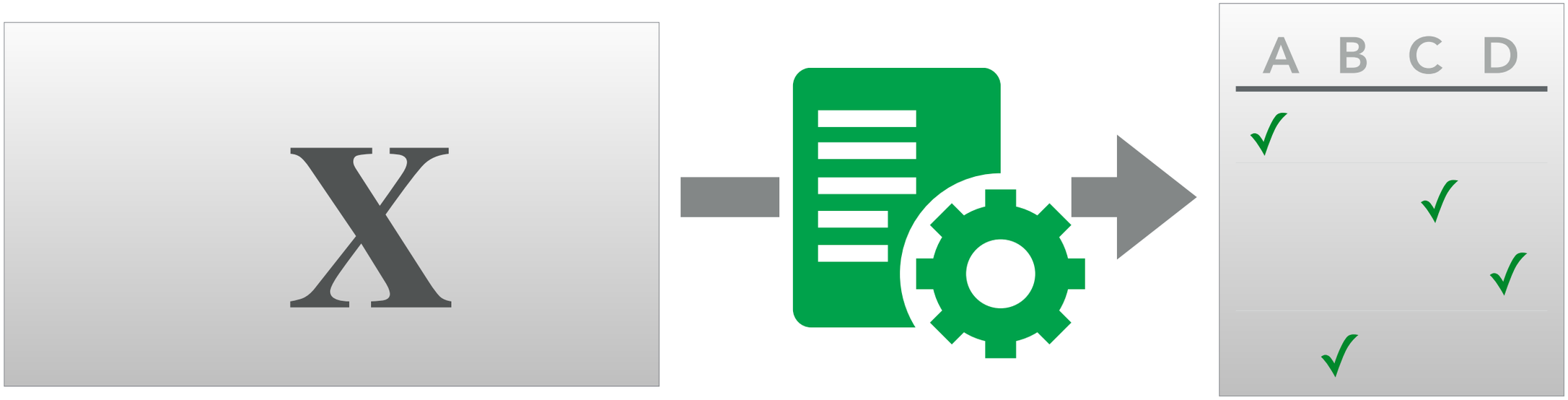
Y	1	1	1	0	1	1	0	1	0	0
P	**0.9**	**0.8**	**0.7**	**0.5**	**0.5**	**0.4**	**0.4**	**0.3**	**0.2**	**0.1**



In a graph with True positive rate vs False positive rate, calculate the area under the curve

Random predictions give always ROC AUC = 0.5, independent of class proportions

MULTICLASS CLASSIFICATION

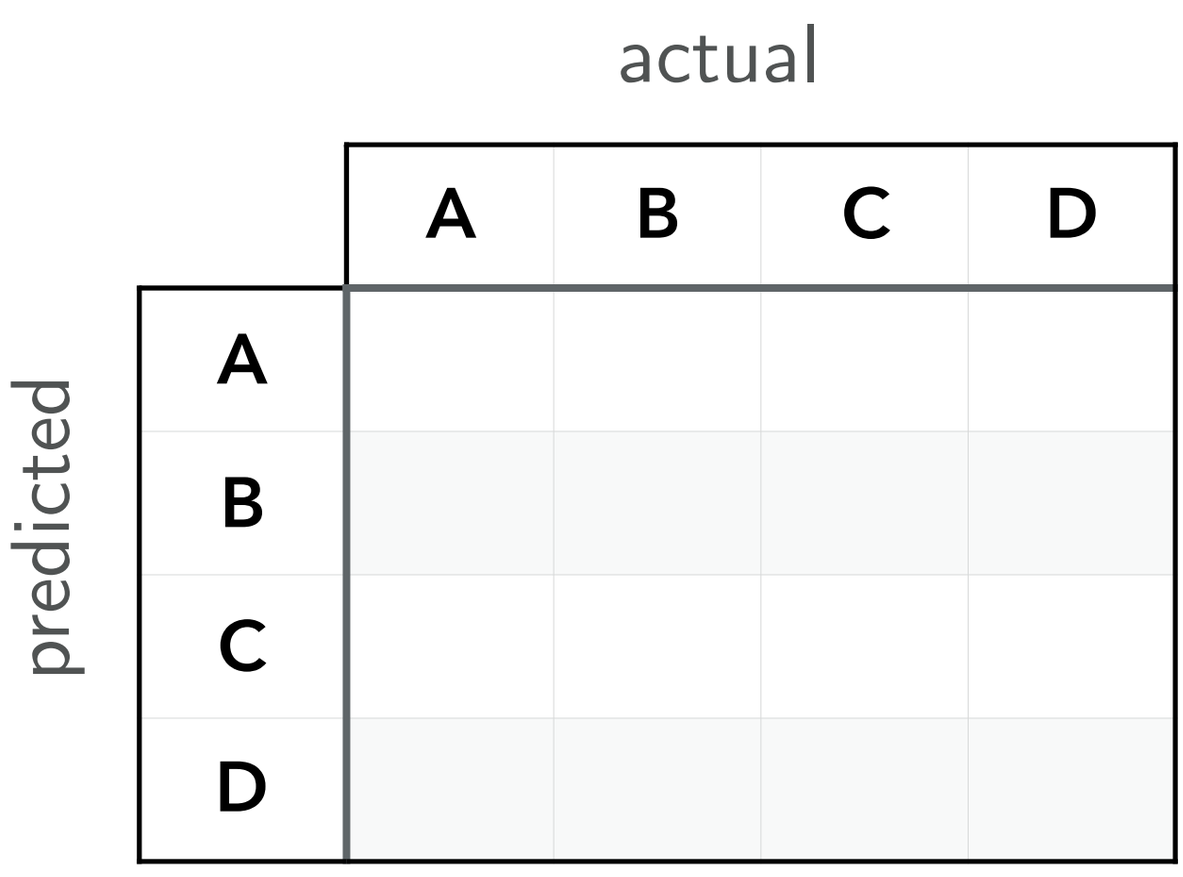


There are different strategies for using binary classifiers in multiclass classification problems:

- One-versus-all
- One-versus-one

CONFUSION MATRIX - MULTICLASS

In the case of multiclass classification, the diagonal represent the right predictions, and the off-diagonal the errors



ERROR FUNCTIONS / SCORE FUNCTIONS (REGRESSION)

Measure the difference between target value and predicted value

MAE

Mean Absolute Error

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

MSE

Mean Squared Error

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

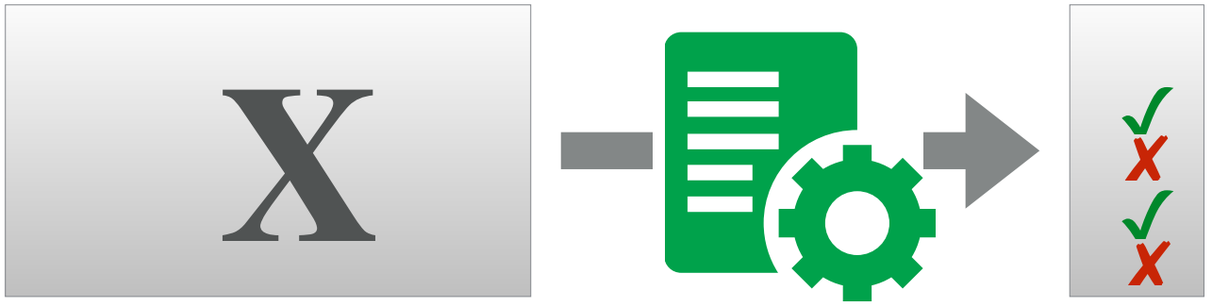
R^2 Score

Coefficient of Determination

$$\frac{\sum(\hat{y}_i - \bar{y})^2}{\sum(y_i - \bar{y})^2} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

MODEL SELECTION

LEARNING MODEL

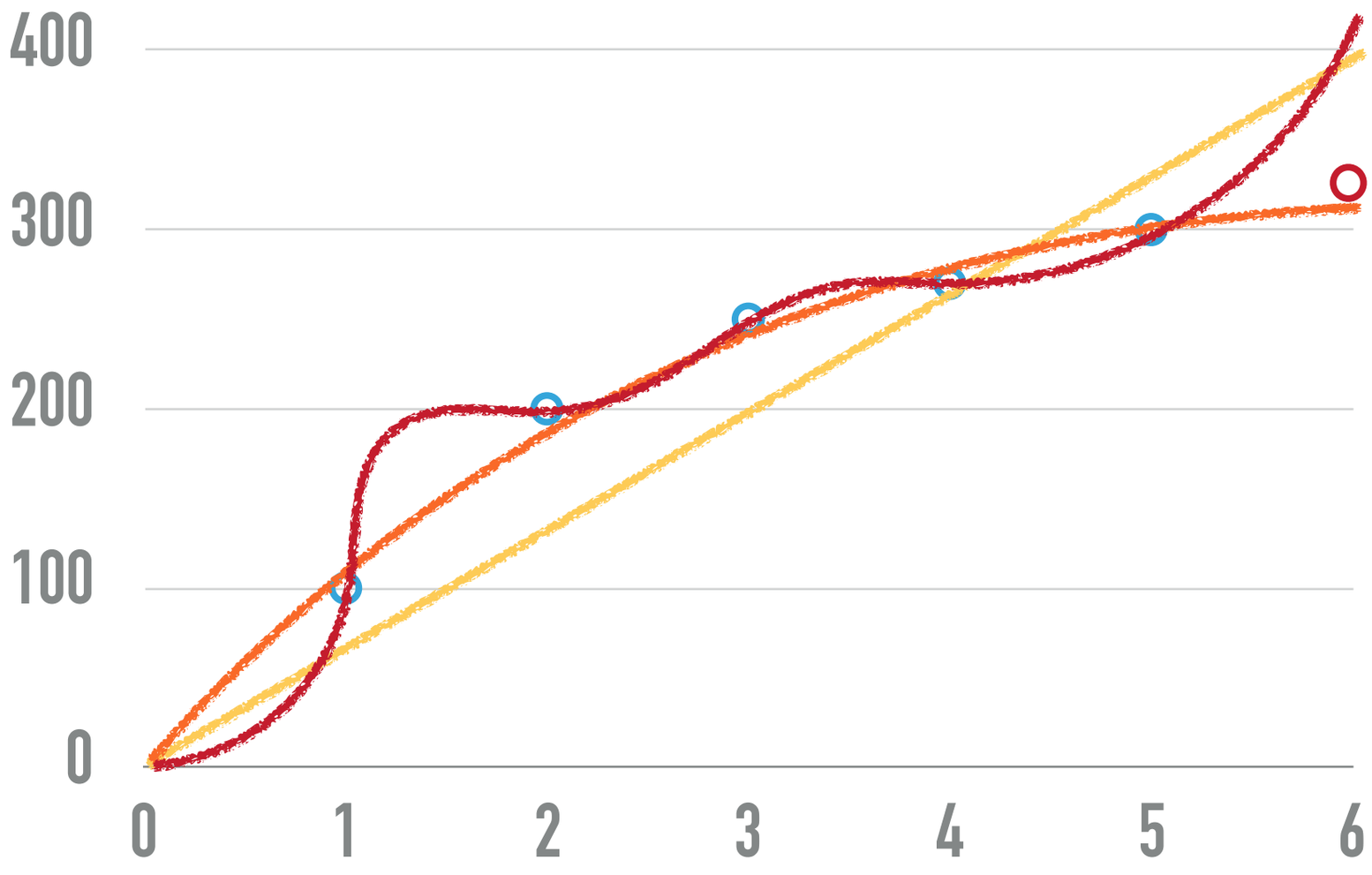


The target function $f: \mathcal{X} \to \mathcal{Y}$ that maps the data to the target is unknown, and might be very noisy or very complex

The hypothesis function $g: \mathcal{X} \to \mathcal{Y}$ is one of the functions in the hypothesis set $\mathcal{H}$ that minimizes a chosen loss function

Fitting data:

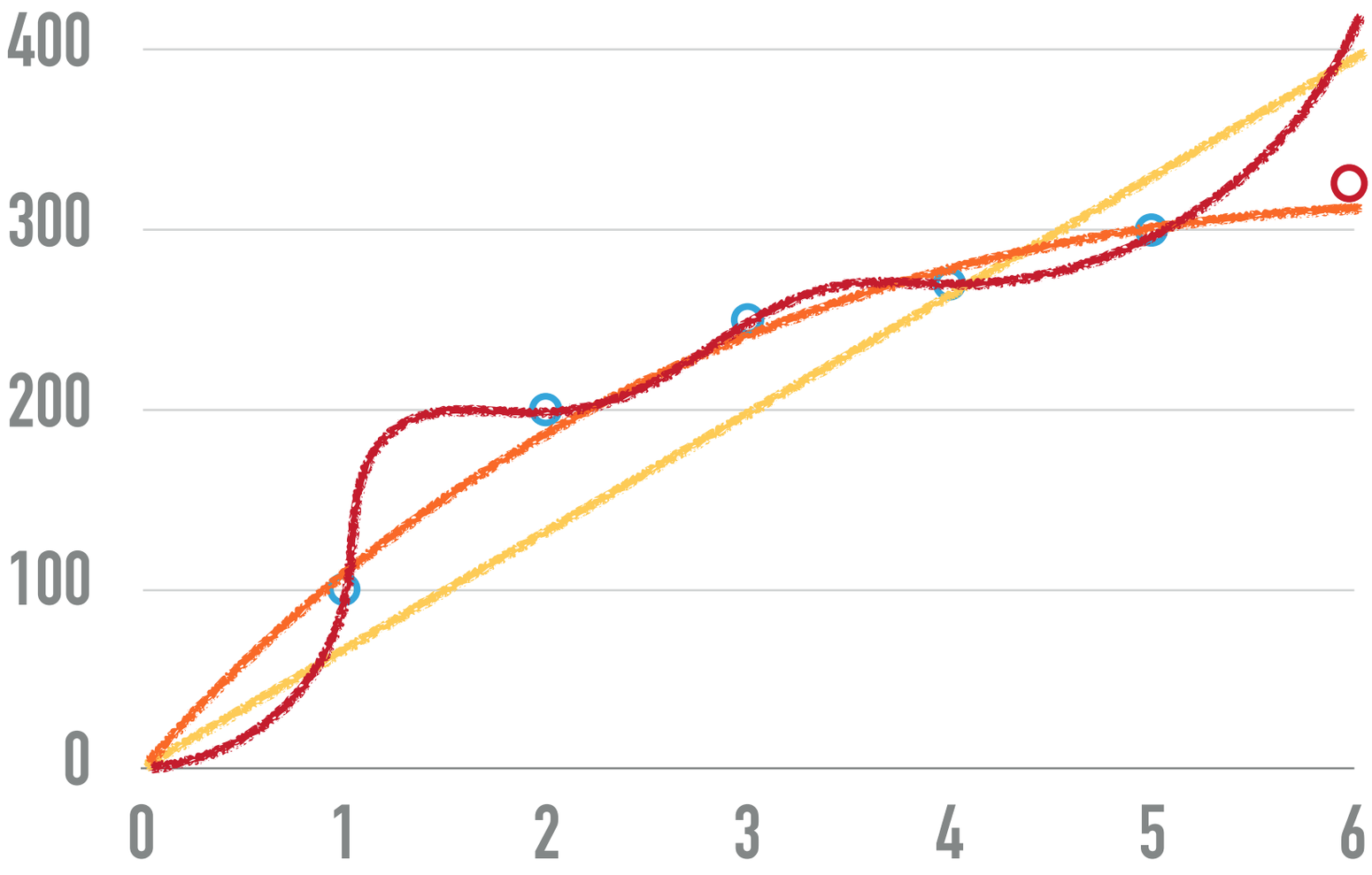
find the model that results in the lowest error with respect to the data



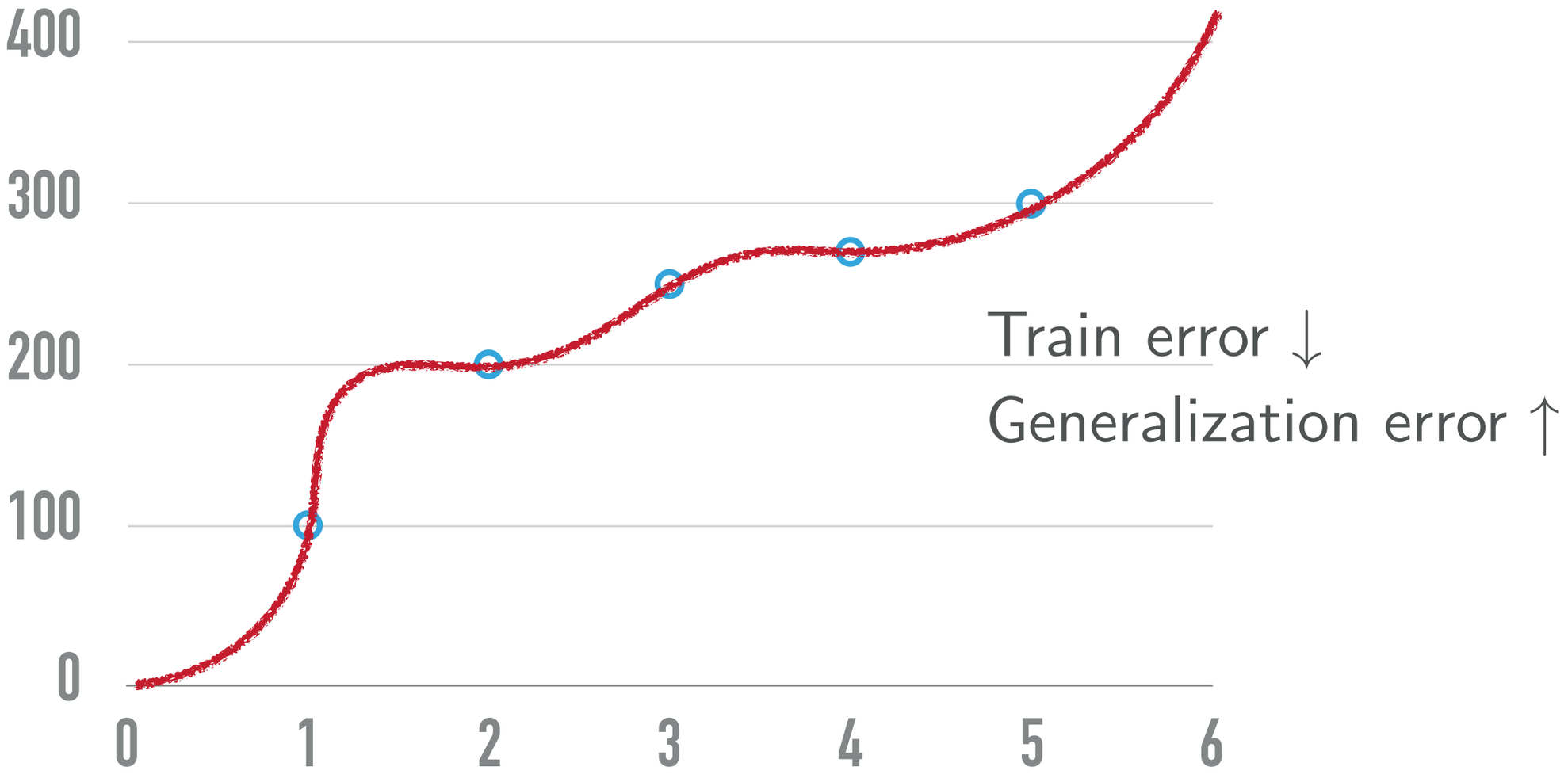
If we have too many parameters, the model may fit the data very well, but fail to generalize to new examples (overfitting)

Learning from data:

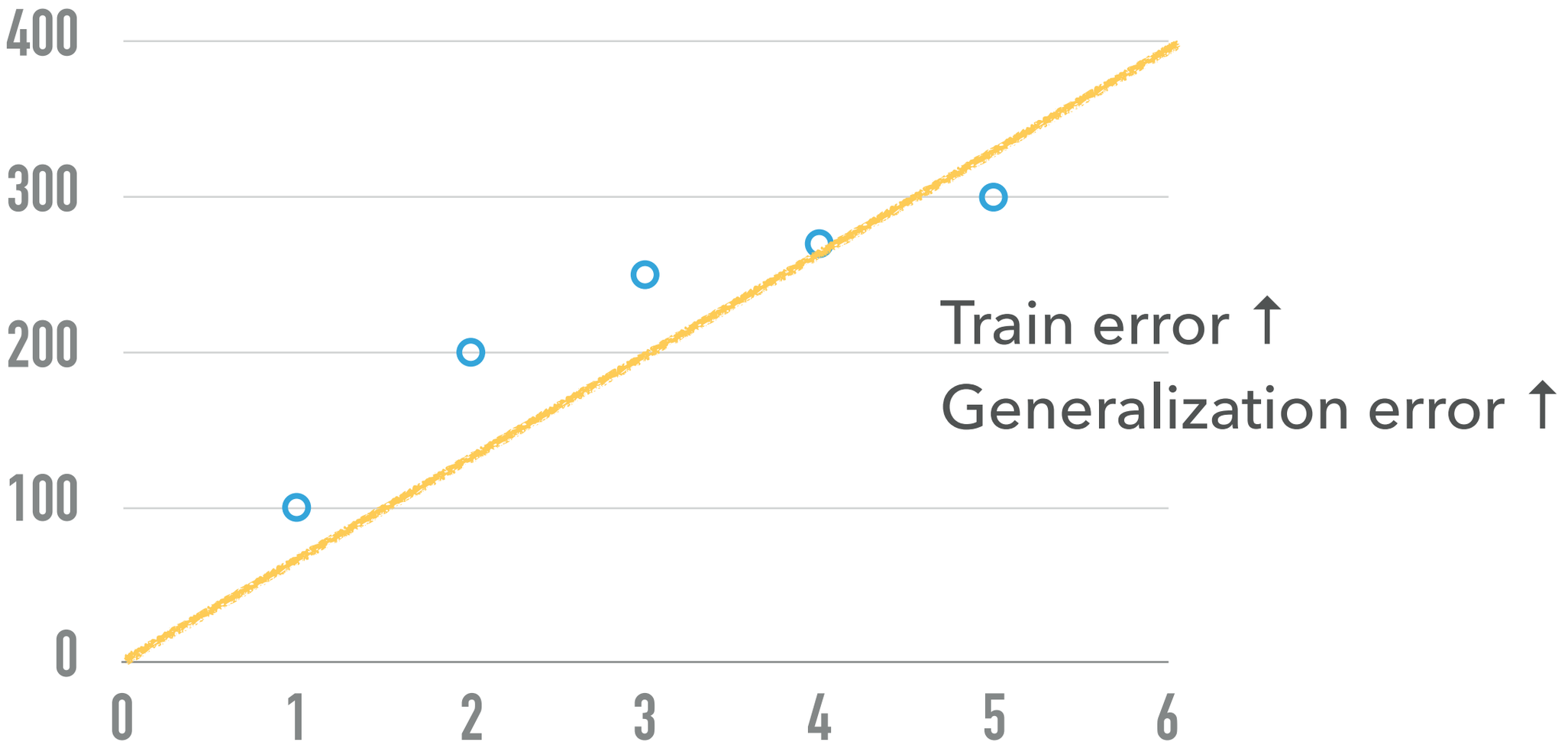
find the model that results in the lowest error with respect to unseen data



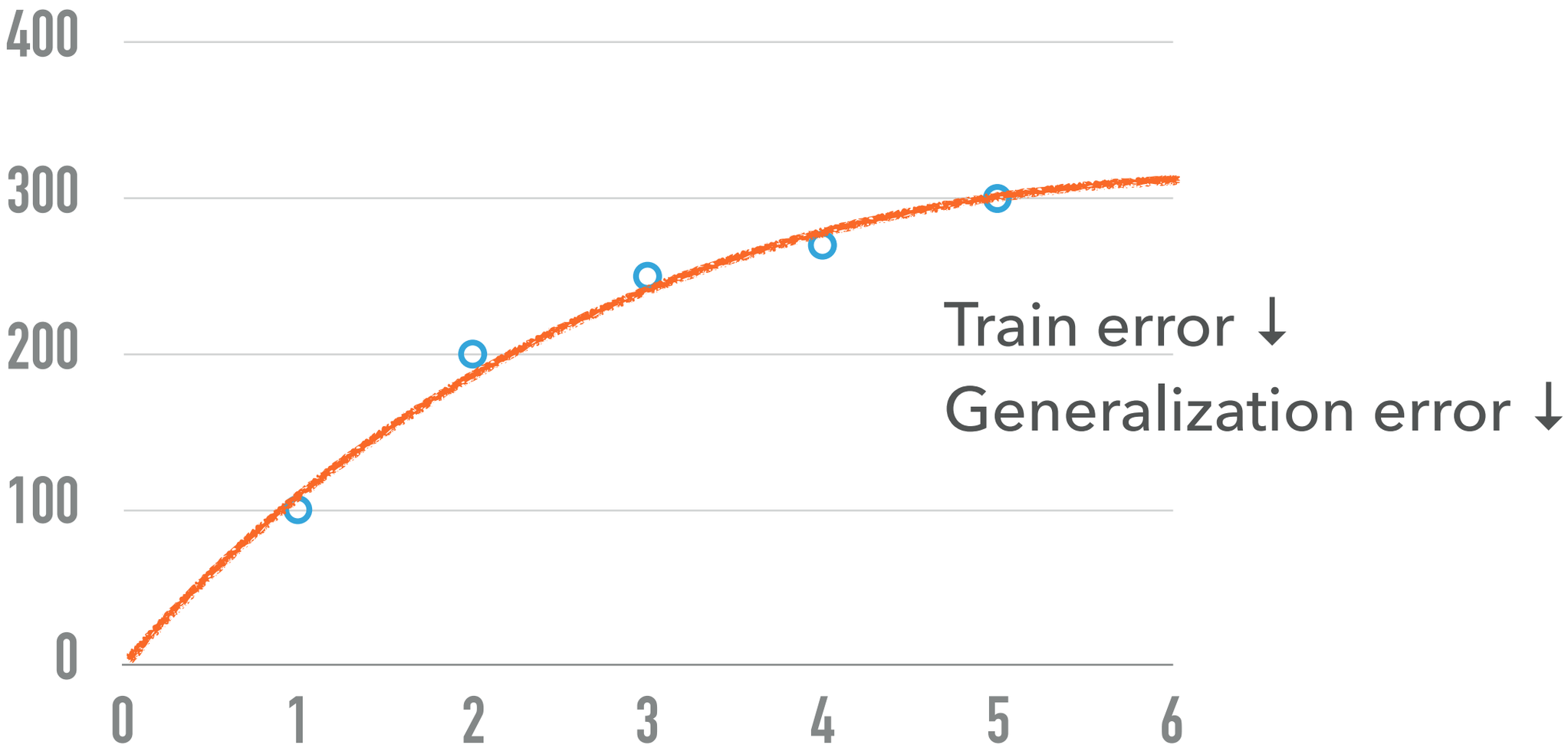
OVERFITTING



UNDERFITTING



BEST MODEL



LEARNING THEORY

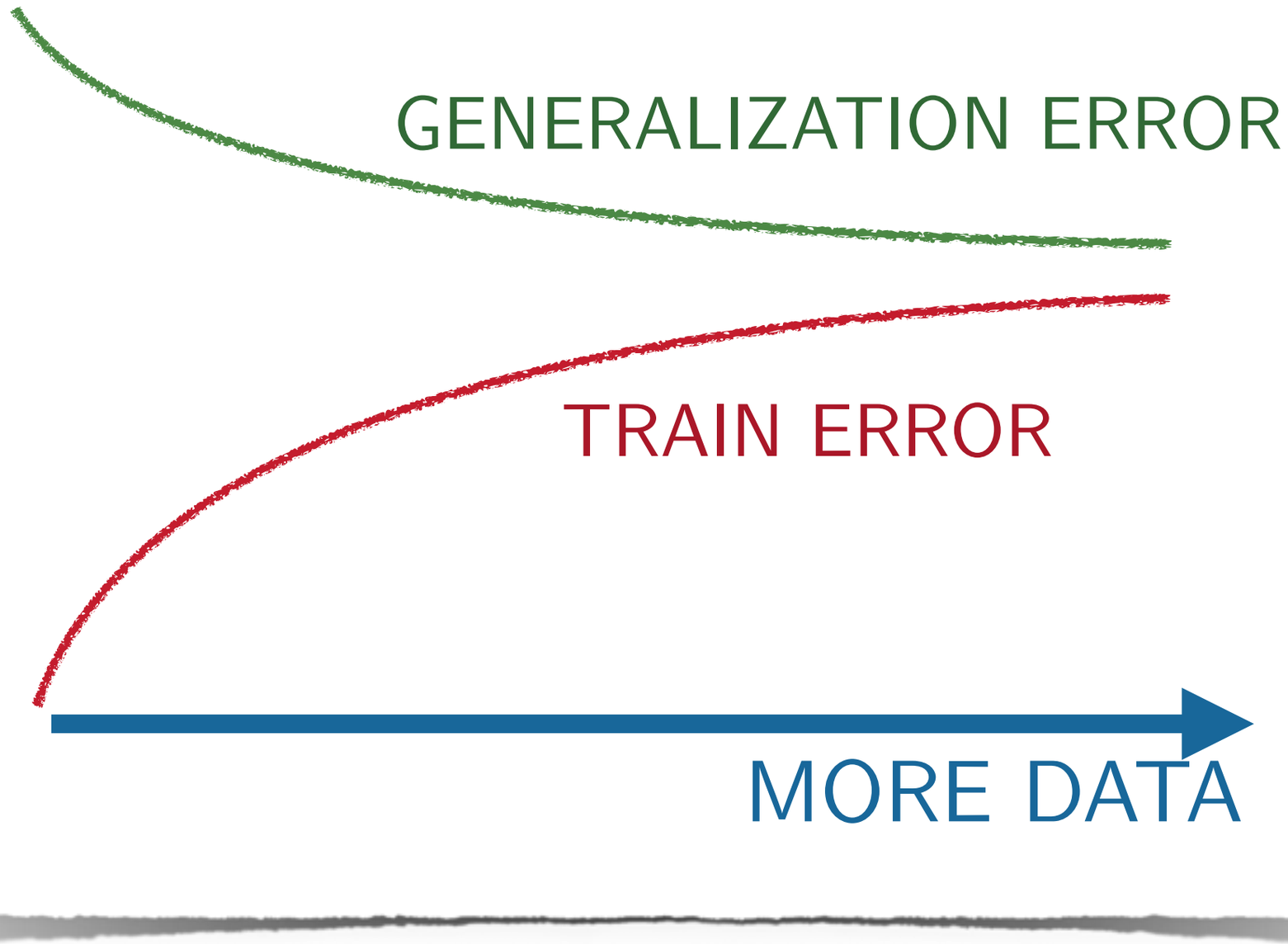
- Do I have enough data for adequate learning?
- Is the model complexity adequate for the problem?
- What is the best strategy to reduce error/ increase performance?

How can my model generalize better?

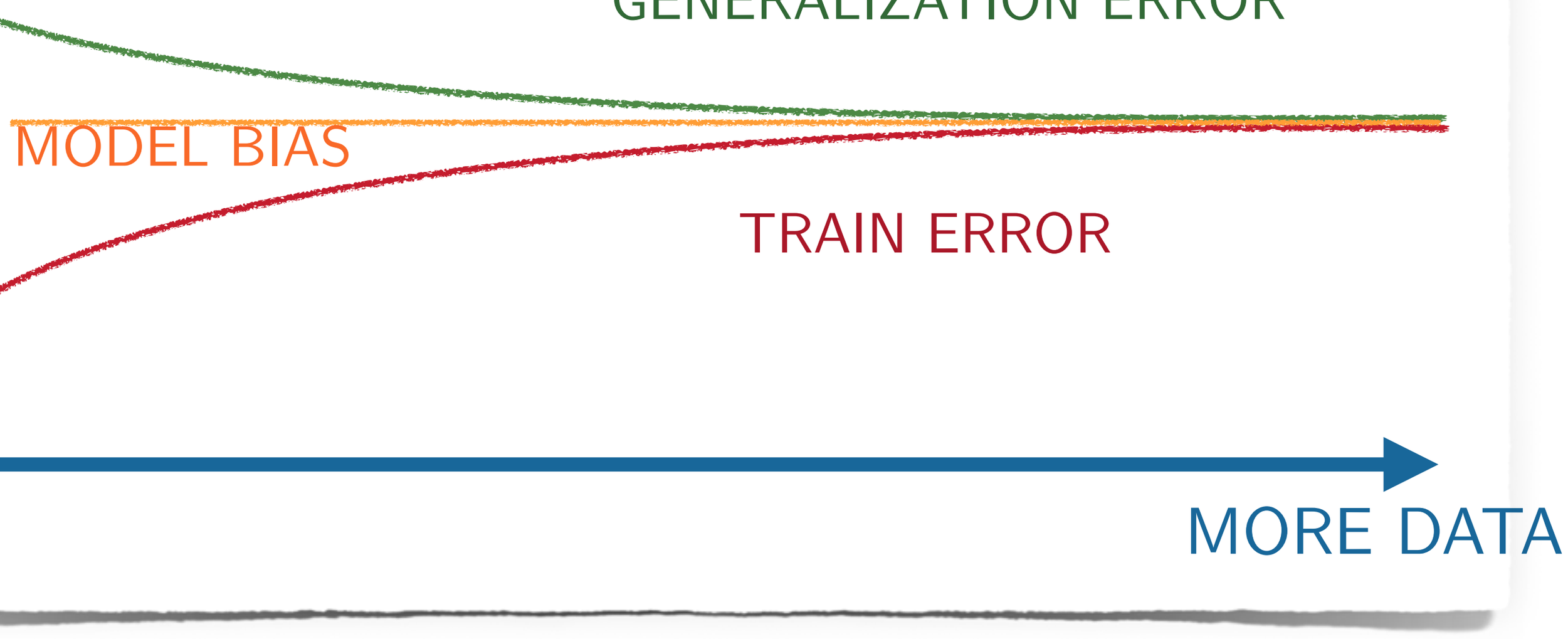
- Have a more complex model?
- Collect more samples?
- Have more dataset features?

SUPPORT FOR STRATEGIC DECISIONS

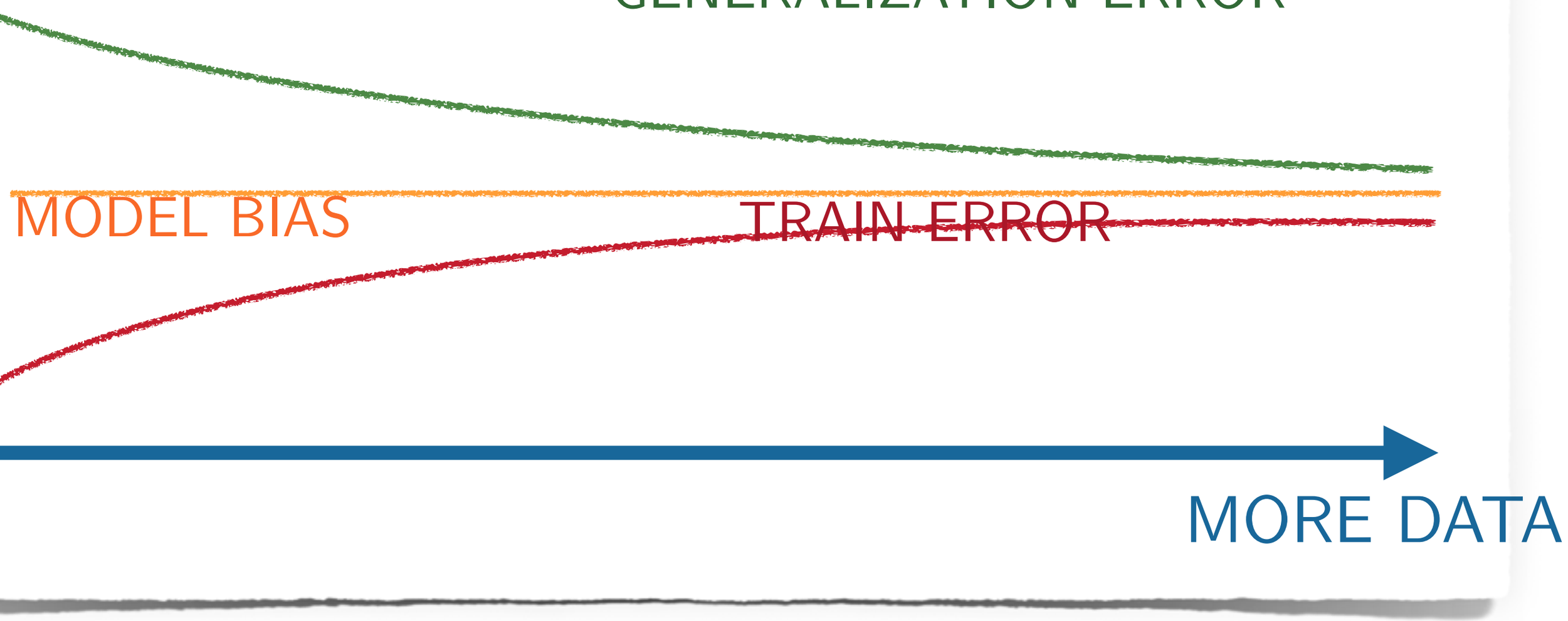
LEARNING CURVE



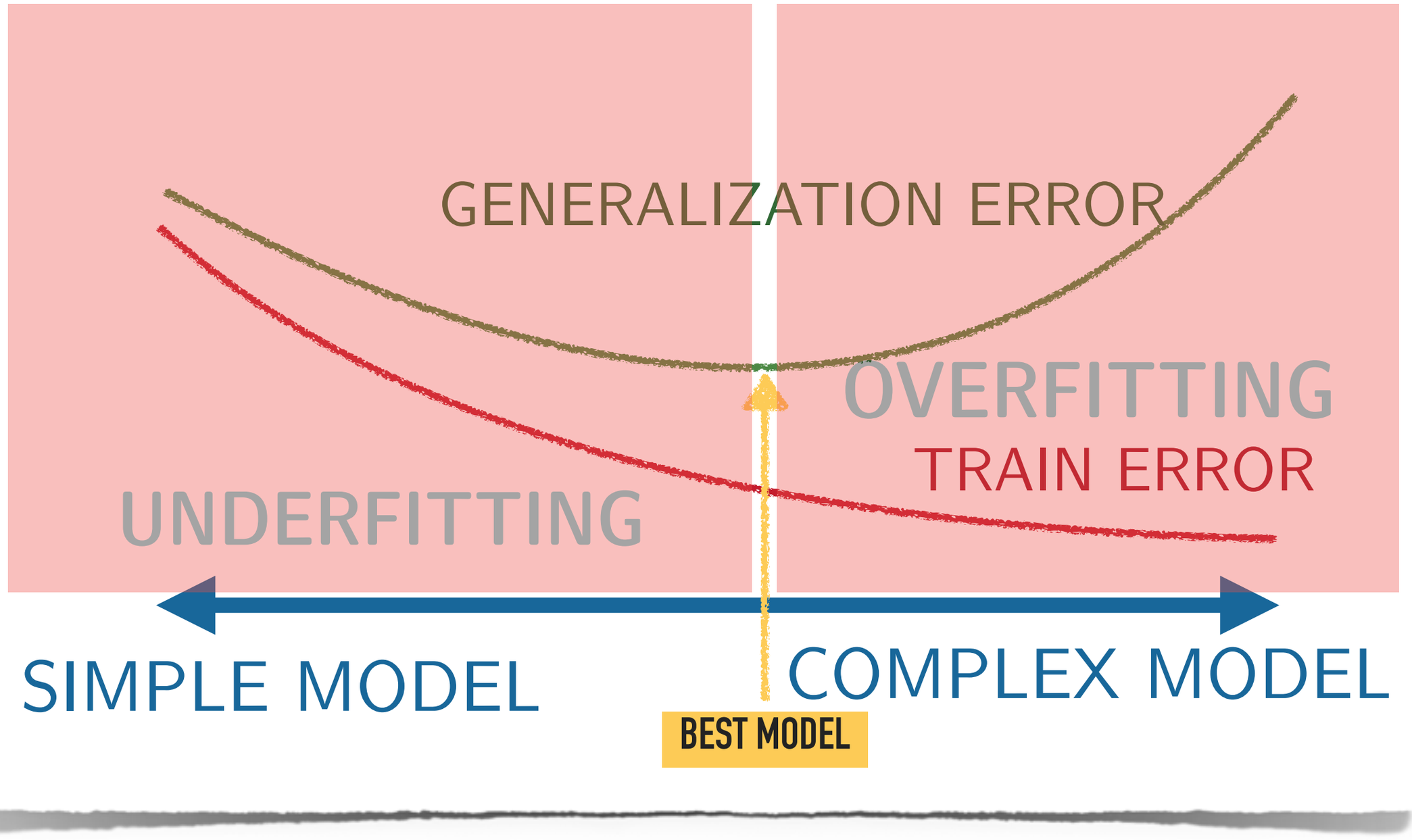
LEARNING CURVE



LEARNING CURVE



LEARNING CURVE



MODEL COMPLEXITY

The model complexity is controlled by changing the **hyperparameters** (e.g. number of degrees in a polynomial regression)

The hyperparameters control the effective number of parameters (degrees of freedom, or VC dimension) of the model

REGULARIZATION

The VC dimension of the model can also be controlled by imposing limits to the choice of parameters

Example:
Linear Regression with no regularization

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

Linear Regression with L1 or L2 regularization

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{x}_i\mathbf{w})^2 + \alpha\sum_{i=1}^{D}|w_i| + \beta\sum_{i=1}^{D}w_i^2$$

MODEL SELECTION - VALIDATION

In practice, the generalization (out of sample) error is estimated via **validation**: we keep part of the training set as validation set, don't use it for fitting the model and use it only for scoring the model.

Once we find the optimal set of hyperparameters, we proceed to fitting the model with the whole training set.

Tradeoff problem: how much of the training set we should keep for validation?

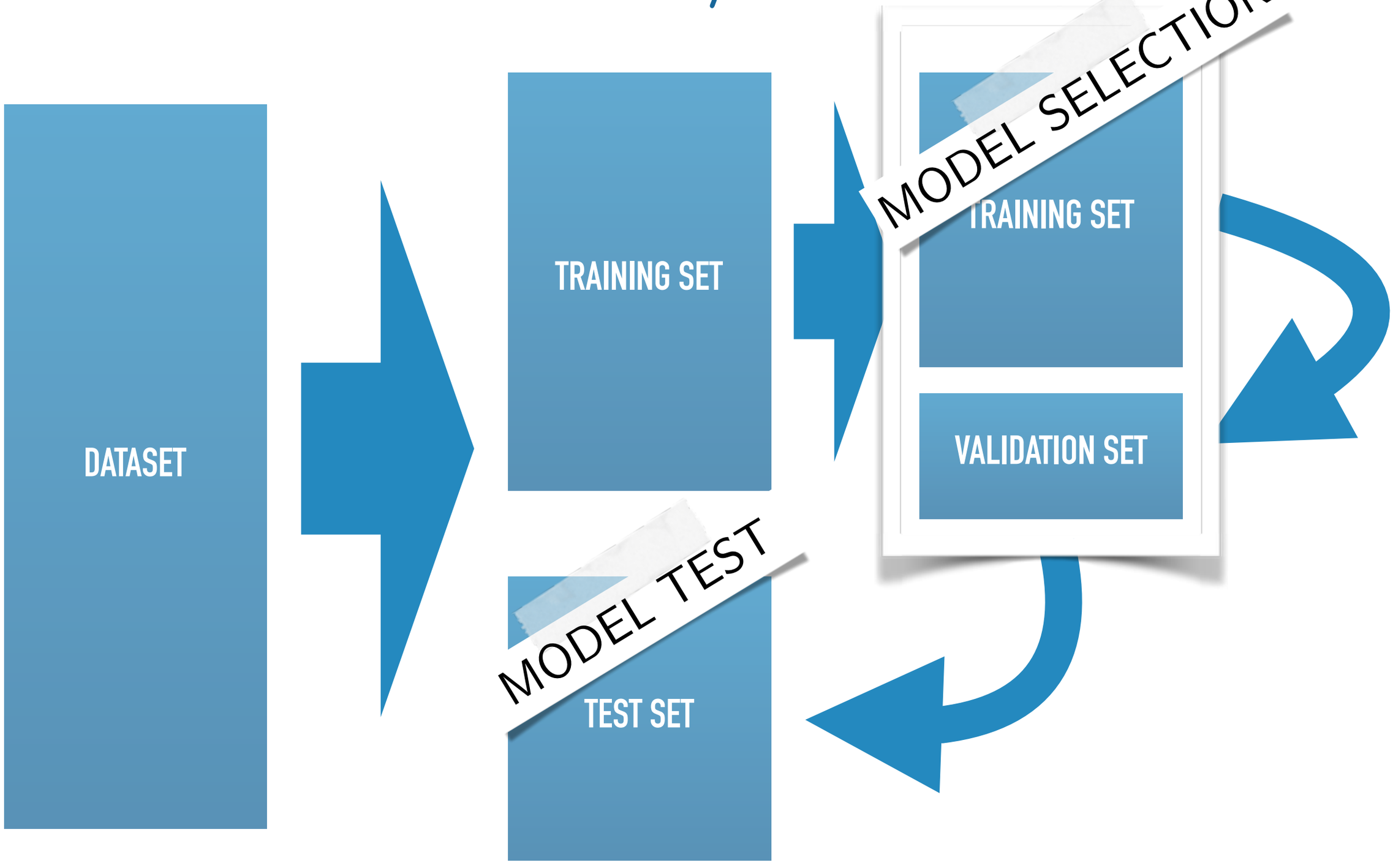
MODEL SELECTION - VALIDATION

Train-test split: part of the training data is used as validation set

K-fold cross-validation: we split the training data into K folds, and we iterate K times by using the k-th fold as validation set and the other K-1 folds to fit the model

Leave-one-out cross-validation: we iterate N times: we keep a single sample for validation, and use N-1 samples to fit the model

MODEL VALIDATION / MODEL TESTING

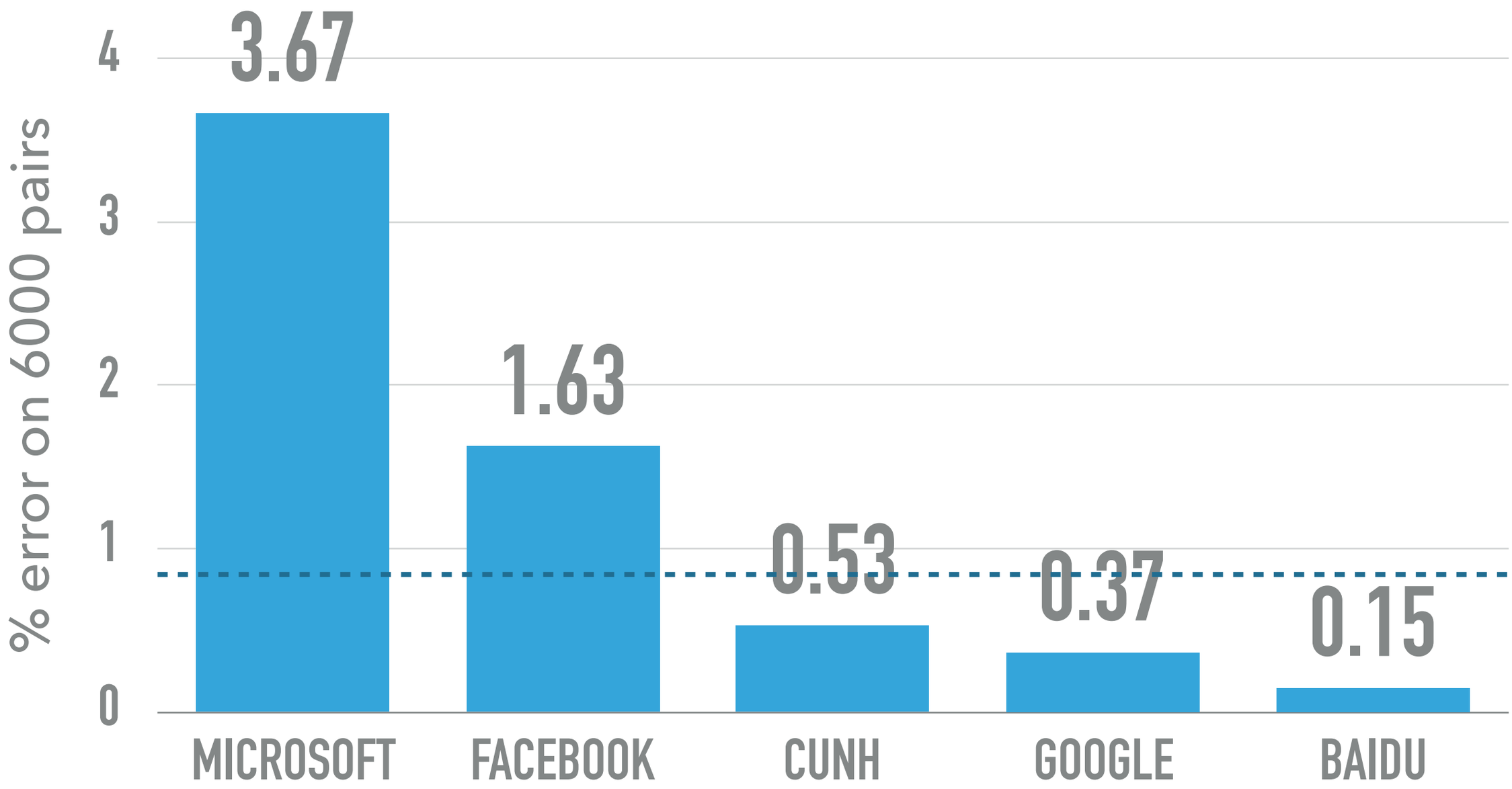


LEARNING EXAMPLE: FACES RECOGNITION

LABELED FACES IN THE WILD

FOR EACH PAIR, DECIDE IF THE IMAGES ARE OF THE SAME PERSON

FACE RECOGNITION ERRORS



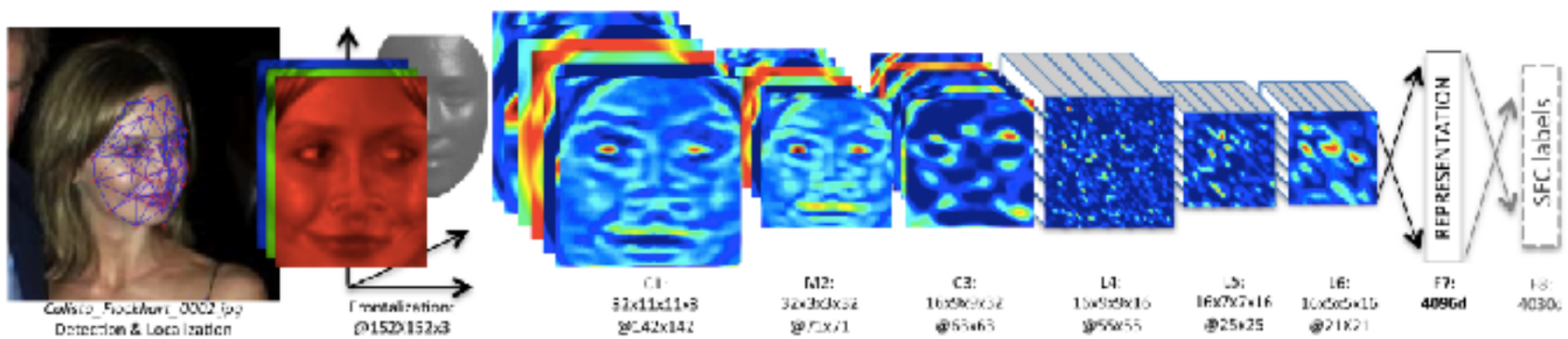
SUPERVISED LEARNING MODELS

- Classification:
  - Naive Bayes
  - Logistic Regression
  - Support Vector Machines
  - Decision Tree
  - Nearest Neighbors
  - Random Forest
  - Gradient Boosting
  - Neural Networks
  - ...
- Regression
  - Linear Regression
  - LASSO (Linear Regression with Regularization)
  - ...

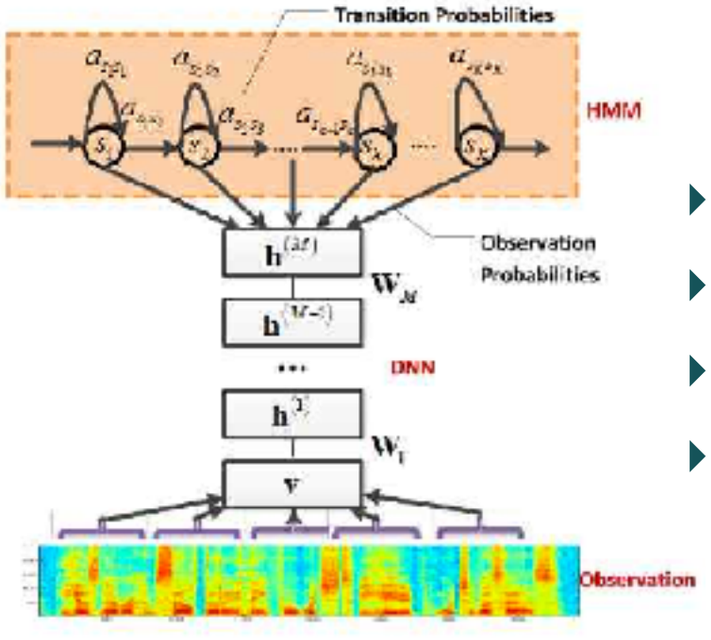
DEEP NEURAL NETWORKS / DEEP LEARNING

image annotation

face recognition

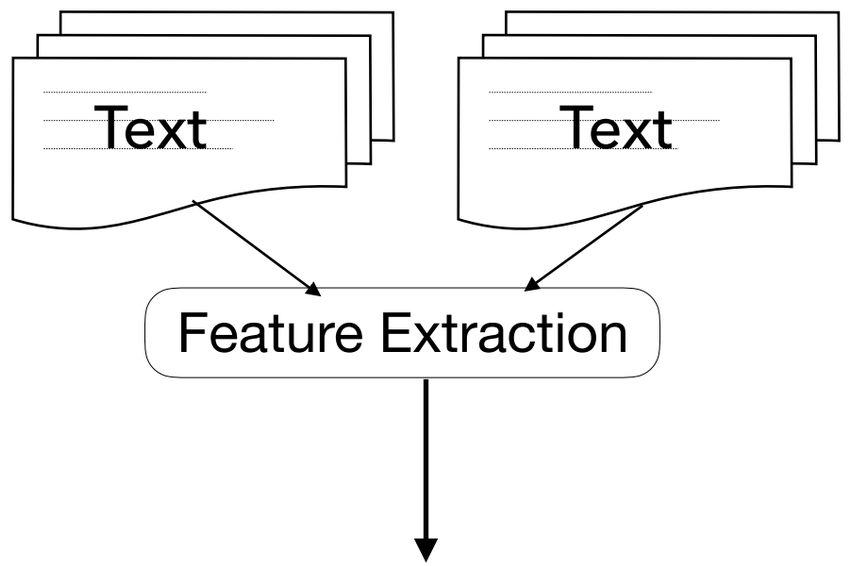


speech recognition

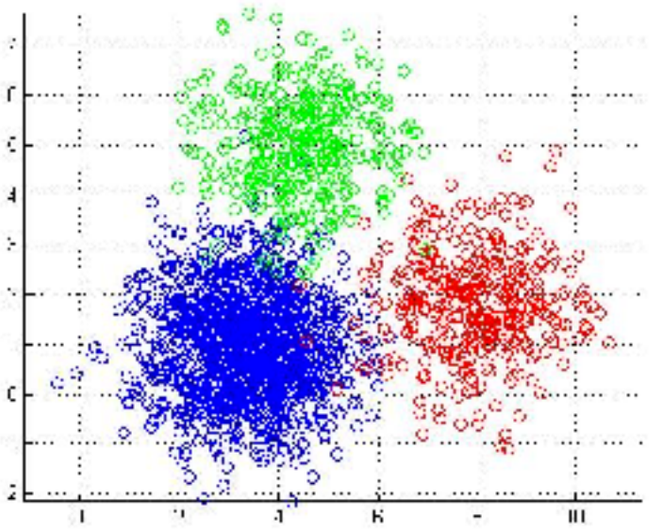


- Google Speech API
- Facebook's wit.ai
- Microsoft's Bing Speech
- Apple Dictation

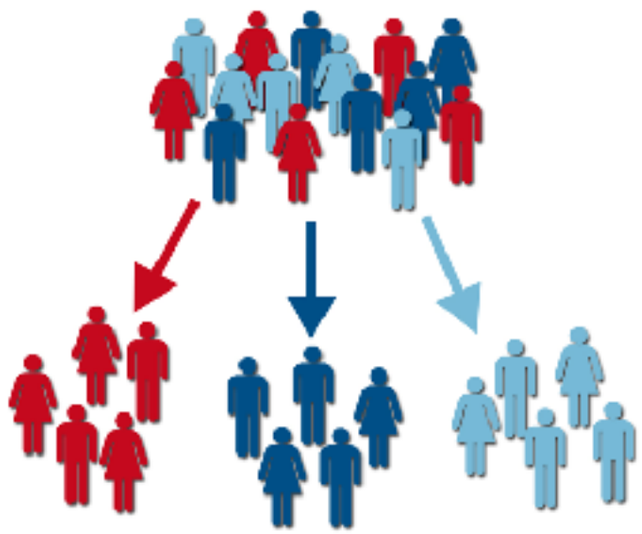
TEXT ANALYSIS



UNSUPERVISED LEARNING

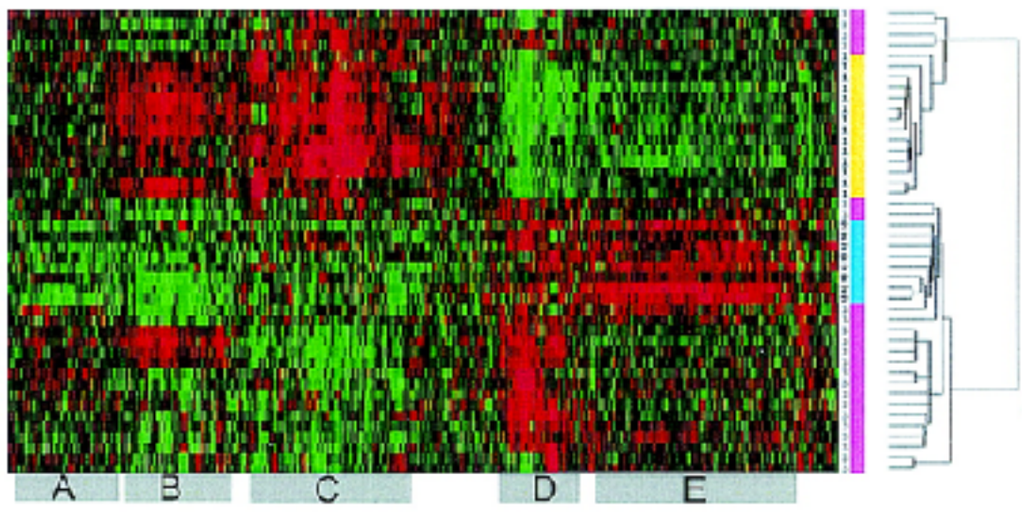


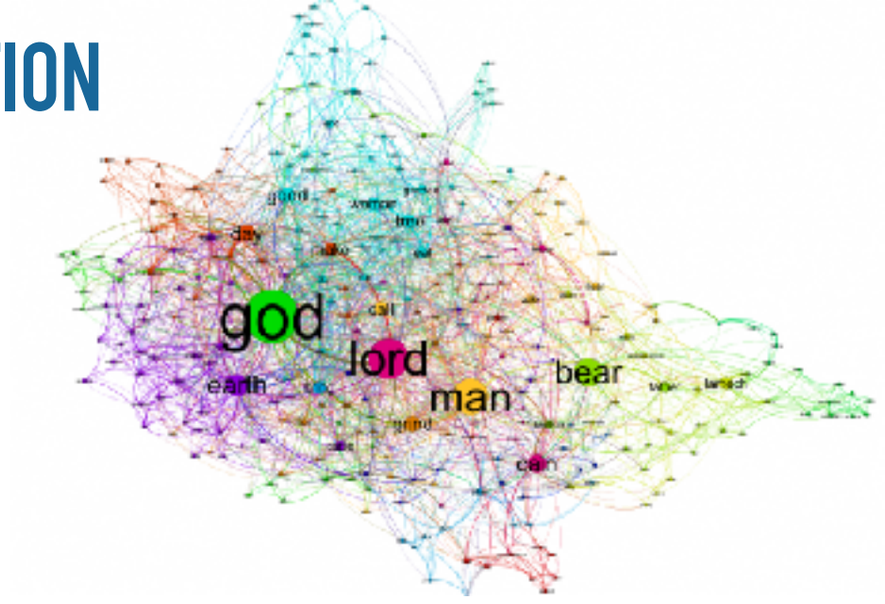
CLUSTERING



CUSTOMER SEGMENTATION

RECOMMENDER SYSTEMS





LATENT SEMANTIC ANALYSIS
TOPIC DETECTION