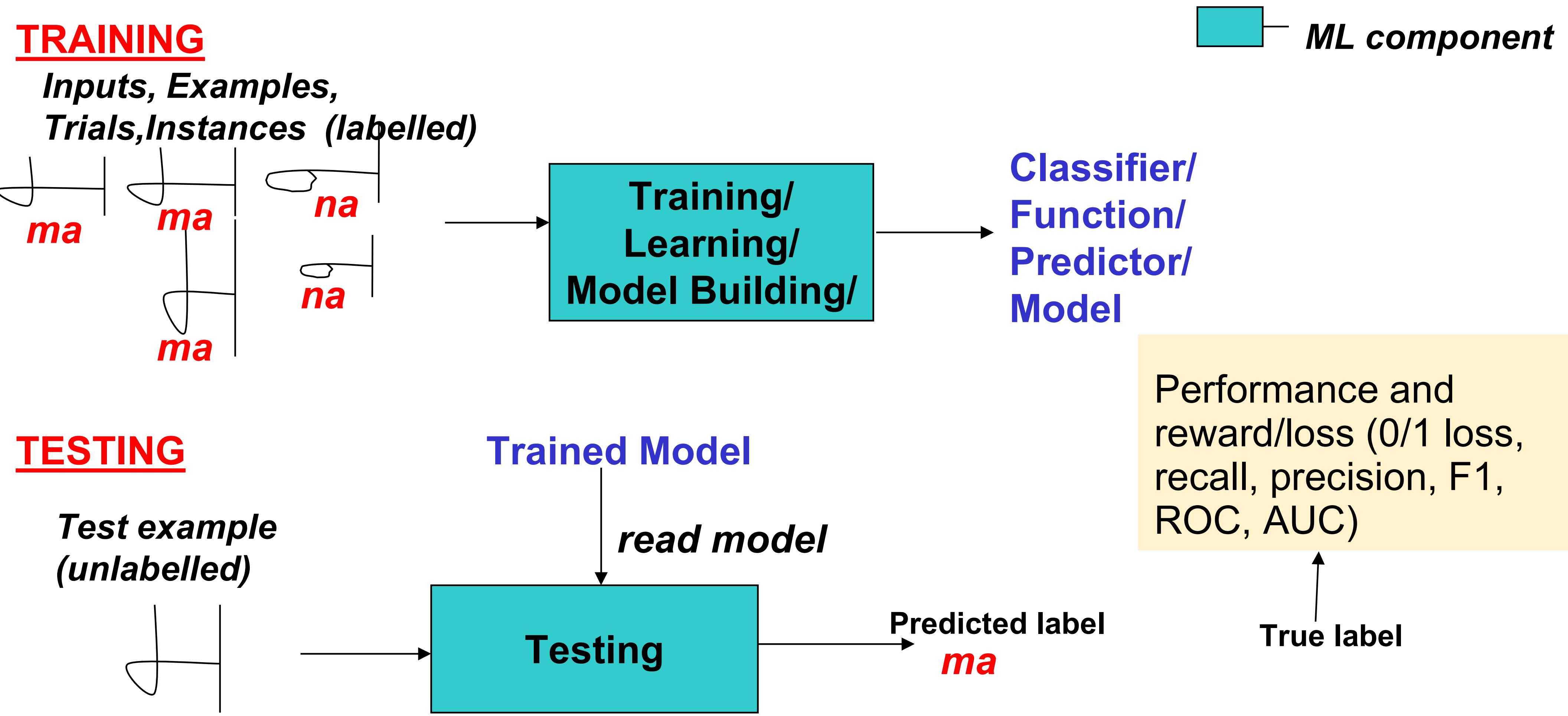
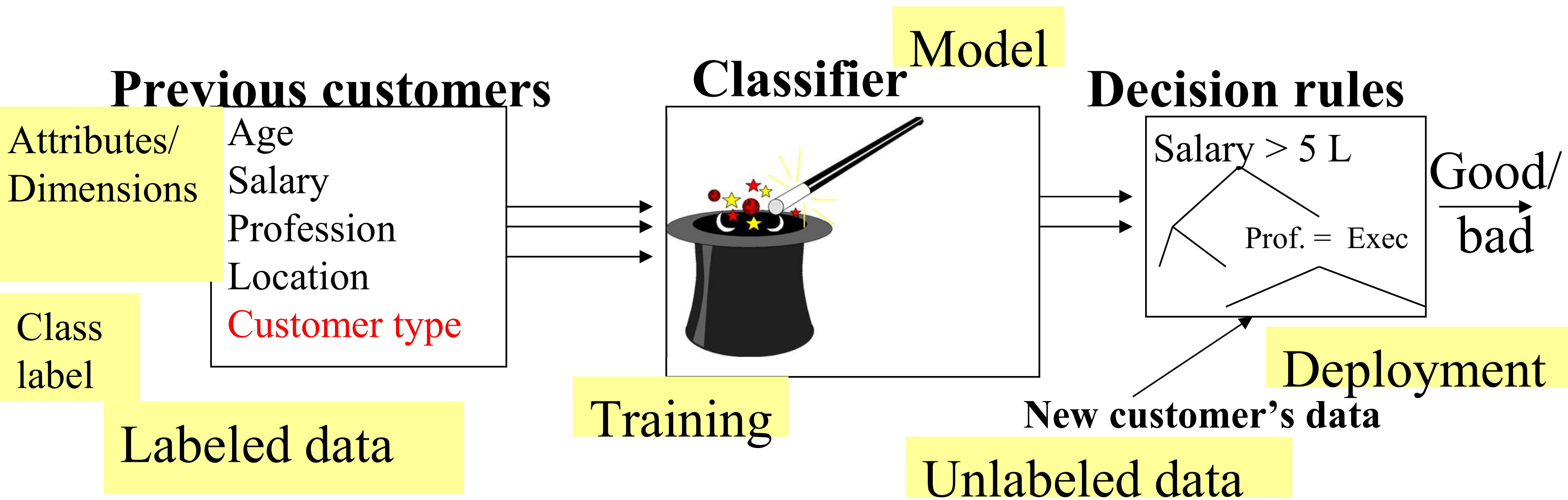
Linear Classifiers
or
Multiclass logistics regression classifiers

Classification



Classification example

- Given old data about customers and payments, predict new applicant's loan eligibility.

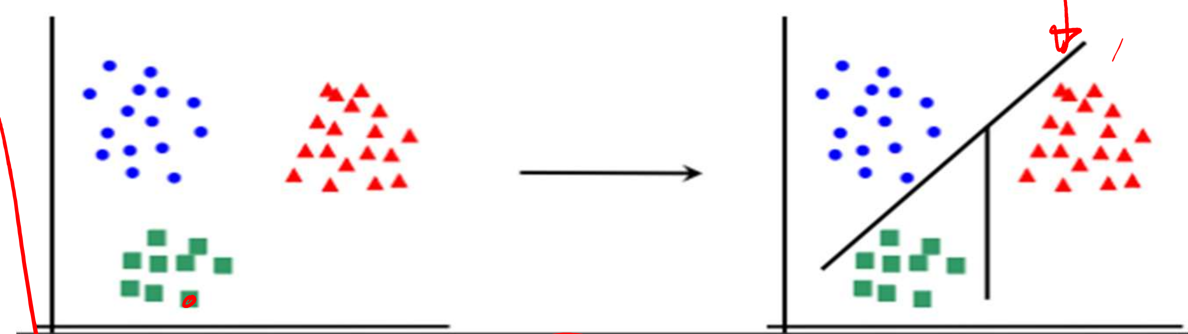


Classification: setup

- Input: Like in regression, obtained after feature engineering $x \in R^d$
- Output: one of K possible class labels $y \in [1,2, \ldots, K]$
- Training data:

$$X = \begin{bmatrix} x^1_1, & x^1_2 & \ldots & x^1_d \\ \vdots & & & \\ x^N_1, & x^N_2 & \ldots & x^N_d \end{bmatrix}$$

classifier

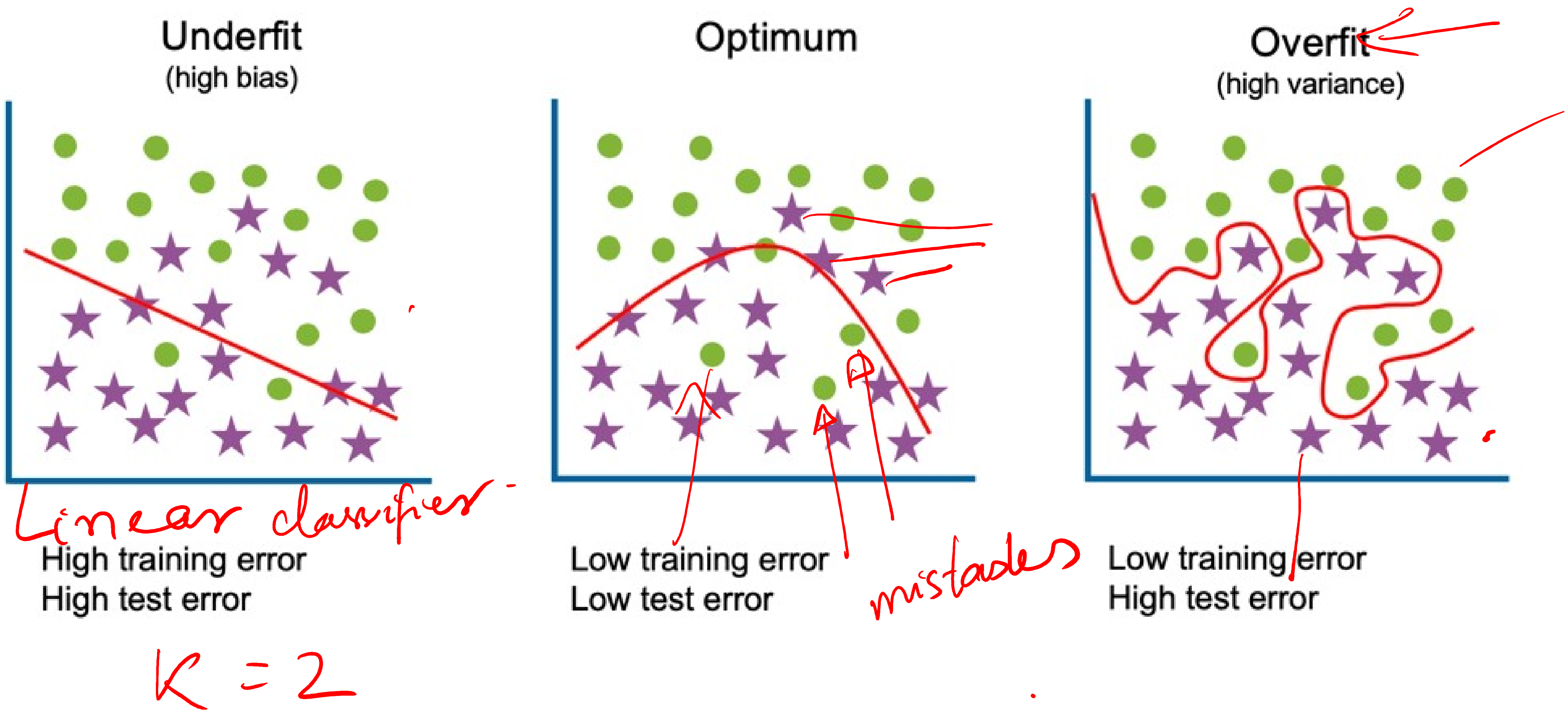


An example training dataset d=2,K=3

$$Y = \begin{bmatrix} 1 \\ 3 \\ \vdots \end{bmatrix}$$

Classifier: $C(x) \rightarrow \hat{y} \in [1 \ldots K]$

Over-fitting in classifiers



From: https://www.javatpoint.com/overfitting-in-machine-learning

Types of classifiers

- Discriminative
  - Logistic regression
  - Decision trees
  - Neural Network
- Probabilistic
  - Generative
  - Conditional
- Kernel-based
  - Nearest neighbor classifier
  - Support Vector Machines

Linear classifiers

$d+1$

$K(d+1)$

- For each class k, define a linear function $\in [1, \ldots K]$
  - $f_k(\boldsymbol{x}) = w_{k1}x_1 + \cdots w_{kd}x_d + w_{k0}$

scoring functions, logits

- Assign predicted label $\hat{y}$ as the class for which score is maximized.

$$\hat{y} = \underset{k \in [1 \ldots K]}{\operatorname{argmax}} f_k(x)$$

Example

- Letter recognition

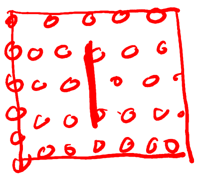
Features:

x_1 = # of vertical lines.

x_2 = # of circles

x_3 = # of horizontal lines

x_4 = # of slanted lines.



$\boxed{0}$ $\boxed{4}$ $\boxed{4}$

x^2 x^3 x^4

$K = 4$

$\boxed{1}$

x^5

$x^1 : [1, 0, 0, 0]$ $x^2 = [0, 1, 0, 0]$

$x^3 = [1, 0, 1, 1]$

$x^4 = [2, 0, 1, 0]$

$x^5 = [1, 0, 1, 0]$

$w_{11}, w_{12}, w_{13}, w_{14} = [10 \ \ -1 \ \ -1, -1]$, $w_{10} = 0$

$w_{21}, w_{22}, w_{23}, w_{24} = [-1, 10, -1, -1]$ $w_{20} = 0$

$w_{31}, w_{32}, w_{33}, w_{34} = [5, 0, 10, 5]$ $w_{30} = 0$

$\vdots$

$$f_1(x^4) = 18$$

$$f_2(x^4) = -1$$

$$f_3(x^4) = 20$$

$$\underset{k \in [1,2,3]}{\operatorname{argmax}}\left(f_k(x)\right) = \hat{y} = 3$$

$$y = 3$$

✓

Special case of binary classifiers

$k = 2$

$f_1(x) = w_{11}x_1 + w_{12}x_2 + \cdots w_{1d}x_d + w_{10}$

$f_2(x) = w_{21}x_1 + w_{22}x_2 + \cdots w_{2d}x_d + w_{20}$

$\underset{k=(1,2)}{\arg\max} f_k(x) = \underline{\arg\max}(\underline{f_1(x)}, \underline{f_2(x)}) \equiv \arg\max(f_1(x) - f_2(x), 0)$

$f_1(x) = (w_{11} - w_{21})x_1 + (w_{12} - w_{22})x_2 + \cdots + (w_{10} - w_{20})$

$= (\vec{w}_1 - \vec{w}_2)\vec{x} + (w_{10} - w_{20})$

$\rightarrow w = \vec{w}_1 - \vec{w}_2$

$$f(x) = w \cdot x + w_0$$

Training objective

- Find parameters such that predicted class match actual class in training data
- Consider training instance: x^i, y^i

$\hat{y} = \arg\max_y f_y(x^i; W)$ ← this needs to match true y_i

$\cancel{\max_W} \; f_{y_i}(x^i, w) - \max_y f_y(x^i, w)$ = 0 for correct prediction

$E(x^i, y^i, W) = -f_{y_i}(x^i, w) + \boxed{\max_y f_y(x^i, w)} \geq 0$

< 0 for incorrect classification.

not differentiable.

Training objective : $\text{minimize}_W \sum_{i=1}^{N} E(x^i, y^i, W)$

Differentiable rewrite with softmax

- Maximum of K real values:
- $\max(o_1, o_2, \ldots, o_K) \approx \log\left(e^{o_1} + e^{o_2} + \cdot \; e^{o_K}\right)$

HW

O_1 was the maximum

$$= o_1 + \log\left(e^{o_1 - o_1} + \underbrace{e^{o_2 - o_1}}_{\leq 1} + \underbrace{e^{o_K - o_1}}_{\leq 1}\right)$$

$$= o_1 + \log(1 + 0.0001 + \quad)$$

$$\approx o_1$$

Final objective

$$\underset{W}{\text{minimize}} \sum_{i=1}^{N} Loss(x^i, y^i, W)$$

loss $(x, y\ W)$ as the surrogate for $E(x, y\ W)$

$$= \sum_{i=1}^{N} -f_{y_i}(x^i, W) + \log \sum_{y=1}^{K} e^{f_y(x^i, W)}$$

Demo

- https://colab.research.google.com/drive/1MtvbOIHgnkV05GJXfefisim6TiCIEXts?usp=sharing