

Classification

Machine Learning Course - CS-433

~~7/8~~ Oct 2025

Robert West

(Slide credits: Nicolas Flammarion)



Definition of classification

We observe some data $S = \{x_n, y_n\}_{n=1}^N \in \mathcal{X} \times \mathcal{Y}$

\mathcal{Y}
Discrete Set

So far:
 $\mathcal{Y} = \mathbb{R}$

Goal: given a new observation x , we want to predict its label y

How:



$$S = \{x_n, y_n\}_{n=1}^N$$

Discrete \mathcal{Y}

\mathcal{A}

$$f_S = \mathcal{A}(S)$$

Classification: relates input to a categorical variable

$$(x, y) \in \mathcal{X} \times \underbrace{\mathcal{Y}}_{\text{Discrete Set}}$$

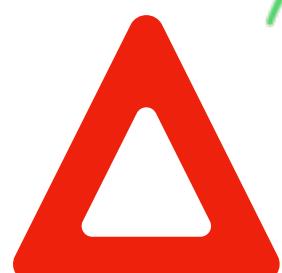
Binary Classification: y can take two values

$$\{1, 2\}$$

$\cancel{K=2}$ $y \in \{c_1, c_2\}$ where c_i are the class labels. We often use $\{0, 1\}$ or $\{-1, 1\}$

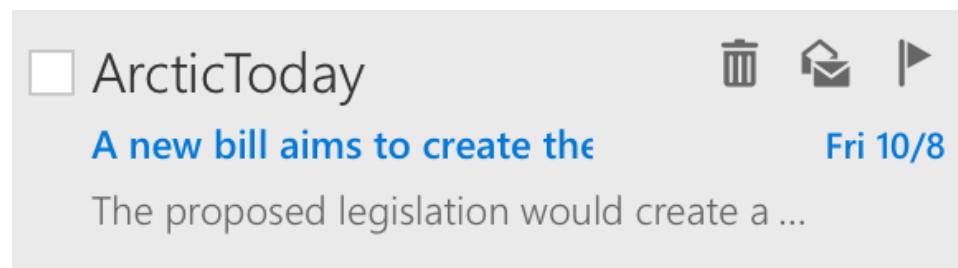
Multi-class classification: y can take more than two values

$y \in \{c_1, \dots, c_K\}$ for a K classes problem. We often use $\{1, \dots, K\}$



no ordering between classes

Spam Detection



Patterns – Cell Press
Meet the newest members c Fri 10/8
The science of data Can't see this email p...

Volkan Cevher
▶ EPFL-CIS RIKEN-AIP Joint Talks on Thu 3:08 PM
Dear All, I hope all is well. I am sending t...

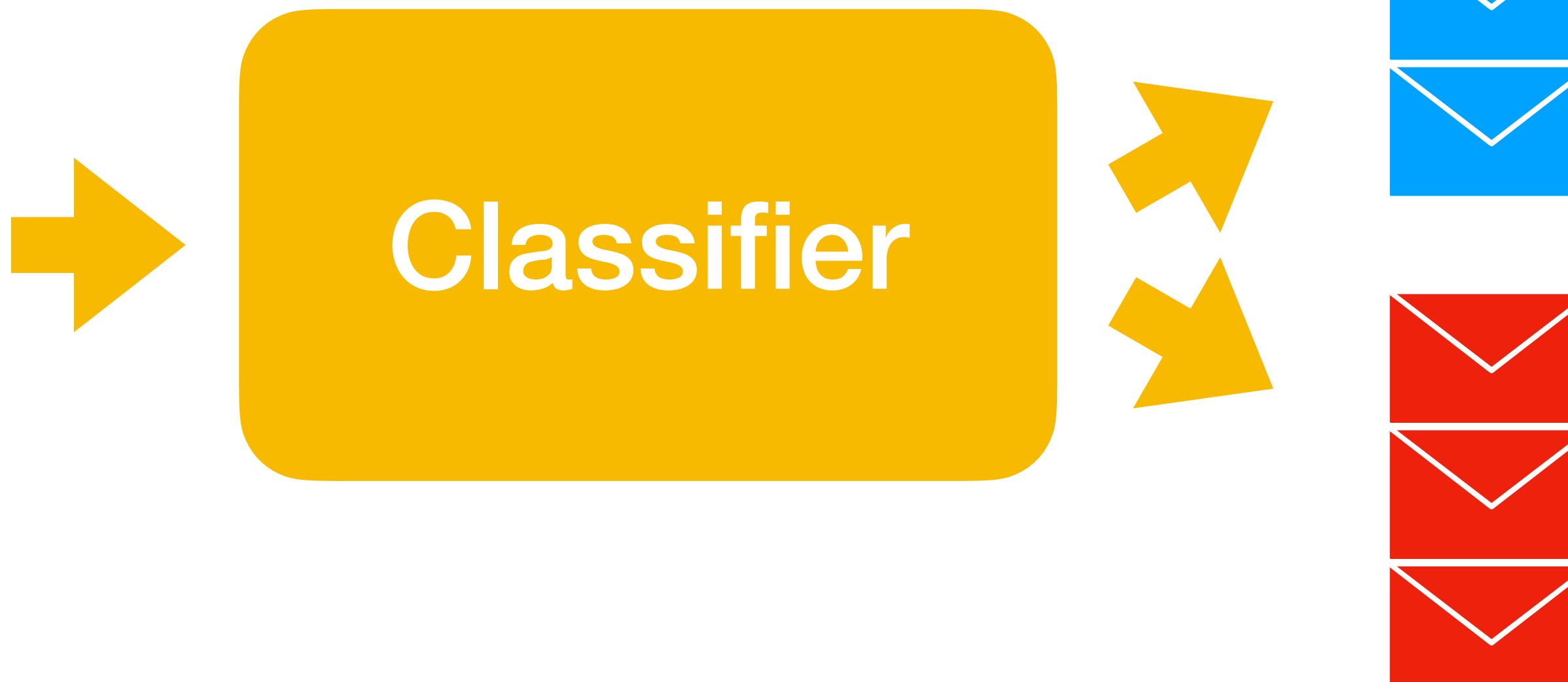
sae.amenagements@...
Aménagements d'études pour ce Thu 2:36 PM
English version below Madame, Monsieu...

westernunionrespo...
Western Union: Please verify 9/26/2021
Dear JULES ADAM, We noticed that you ...

Bachmann Jennifer
▶ coffee 10:35 AM
Hello, C'est Goodlife Coffee 😊 Ils m'ont...

Gestion de l'Ecole do... 0
Workshops for mentors 8:33 AM
Dear Thesis Directors, We hope our ema...

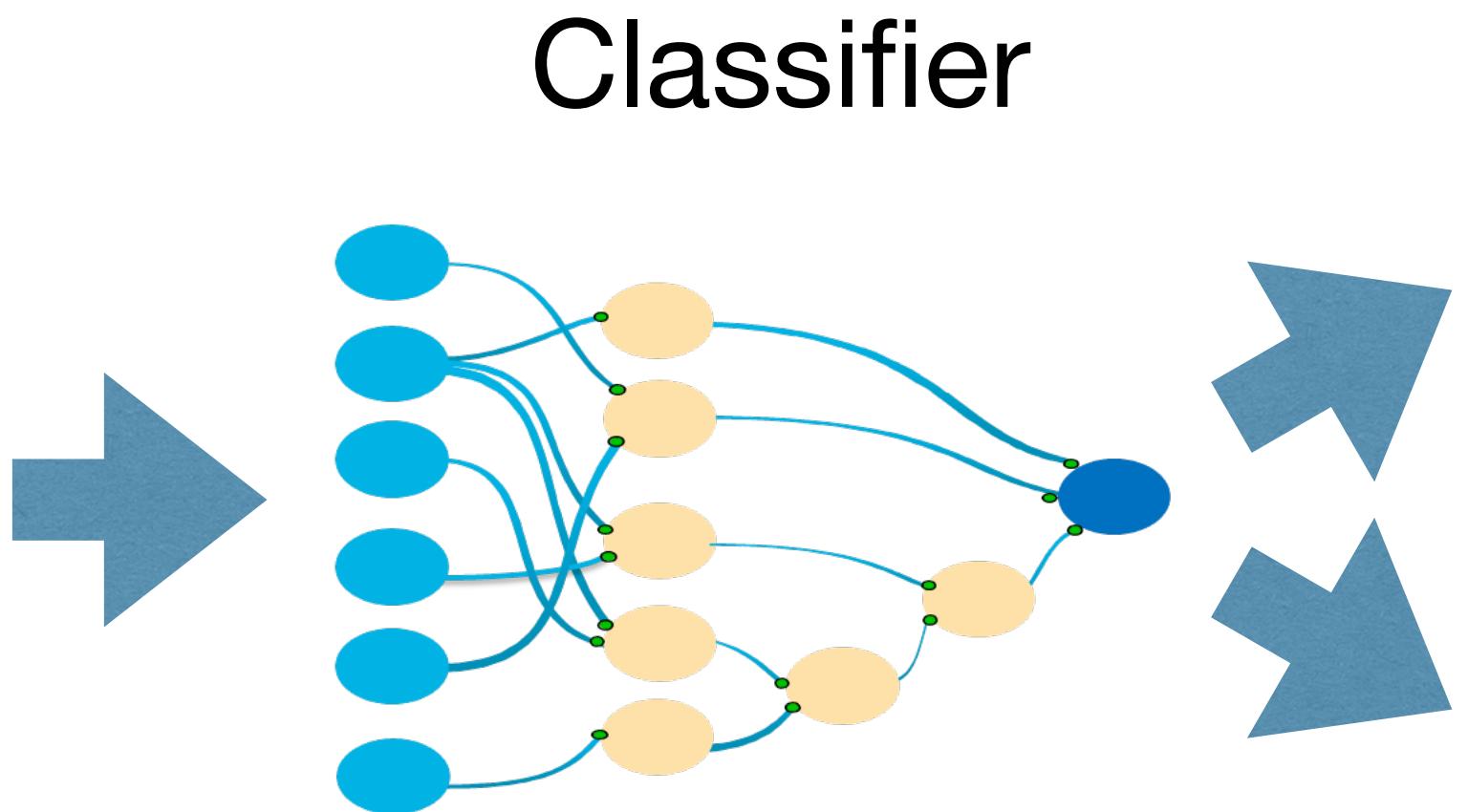
The Boston Globe
Parenting Unfiltered: Your W 2:00 AM
Beer. Blankets. Burgers. Dumplings! ...



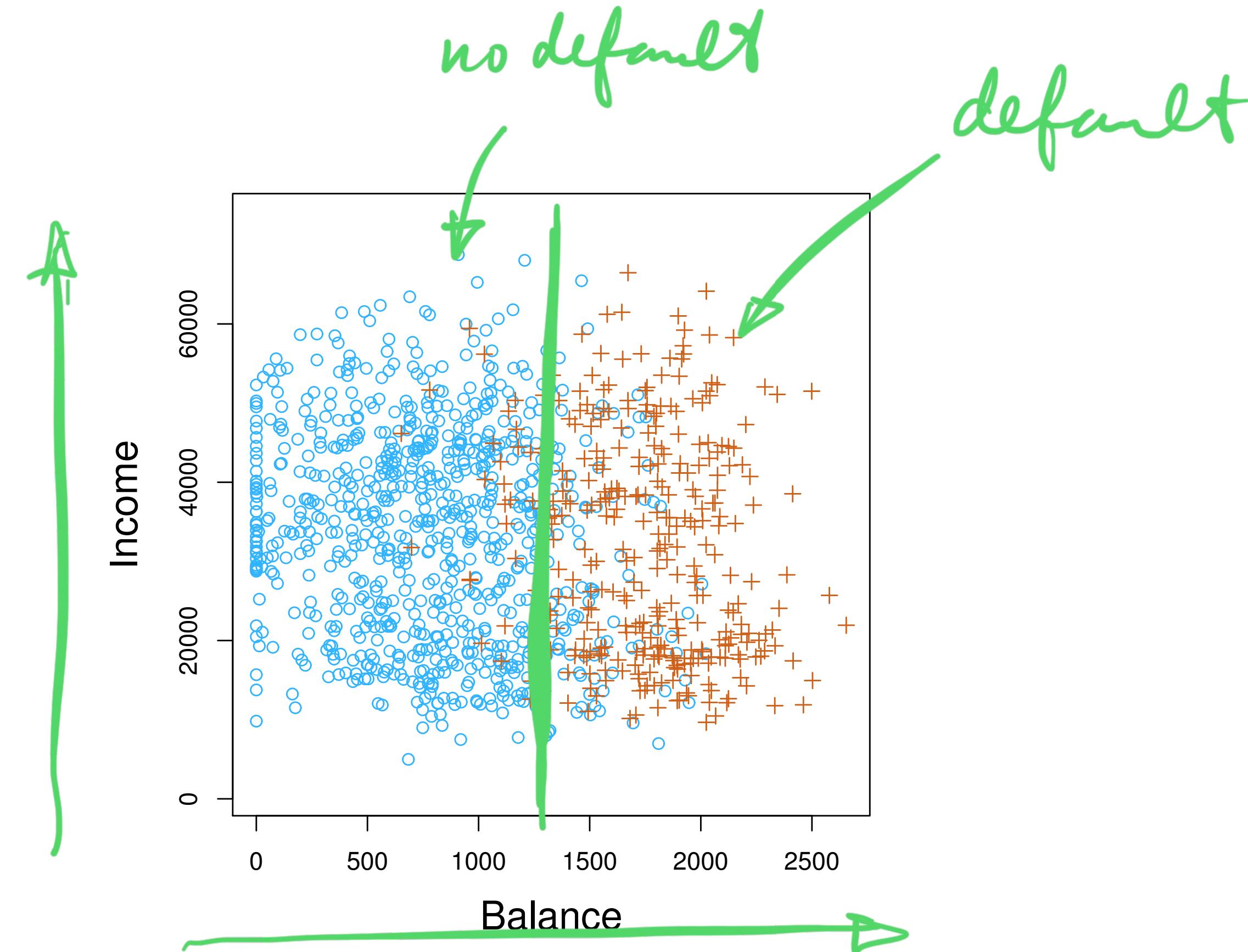
Inbox

Spam Folder

Image classification



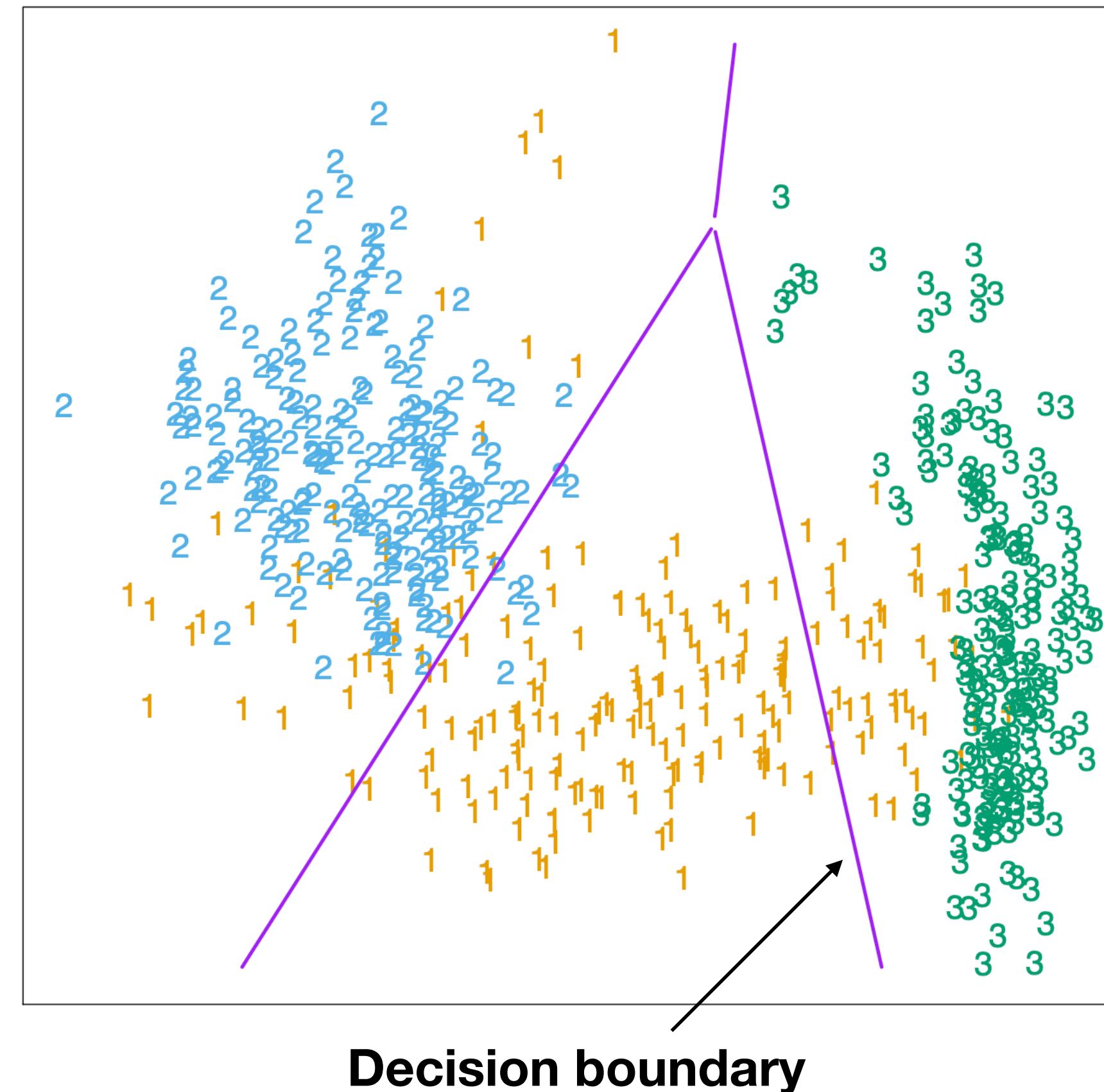
Credit Card Default



- + individual who defaulted on their credit card payments
- individual who did not

Classifier

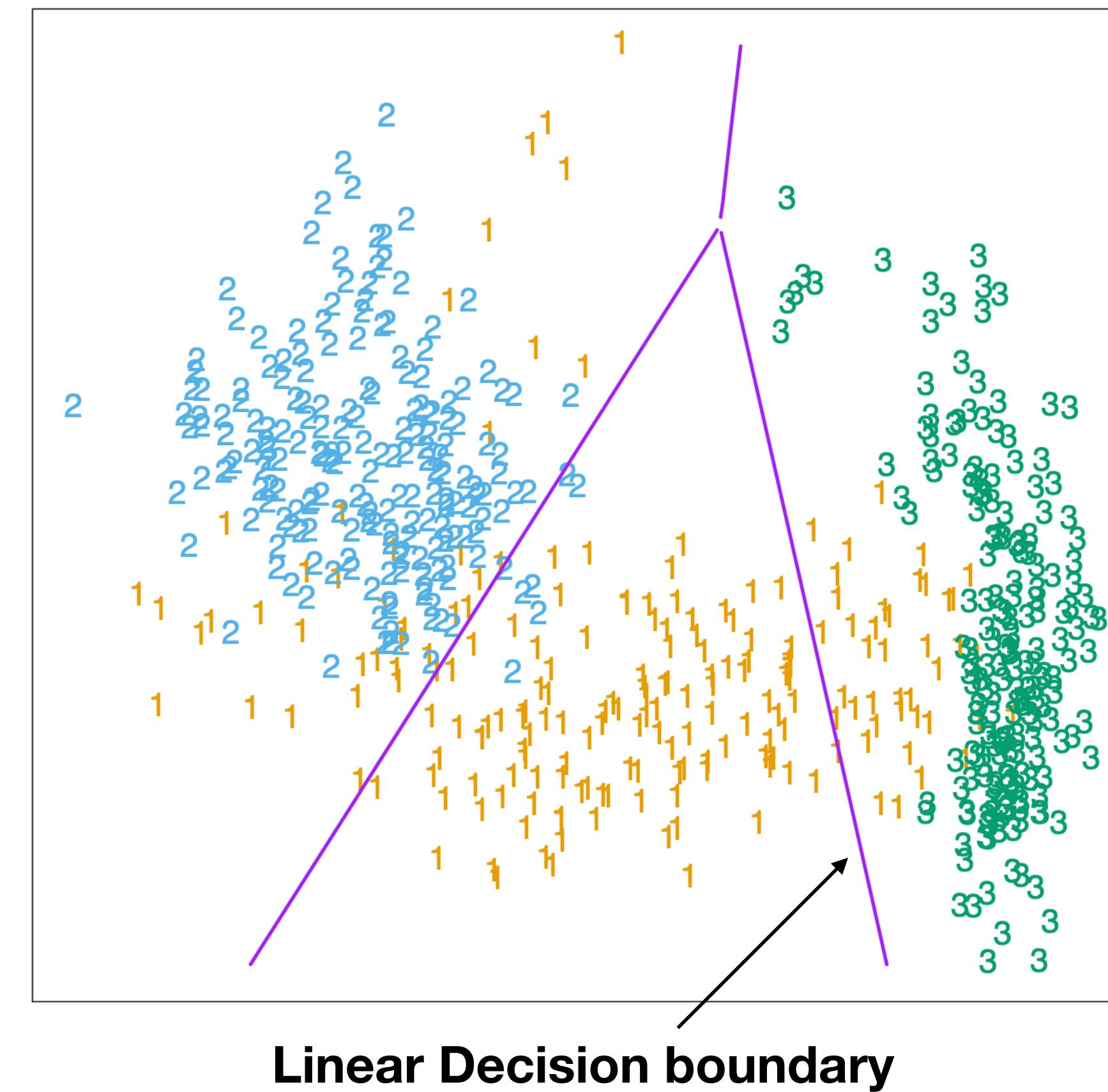
A classifier $f: \mathcal{X} \rightarrow \mathcal{Y}$ divides the input space into a collection of regions belonging to each class



Classifier

A classifier $f: \mathcal{X} \rightarrow \mathcal{Y}$ divides the input space into a collection of regions belonging to each class

It can be linear

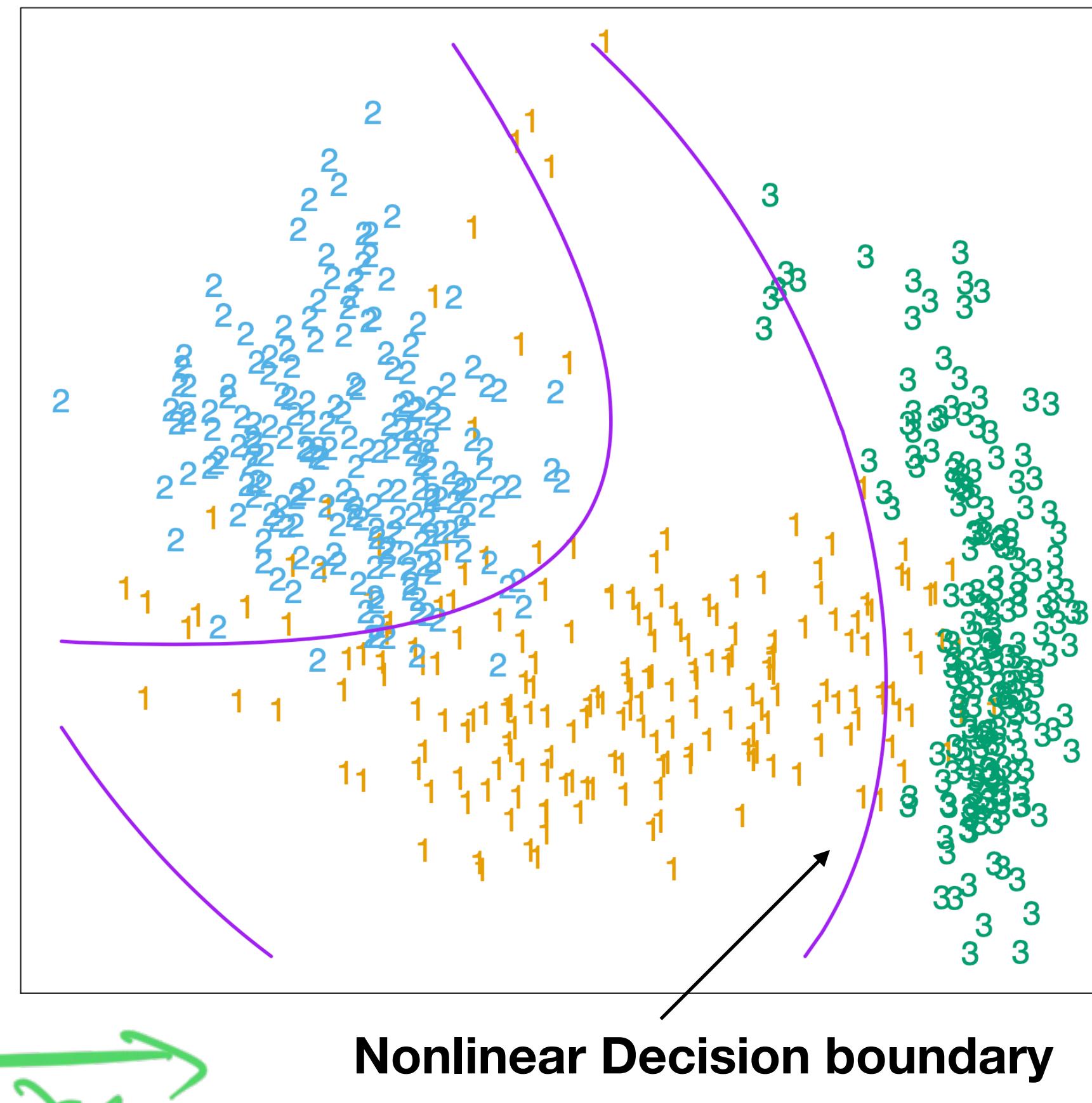


Classifier

A classifier $f: \mathcal{X} \rightarrow \mathcal{Y}$ divides the input space into a collection of regions belonging to each class

It can also be nonlinear

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



Lin.:

$$f_w(x) = w^T x$$

$$(x_1, x_2, x_1^2, x_2^2, x_1 x_2 \dots)$$

Classification: a special case of regression?

Classification is a **regression problem** with discrete labels:

$$(x, y) \in \mathcal{X} \times \{0, 1\} \subset \mathcal{X} \times \mathbb{R}$$

Reg.: $y = \mathbb{R}$

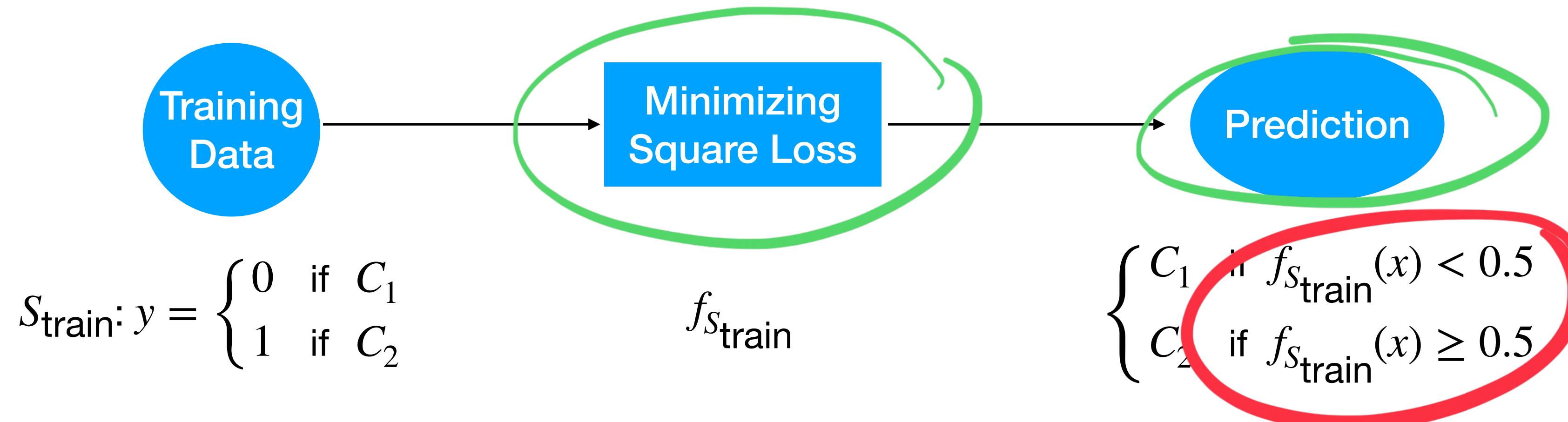
Could we use previously seen regression methods to solve it?

Classification: a special case of regression?

Classification is a **regression problem** with discrete labels:

$$(x, y) \in \mathcal{X} \times \{0, 1\} \subset \mathcal{X} \times \mathbb{R}$$

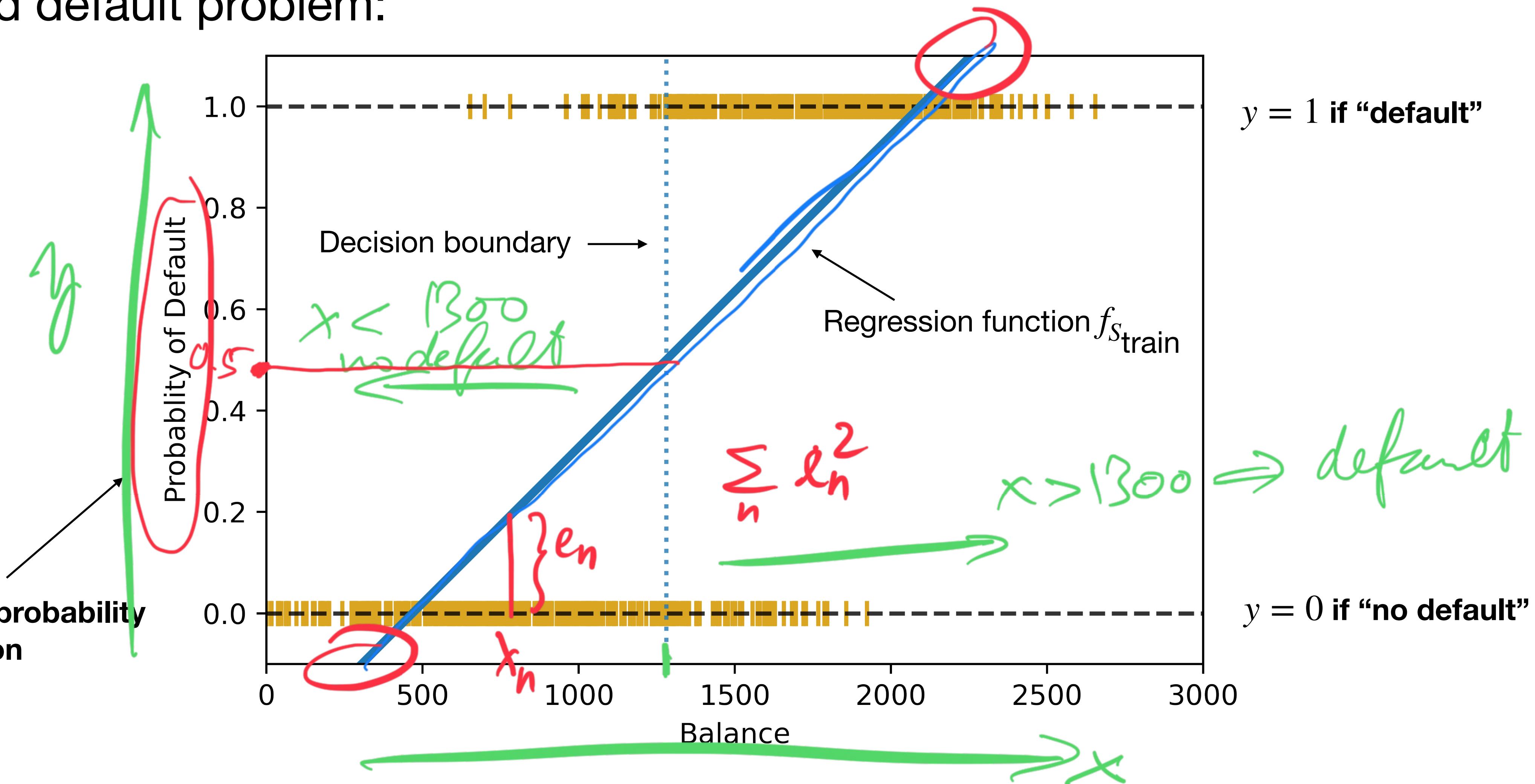
Could we use previously seen regression methods to solve it?



Is it a good idea?

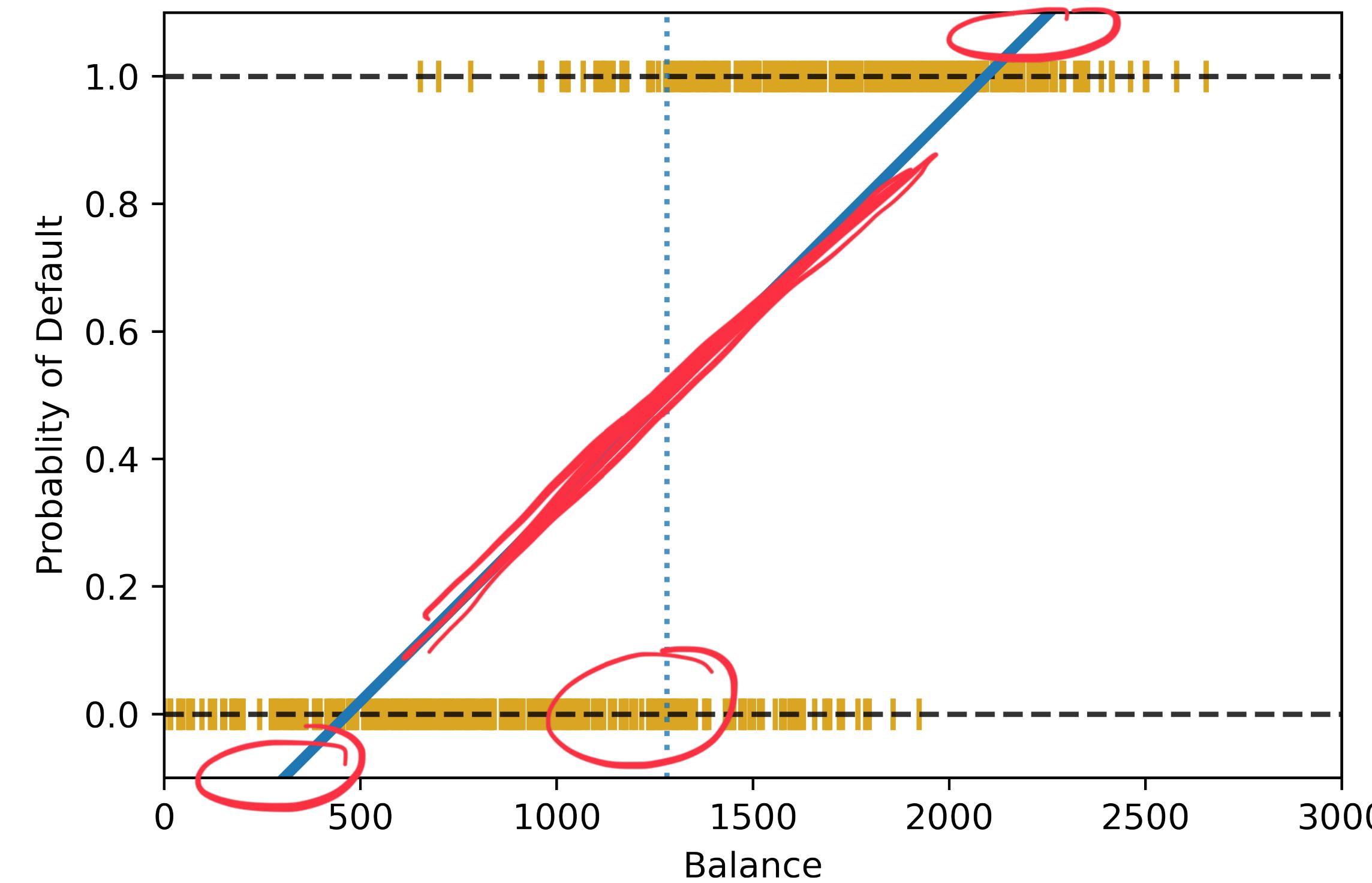
$$d = 1$$
$$\mathcal{X} = \mathbb{R}$$

Credit-card default problem:



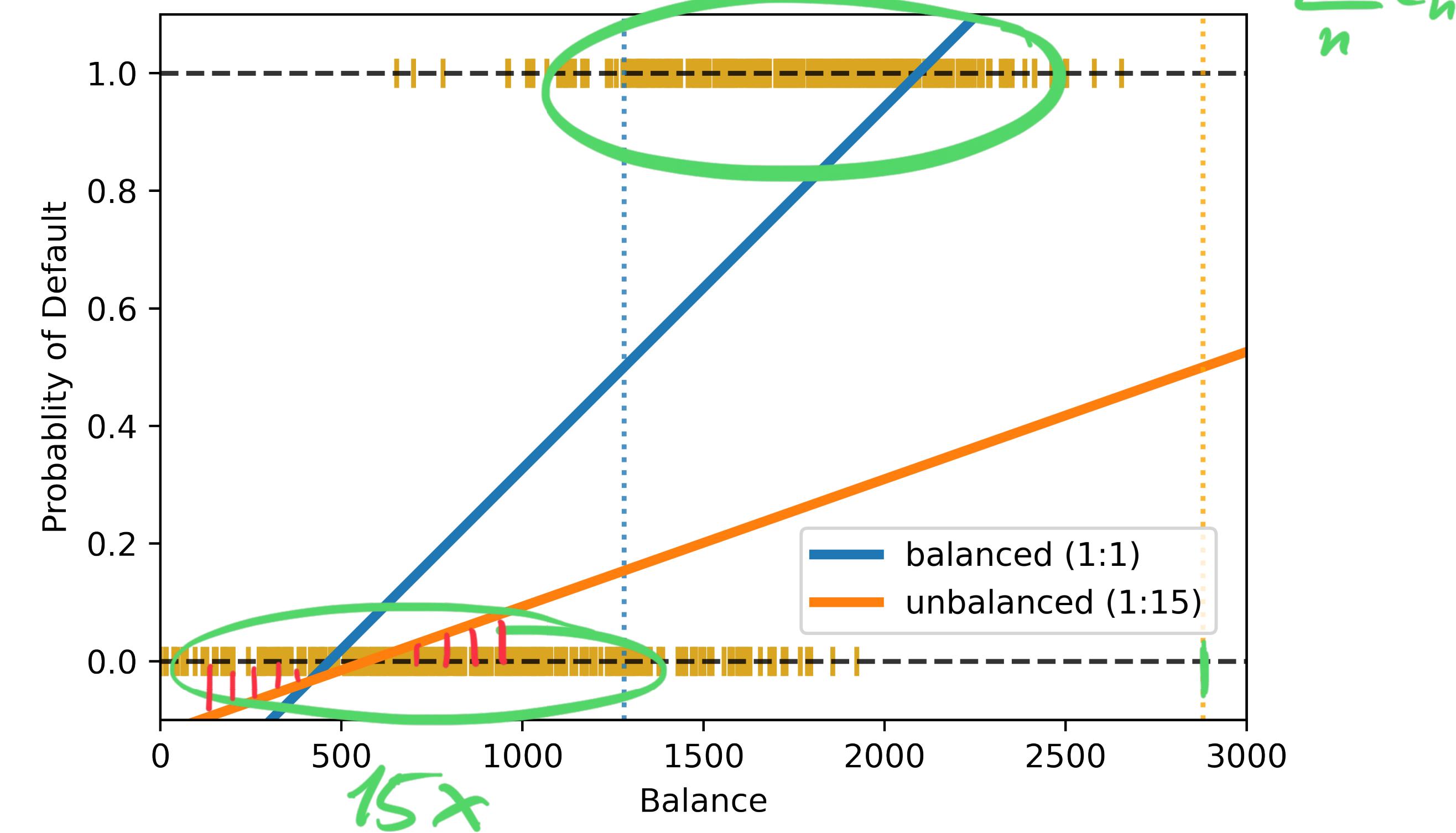
Classification is not just a special form of regression

A. The predicted values are not probabilities (not in $[0,1]$)



Classification is not just a special form of regression

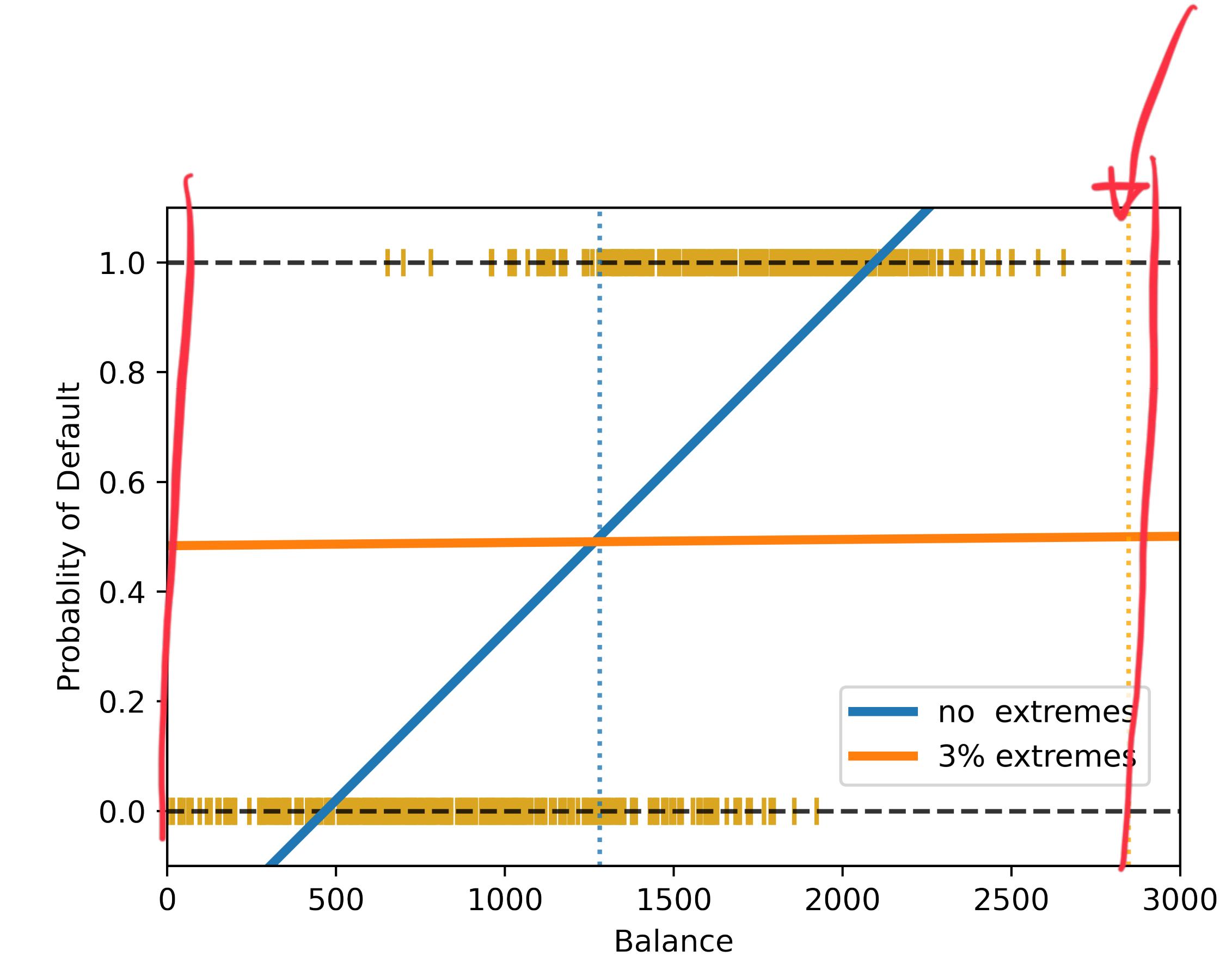
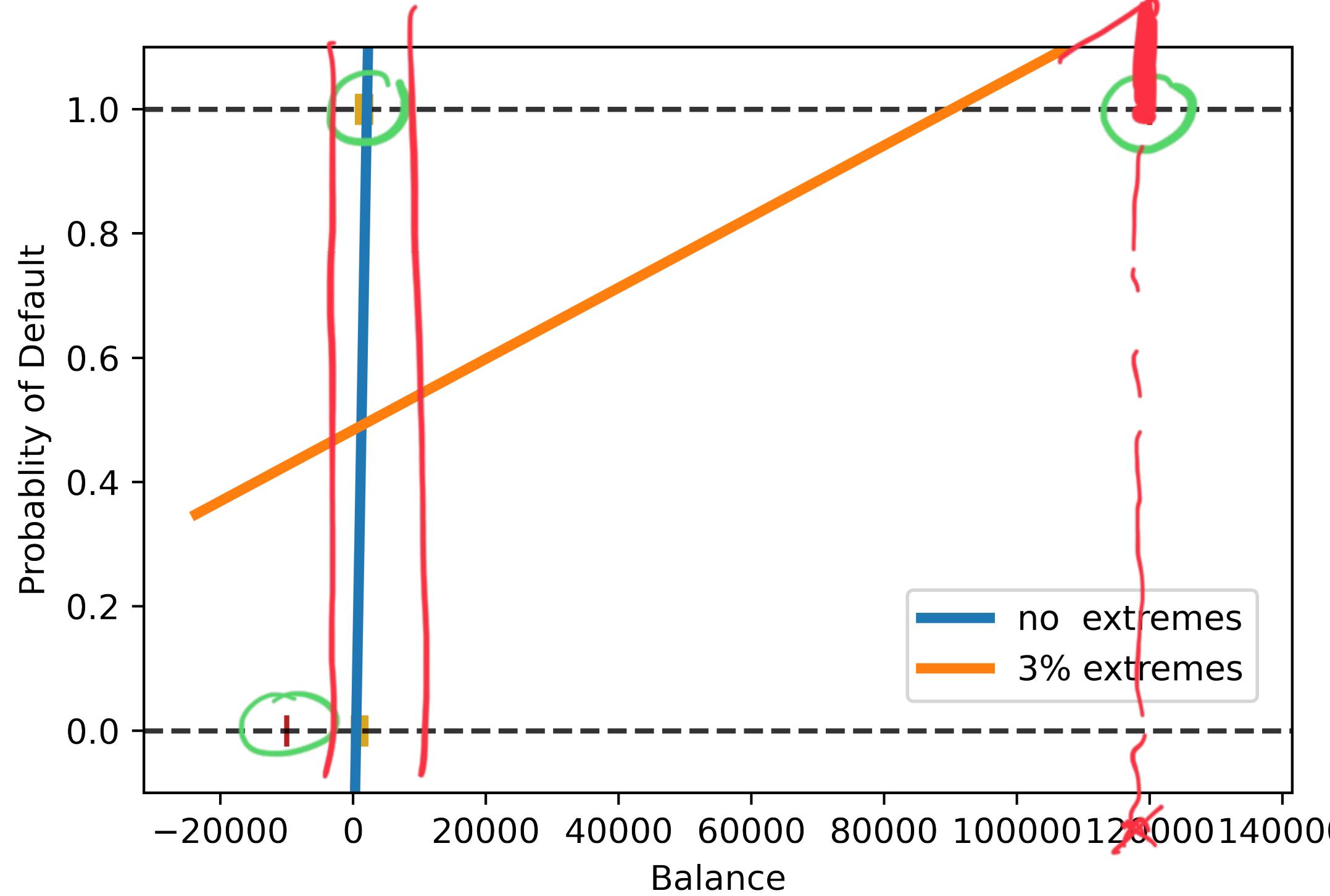
B. Sensitivity to unbalanced data



The position of the line depends crucially on how many points are in each class

Classification is not just a special form of regression

C. Sensitivity to extreme values:

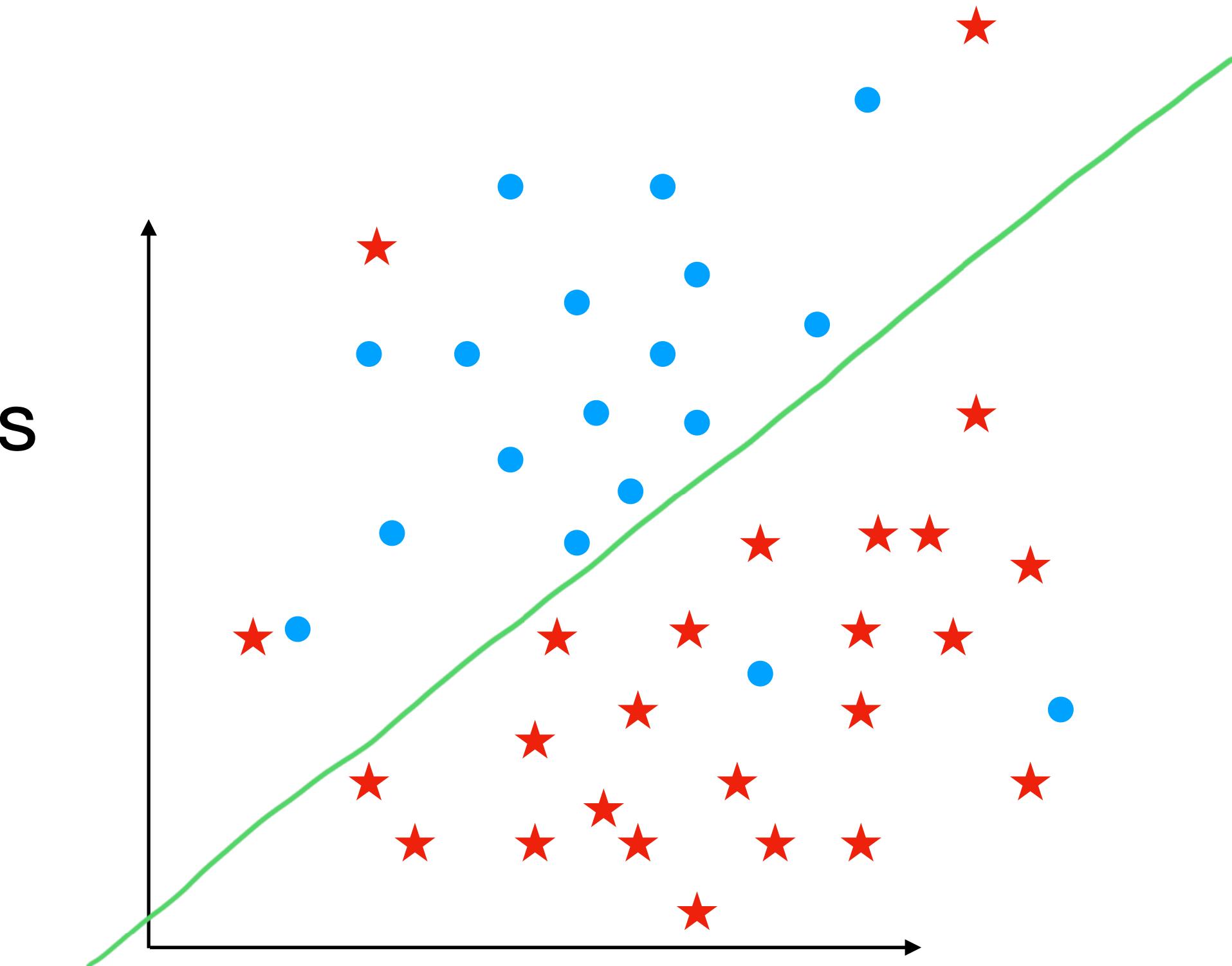


The position of the line depends crucially on where the points lie

Why: the square loss we used for regression is not suitable for classification

How to perform classification?

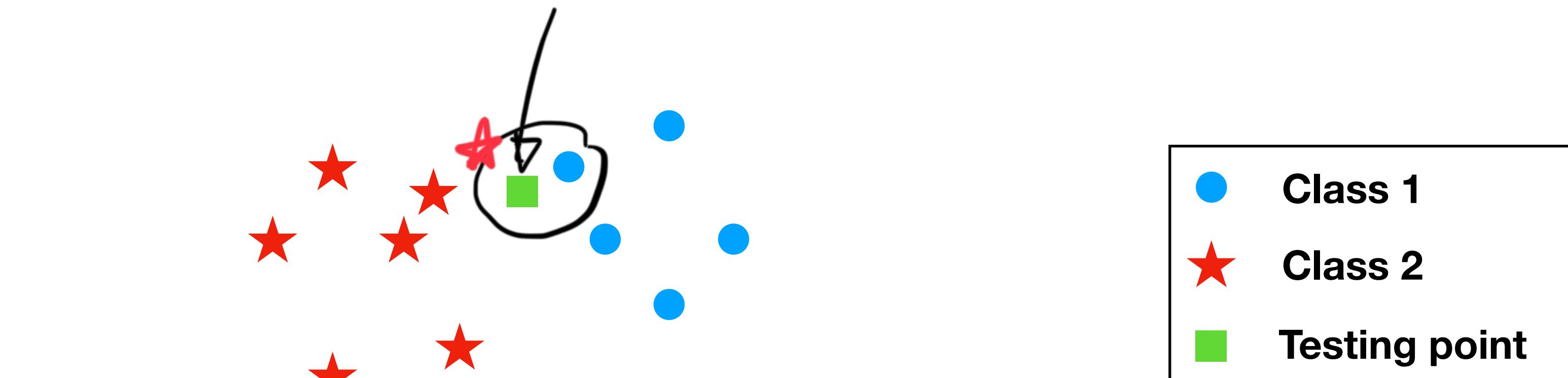
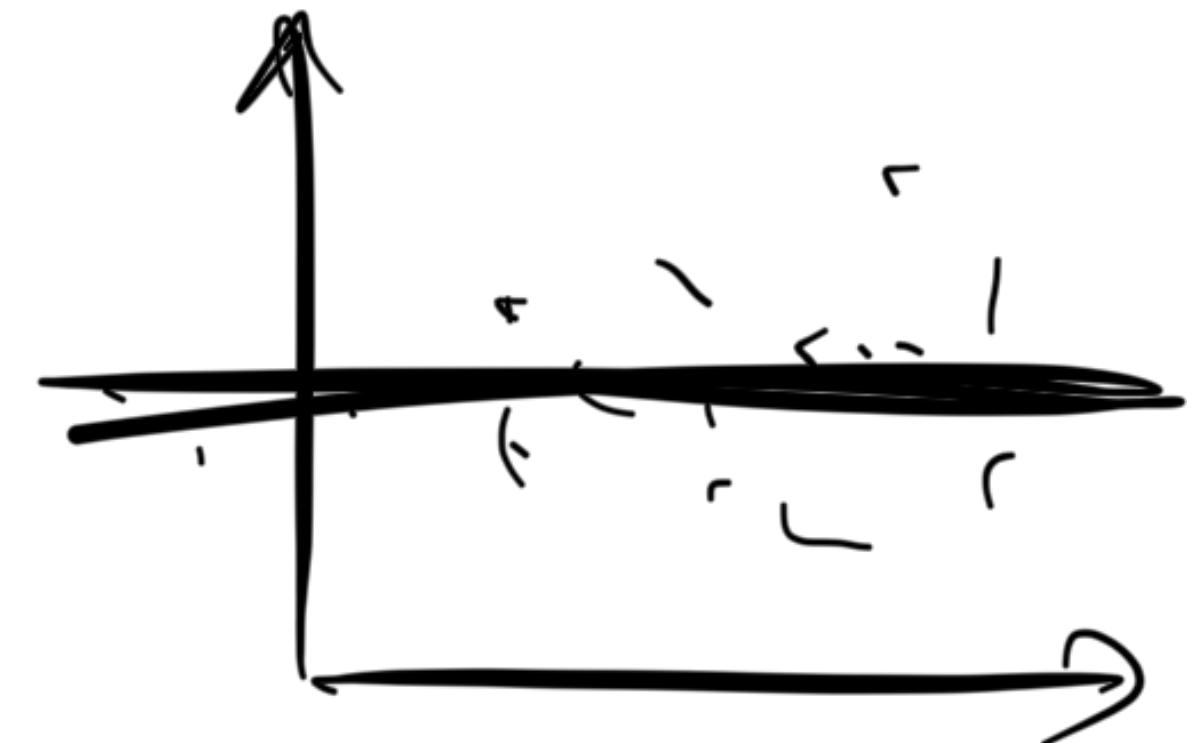
- Many approaches have been developed
 - We won't cover them in detail today
 - Instead, we will provide quick introductions
-
- Fundamental task of classification:



Divide the space into distinct decision regions

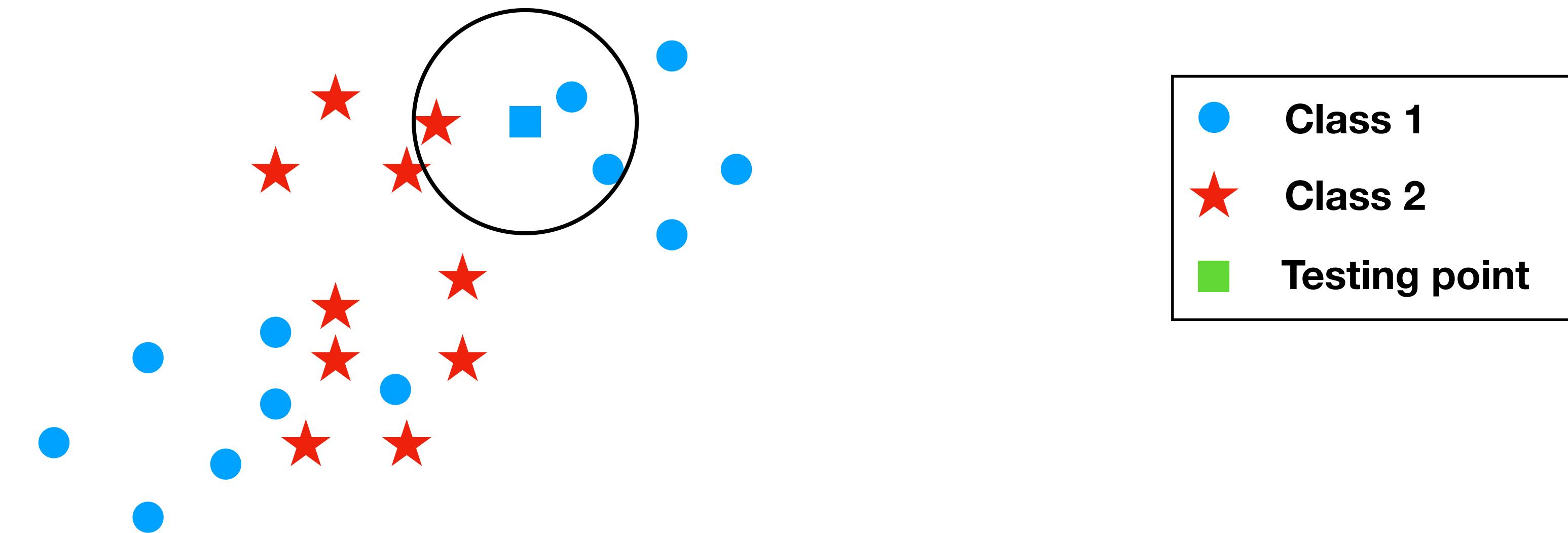
k-Nearest Neighbor

Loo CV



k-Nearest Neighbor

Assume that nearby points are likely to have similar labels



A new point x is classified based on the **majority vote of its k -nearest neighbors**

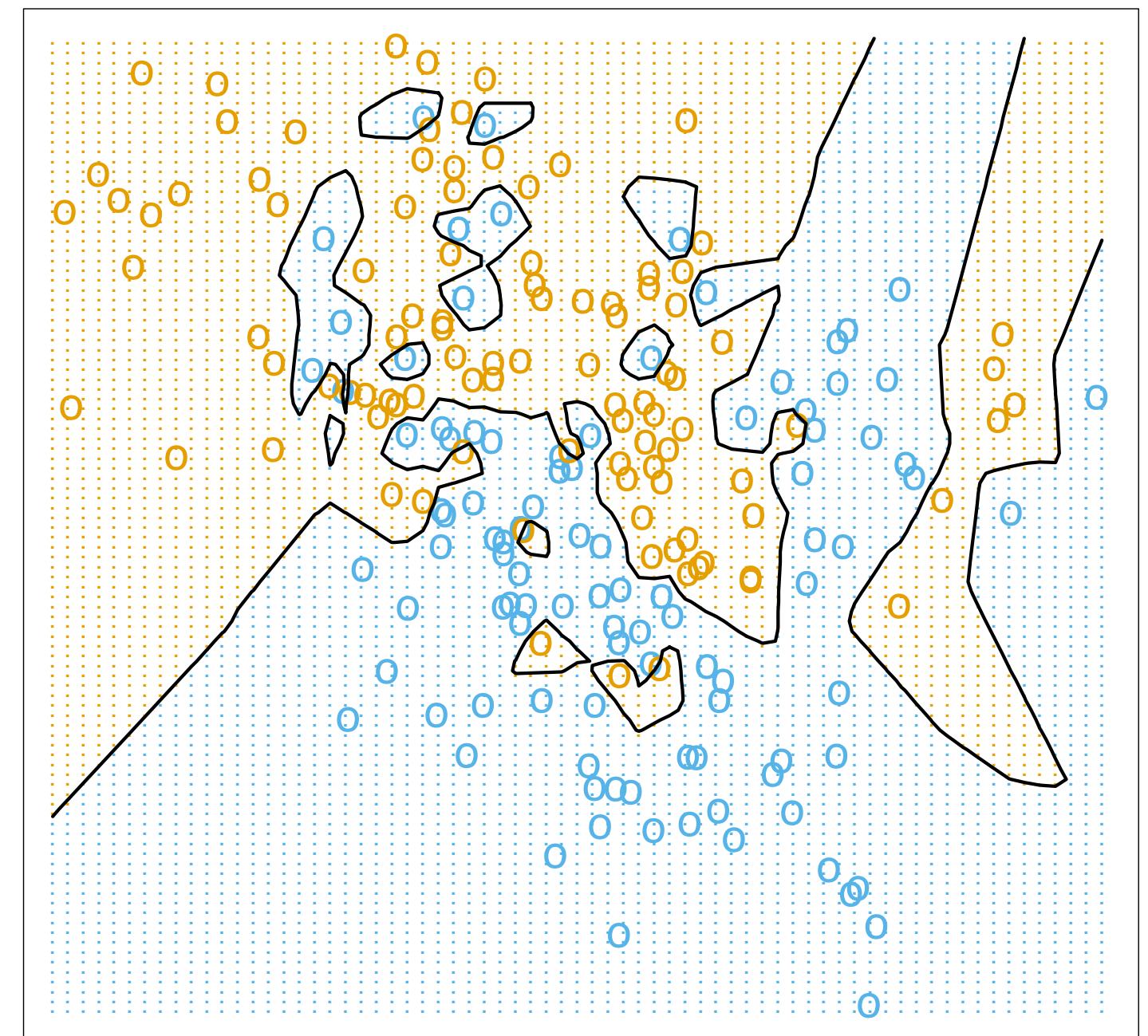
k-Nearest Neighbor

Pros:

- **No optimization** or training
- **Easy** to implement
- Works well in **low dimensions**, allowing for very complex decision boundaries

Cons:

- **Slow** at query time
- Not suitable for high-dimensional data
- Choosing the right local distance is crucial



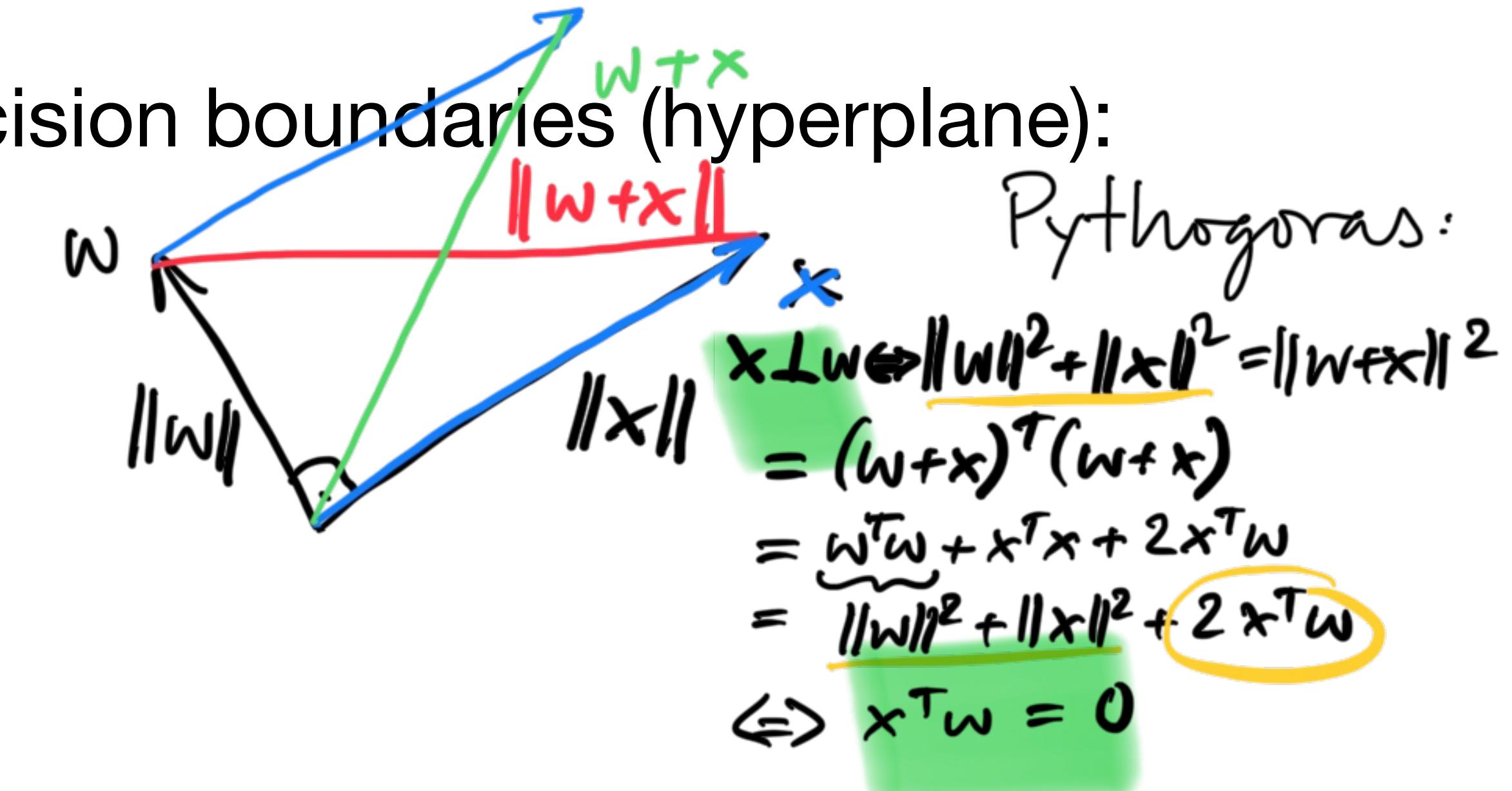
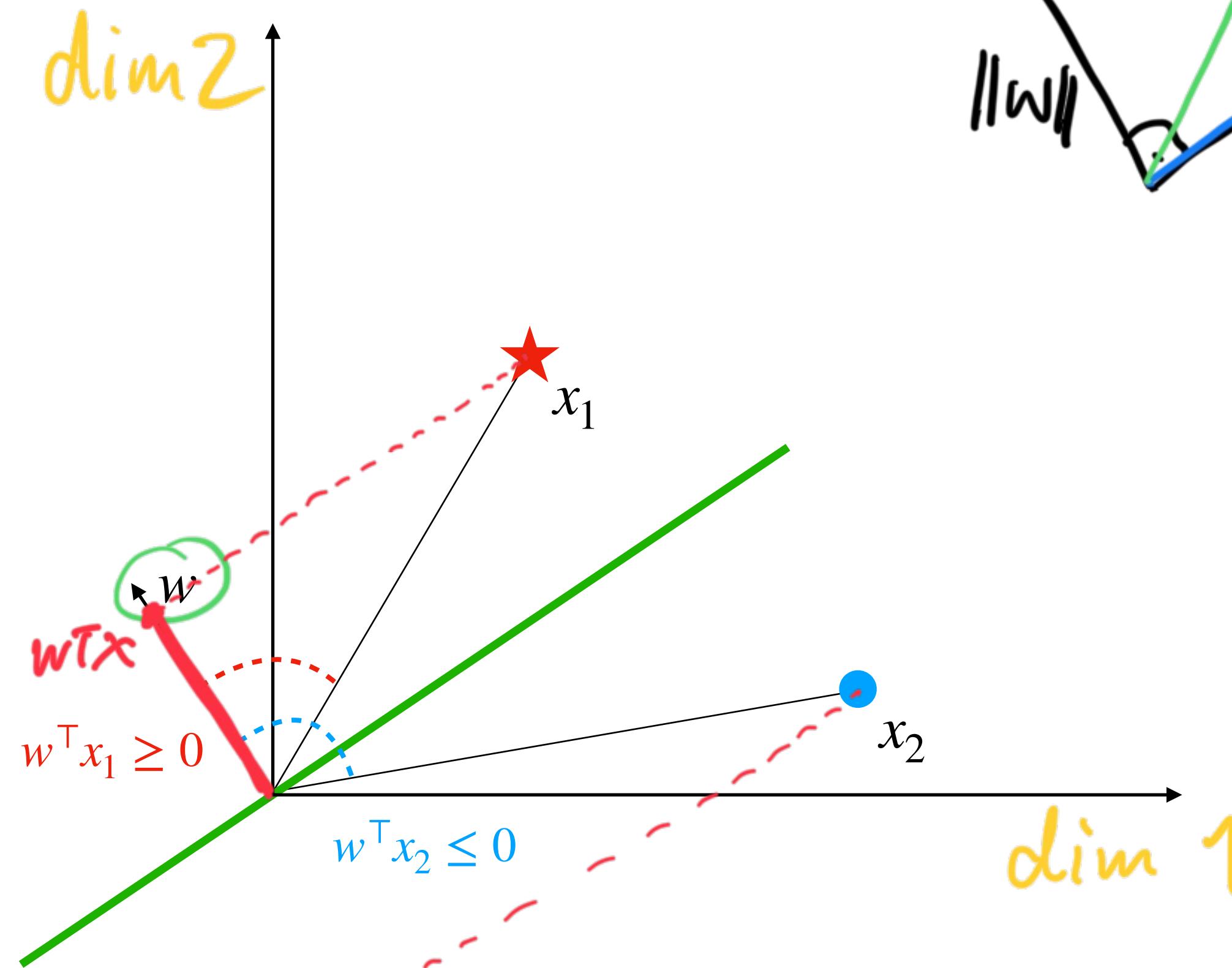
Linear Decision boundaries

Assume we restrict ourselves to linear decision boundaries (hyperplane):

→ Prediction: $f(x) = \text{sign}(x^T w)$

$d=2$

forall point x on line:
 $w^T x = 0$

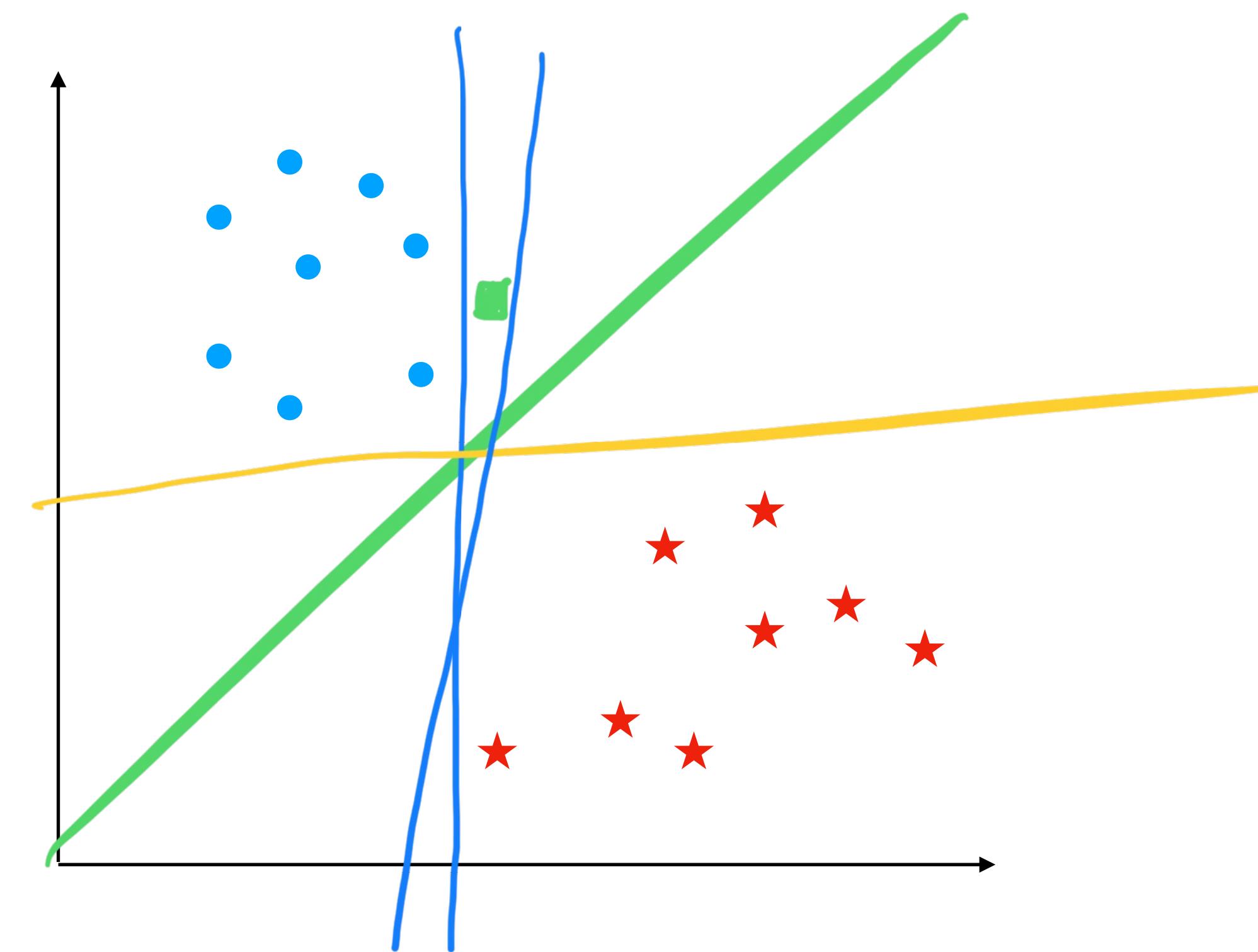


$$\begin{aligned} \|w\| &= 1 \\ w^T w &= 1 \\ (x-p)^T w &= 0 \\ x^T w - p^T w &= 0 \\ x^T w &= p^T w \\ ||p|| \|w^T w\| &= ||p|| \end{aligned}$$

Separating hyperplane

Assume we restrict ourselves to linear decision boundaries (hyperplane):

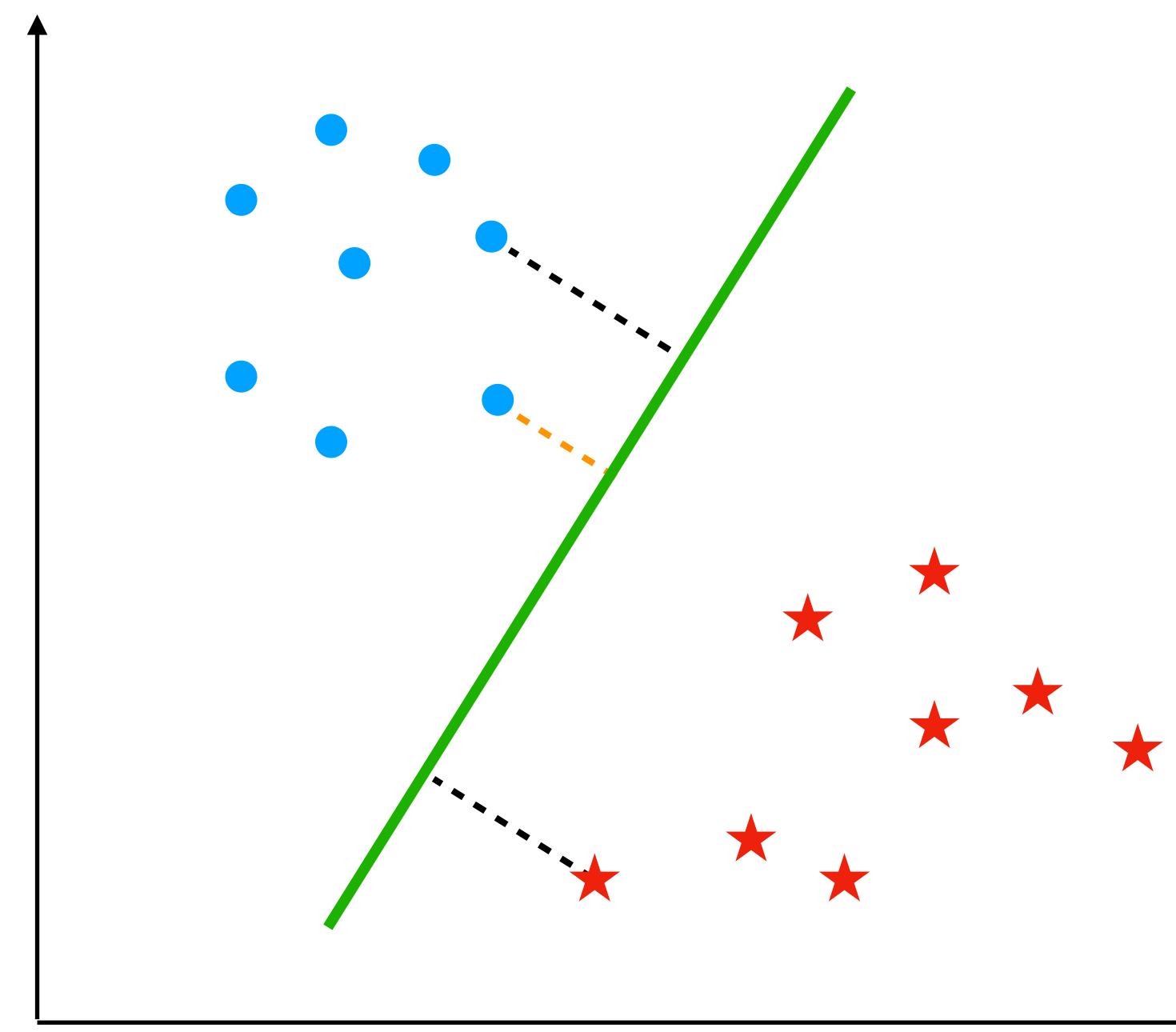
Assume the data are linearly separable, i.e., a separating hyperplane exists



Which separating hyperplane would you pick?

Margin

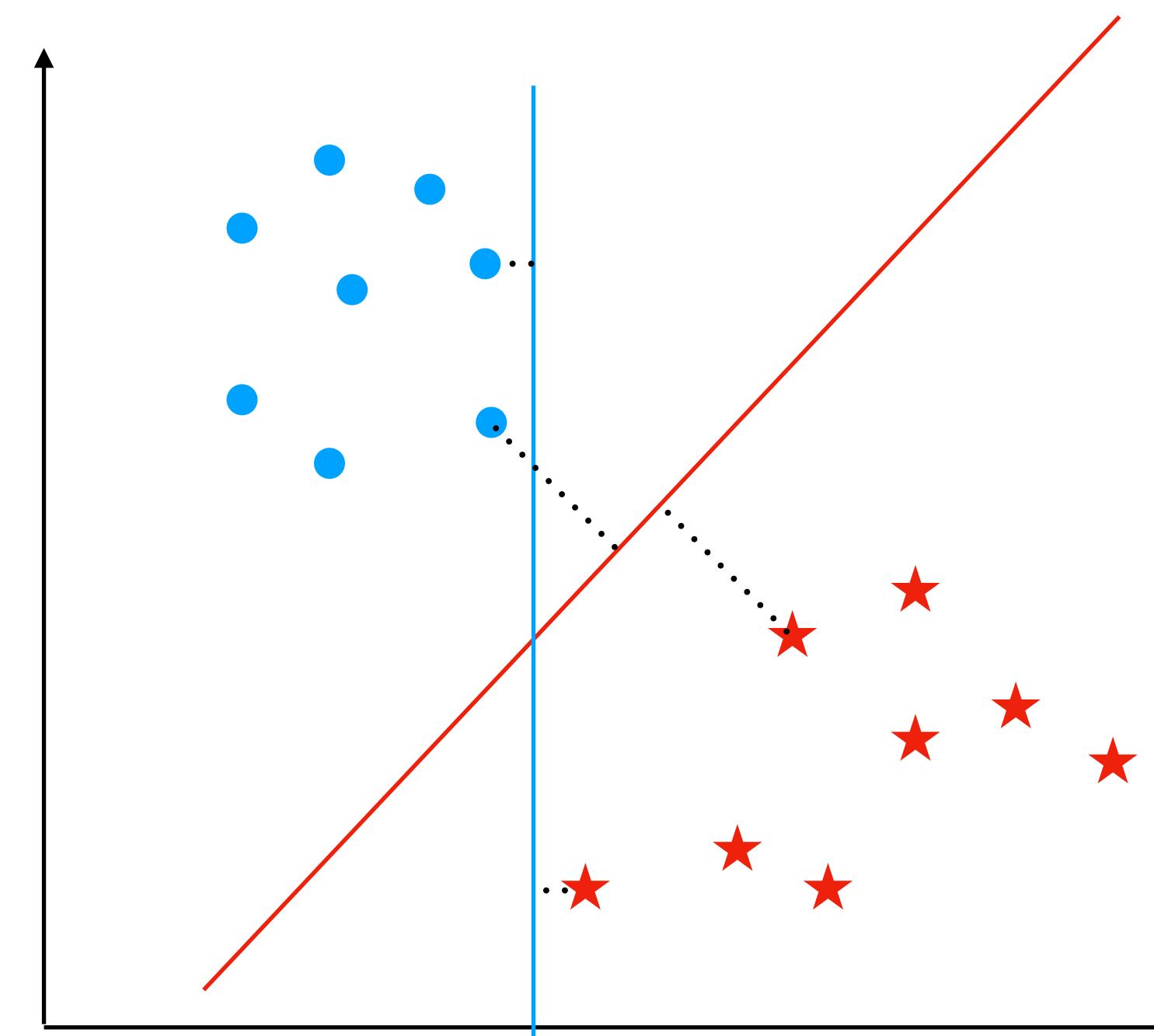
Key concept: The margin is the distance from the hyperplane to the closest point



→ Take the one with the largest margin!

Max-margin separating hyperplane

Choose the hyperplane which maximizes the margin



Why: If we slightly change the training set, the number of misclassifications will stay low

→ It will lead us to support vector machine (SVM) and logistic regression

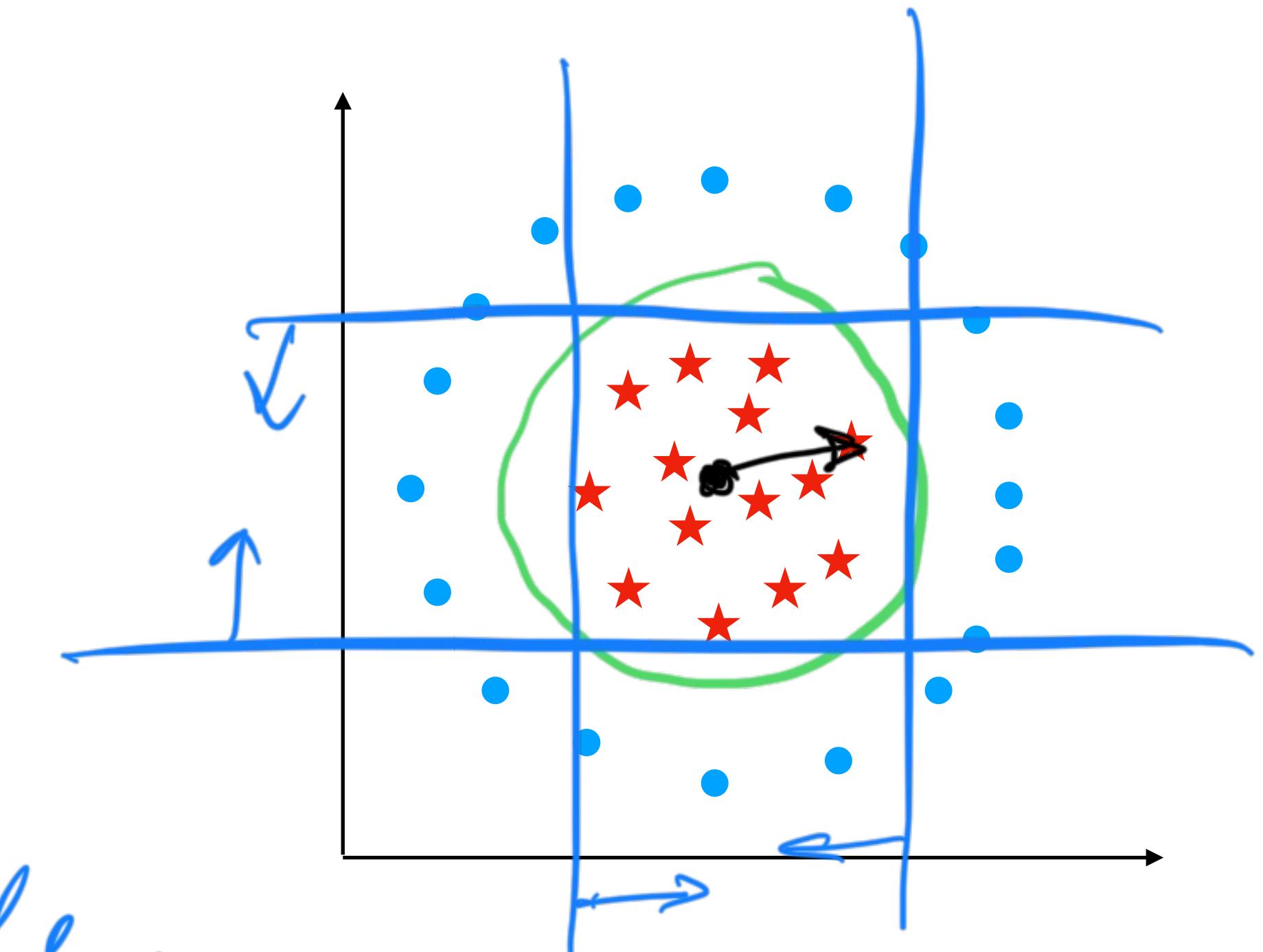
Nonlinear classifier

- Linear decision boundaries will not always work

circle around origin:

$$(x_1, x_2, x_1^2, x_2^2, 1)$$
$$w = (0, 0, 1, 1, -1)$$
$$\underline{x_1^2 + x_2^2 - 1 \geq 0}$$

radius of green circle

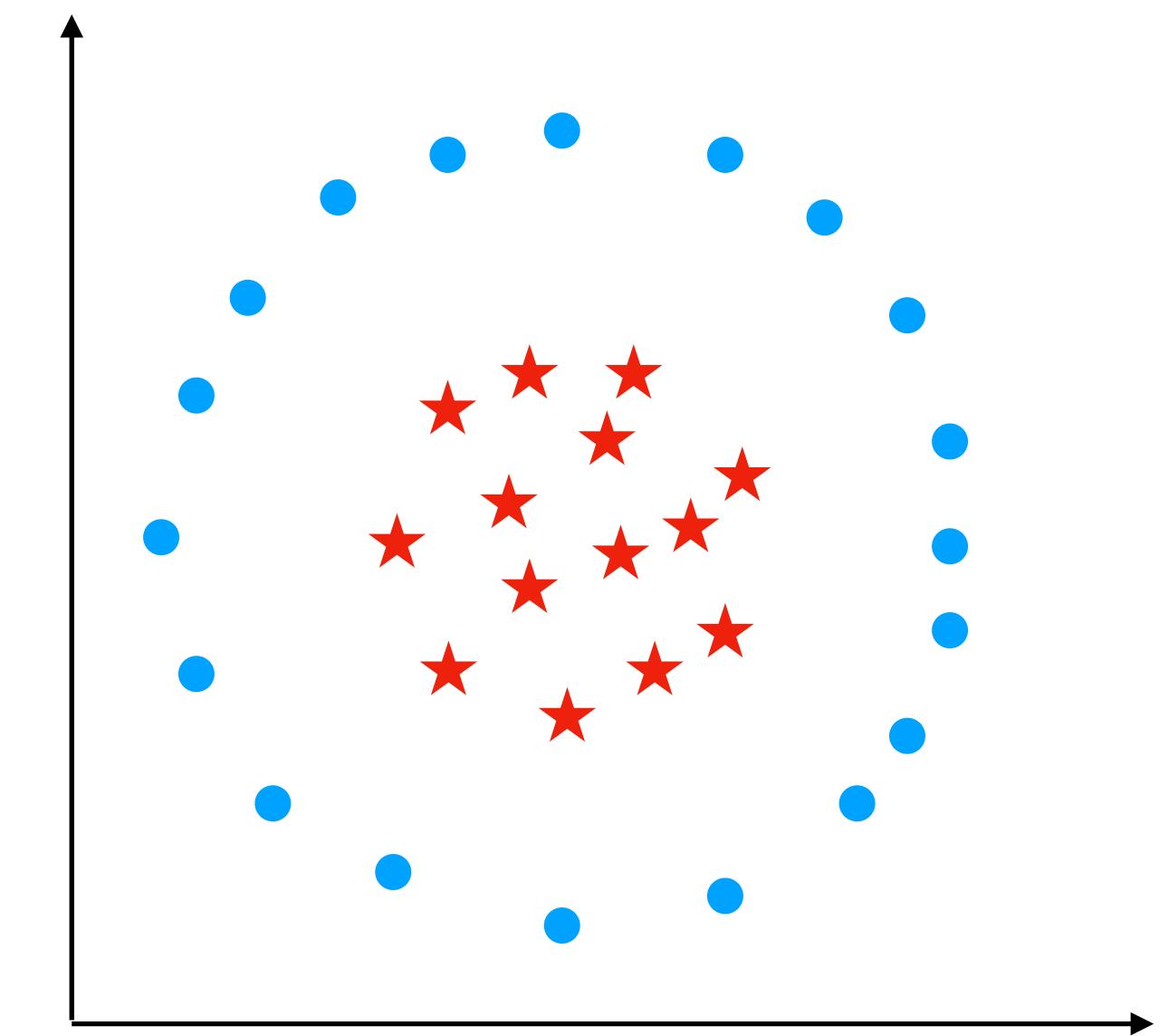


circle around (μ_1, μ_2) :

$$(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2 \geq 1$$

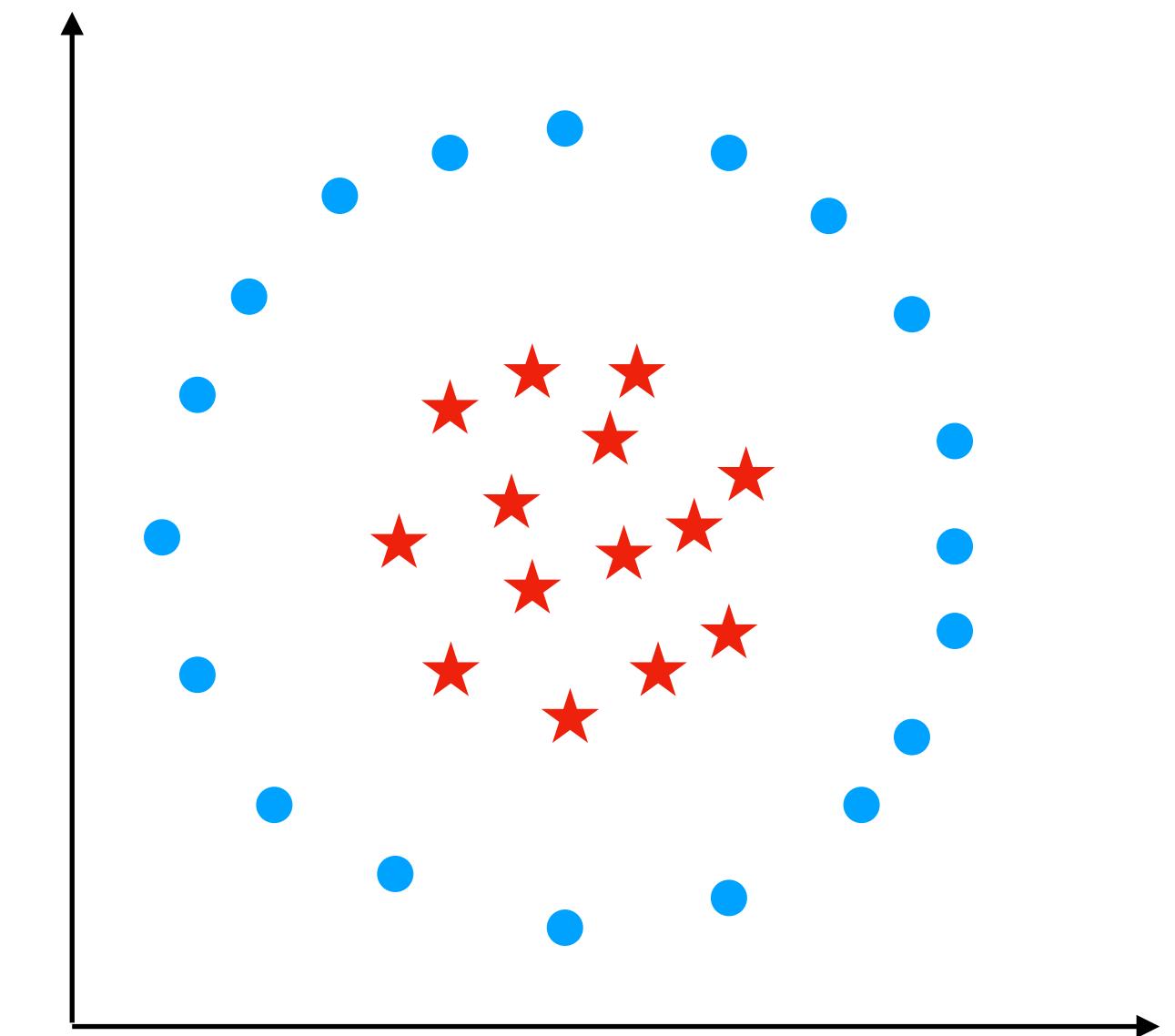
Nonlinear classifier

- Linear decision boundaries will not always work
- Feature augmentation (x, x^2, x^3, x^4)



Nonlinear classifier

- Linear decision boundaries will not always work
- Feature augmentation (x, x^2, x^3, x^4)
- Kernel Method



Formalizing Binary Classification

Setting: $(X, Y) \sim \mathcal{D}$ with ranges $\mathcal{X}, \mathcal{Y} = \{-1, 1\}$

$\{0, 1\}$

~~Reg.:~~
 ~~$l_2(y, y') = (y - y')^2$~~

Loss function: (0-1 Loss) $\ell(y, y') = 1_{y \neq y'} = \begin{cases} 1 & \text{if } y \neq y' \\ 0 & \text{if } y = y' \end{cases}$

True risk for the classification:

$$L_{\mathcal{D}}(f) = \mathbb{E}_{\mathcal{D}}[1_{Y \neq f(X)}] = \mathbb{P}_{\mathcal{D}}[Y \neq f(X)]$$

classification error

probability of making an error

$E[X] =$
 ~~$Pr(X=0) \cdot 0 + Pr(X=1) \cdot 1$~~

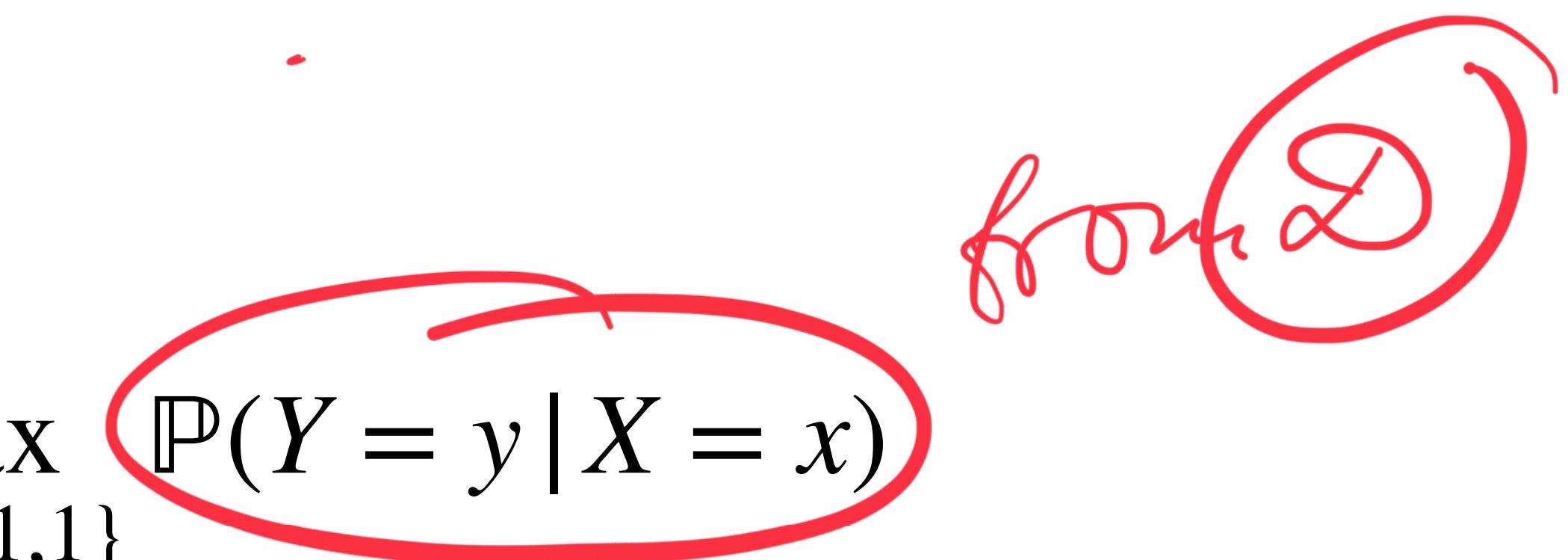
Goal: minimize $L_{\mathcal{D}}(f)$

Bayes classifier

What is the **optimal performance**, regardless of the finiteness of the training data?

Def: The classifier $f_* = \arg \min_f L_{\mathcal{D}}(f)$ is called the **Bayes classifier**

Claim:

$$f_*(x) \in \arg \max_{y \in \{-1,1\}} \mathbb{P}(Y = y | X = x)$$


Note: Bayes classifier is an *unattainable* gold standard, as we never know the underlying data distribution \mathcal{D} in practice

Proof of the Bayes classifier

Claim 1: $\forall x \in \mathcal{X}, f_*(x) \in \arg \min_{y \in \mathcal{Y}} \mathbb{P}(Y \neq y | X = x) \implies f_* \in \arg \min_{f: \mathcal{X} \rightarrow \mathcal{Y}} L_{\mathcal{D}}(f)$

$$\begin{aligned} L_{\mathcal{D}}(f) &= \mathbb{E}_{X,Y}[1_{Y \neq f(X)}] = \mathbb{E}_X[\mathbb{E}_{Y|X}[1_{Y \neq f(X)} | X]] \quad \cancel{\text{Law of total expectation}} \\ &= \mathbb{E}_X[\mathbb{P}(Y \neq f(X) | X)] \quad \cancel{\text{E}\{1\} = P} \\ &\geq \mathbb{E}_X[\min_{y \in \mathcal{Y}} \mathbb{P}(Y \neq y | X)] \\ &= \mathbb{E}_X[\mathbb{P}(Y \neq f_*(X) | X)] = \mathbb{E}_{X,Y}[1_{Y \neq f_*(X)}] = L_{\mathcal{D}}(f_*) \end{aligned}$$

Claim 2:

$$f_*(x) \in \arg \max_{y \in \mathcal{Y}} \mathbb{P}(Y = y | X = x) \iff f_*(x) \in \arg \min_{y \in \mathcal{Y}} \mathbb{P}(Y \neq y | X = x)$$

Two classes of classification algorithms

- **Non-parametric:** approximate the conditional distribution $\mathbb{P}(Y = y | X = x)$ via local averaging
 - Follow nearest neighbors' decisions (KNN)
- **Parametric:** approximate true distribution \mathcal{D} via training data
 - Minimize the empirical risk on training data (ERM)

Classification by empirical risk minimization

$$y \in \{-1, +1\}$$

How: minimize the empirical risk instead of the true risk:

$$\min_{f: \mathcal{X} \rightarrow \mathcal{Y}} L_{\text{train}}(f) := \frac{1}{N} \sum_{n=1}^N \underbrace{\mathbf{1}_{f(x_n) \neq y_n}}_{0-1 \text{ loss}} = \frac{1}{N} \sum_{n=1}^N \mathbf{1}_{y_n f(x_n) \leq 0}$$

Problem: L_{train} is not convex:

1. The set of classifiers is not convex because \mathcal{Y} is discrete
2. The indicator function $\mathbf{1}$ is not convex because it is not continuous

Convex relaxation of the classification risk

1. Instead of learning $f: \mathcal{X} \rightarrow \mathcal{Y}$, learn $g: \mathcal{X} \rightarrow \mathbb{R}$ in a convex subset of continuous functions \mathcal{G} , and predict with $f(x) = \text{sign}(g(x))$. The problem becomes

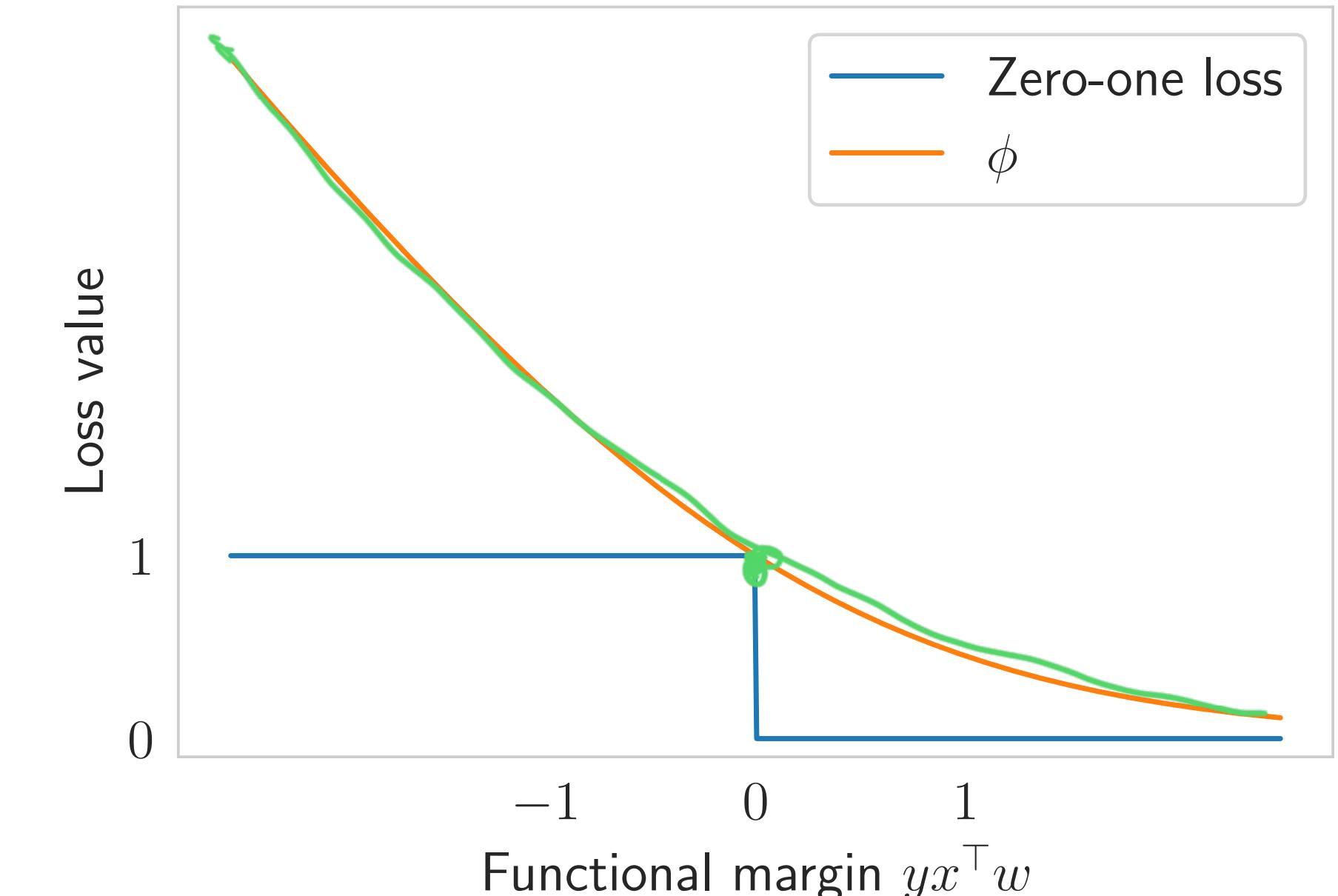
$$\min_{g \in \mathcal{G}} \frac{1}{N} \sum_{n=1}^N \mathbf{1}_{y_n g(x_n) \leq 0}$$

2. Replace the indicator function by a convex $\phi: \mathbb{R} \rightarrow \mathbb{R}$ and minimize

$$\min_{g \in \mathcal{G}} \frac{1}{N} \sum_{n=1}^N \phi(y_n g(x_n))$$

ϕ is a function of the functional margin $y_n g(x_n)$

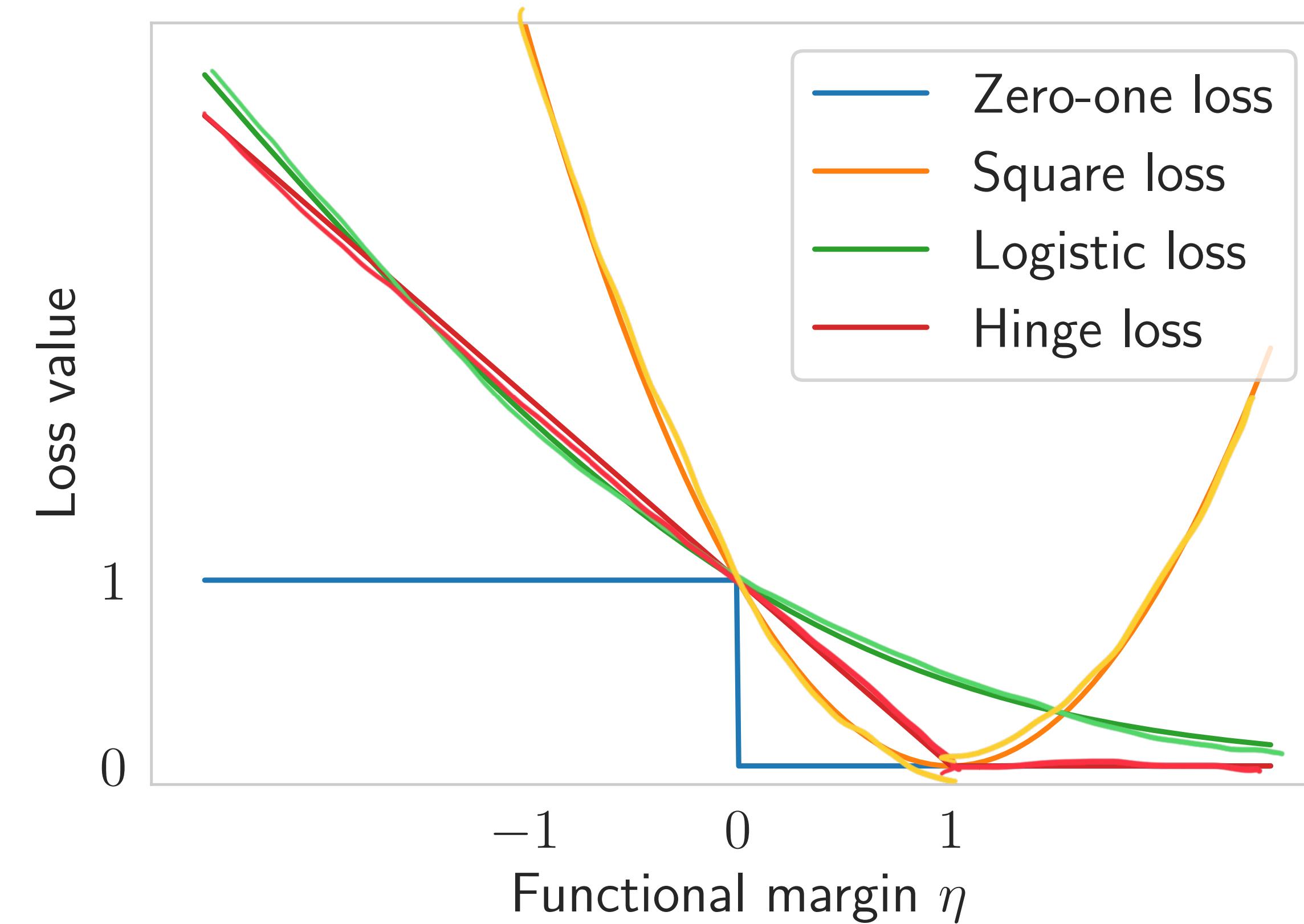
→ This is a convex problem!



Remark: possible to bound the zero-one risk $L(f)$ by the ϕ risk *

* Under technical assumptions on the function ϕ

Losses for Classification



Logistic loss → logistic regression

Hinge loss → max margin classification

Do we still have time?

No.

Bonus: a good regressor implies a good classifier

Consider $\mathcal{Y} = \{0,1\}$, for all regression functions $\eta : \mathcal{X} \rightarrow \mathbb{R}$ we can define a classifier as

$$\begin{aligned} & \mathcal{X} \rightarrow \{0,1\} \\ f_\eta : \quad & x \mapsto 1_{\eta(x) \geq 1/2} \end{aligned}$$

Claim:

$$L_{\mathcal{D}}^{\text{classif}}(f_\eta) - L_{\mathcal{D}}^{\text{classif}}(f^*) \leq 2\sqrt{L_{\mathcal{D}}^{\ell_2}(\eta) - L_{\mathcal{D}}^{\ell_2}(\eta^*)}$$

Where $L_{\mathcal{D}}^{\text{classif}}(f_\eta) = \mathbb{E}_{\mathcal{D}}[1_{f(X) \neq Y}]$, $L_{\mathcal{D}}^{\ell_2}(f) = \mathbb{E}_{\mathcal{D}}[(Y - f(X))^2]$ and $\eta_* = \arg \min_{\eta} L_{\mathcal{D}}^{\ell_2}(\eta)$

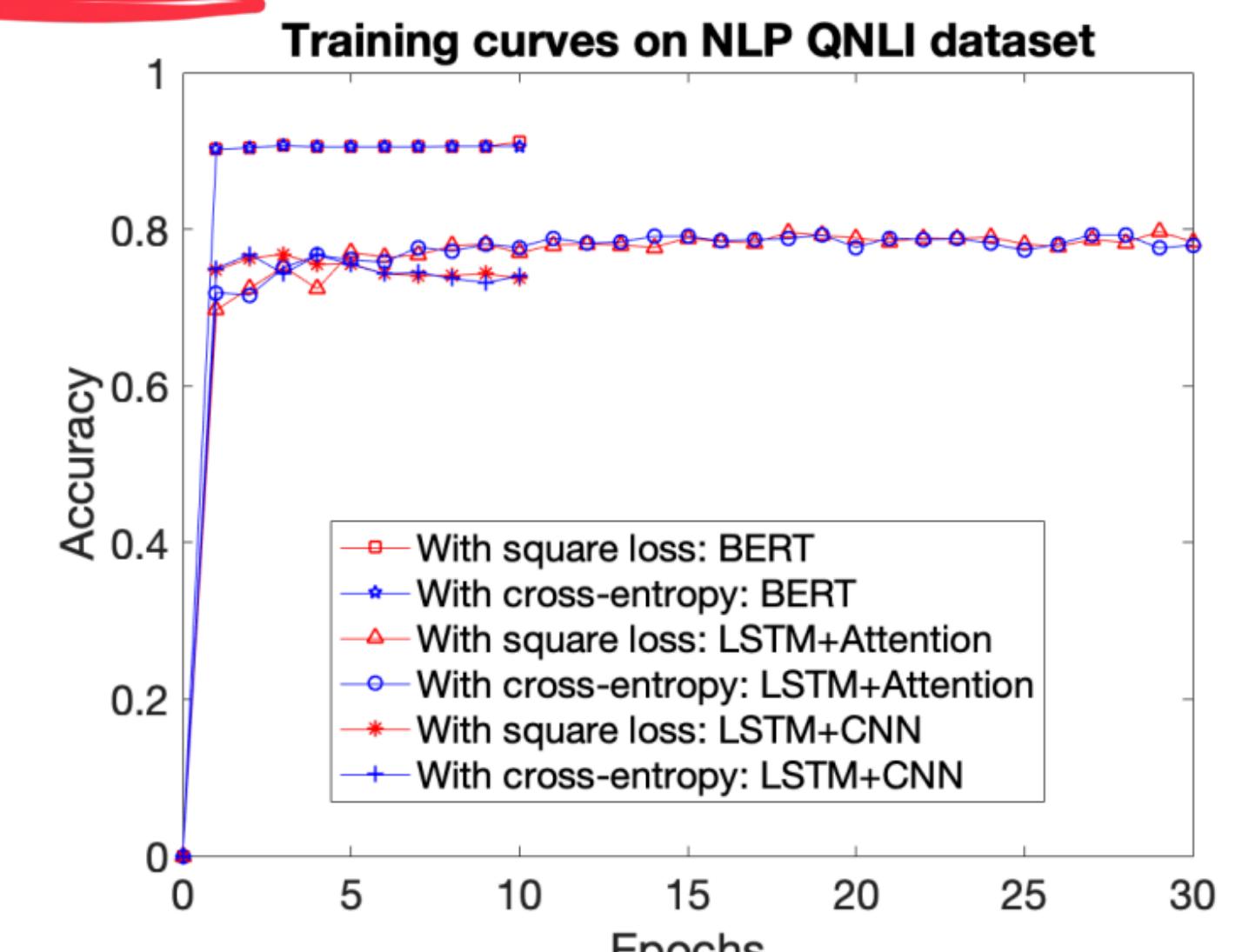
► If η is good for regression then f_η is good for classification too (converse is not true)

Bonus: does the loss function matter? ~~(Over-parameterization regime)~~

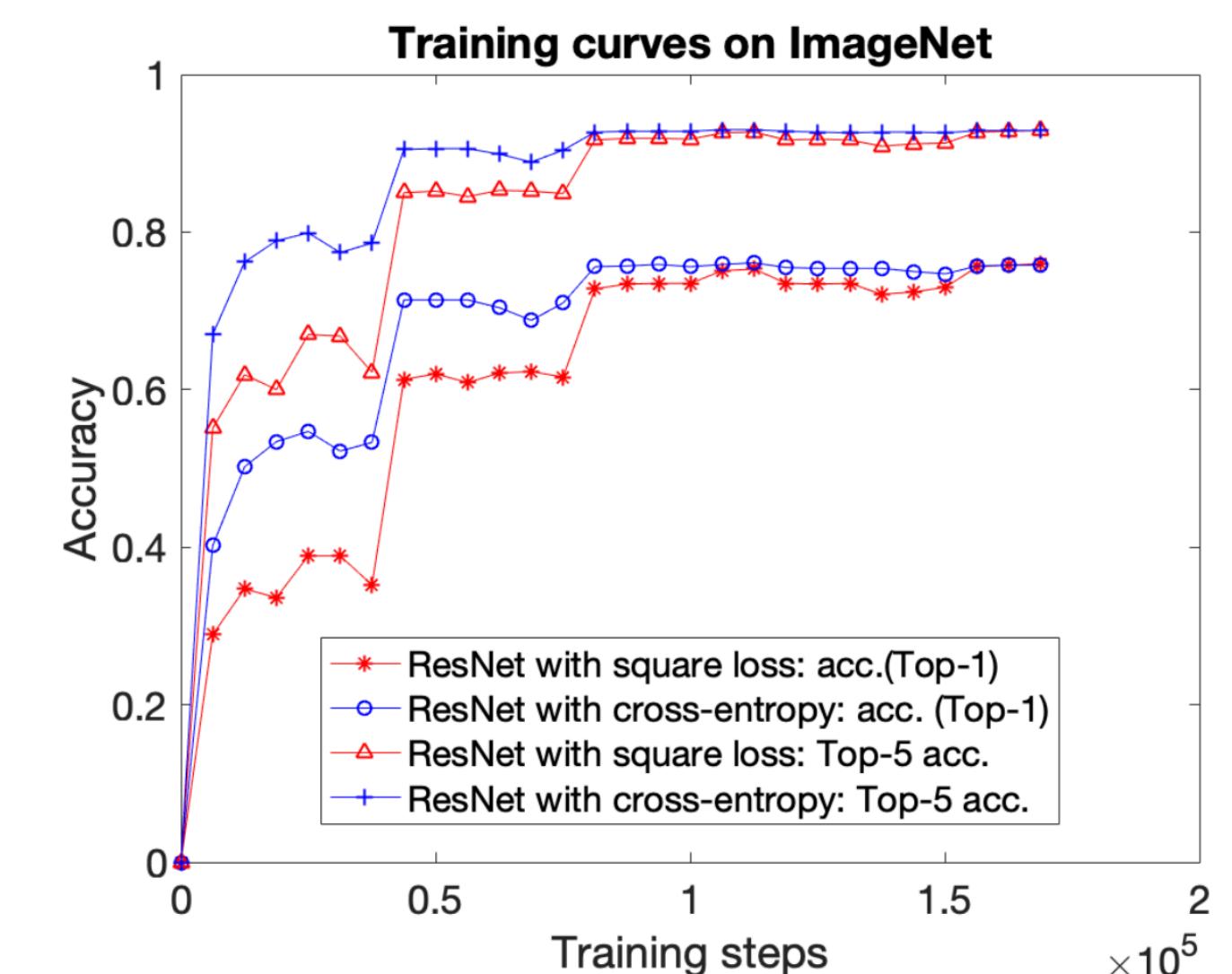
Assume sufficient over-parameterization ($n \ll d$), i.e. all training points are *equally close* to the separating hyperplane, with high probability

Optimization (training):

the outcome of optimization with *gradient descent*, is the same for both logistic loss and square loss



(a) NLP tasks



(c) Vision tasks

Recap

- Classification:
 - Mapping inputs to discrete outputs (categorical)
 - **Not** a special form of regression!
- Ways to perform classification:
 - Non-parametric: K-Nearest-Neighbors
 - Parametric: learning X-to-Y mapping via ERM
 - More complex decision boundaries? Non-linear classifiers
- Classification vs. Regression:
 - Classical regime: classification tasks cannot be well-solved using regression methods
 - Over-parameterization regime: solutions trained from both regression and classification methods are equivalent