University of Zurich[UZH]

Institute of Computational Linguistics

Machine Translation

6 Linear Models

Mathias Müller

Last time

Matrix-vector multiplication: right

$m_1 = [1 \;\; 2 \;\; 3]$

$$M = \begin{array}{c} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \end{bmatrix} \end{array} \begin{array}{l} m_1 \\ m_2 \end{array} \qquad \vec{a} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

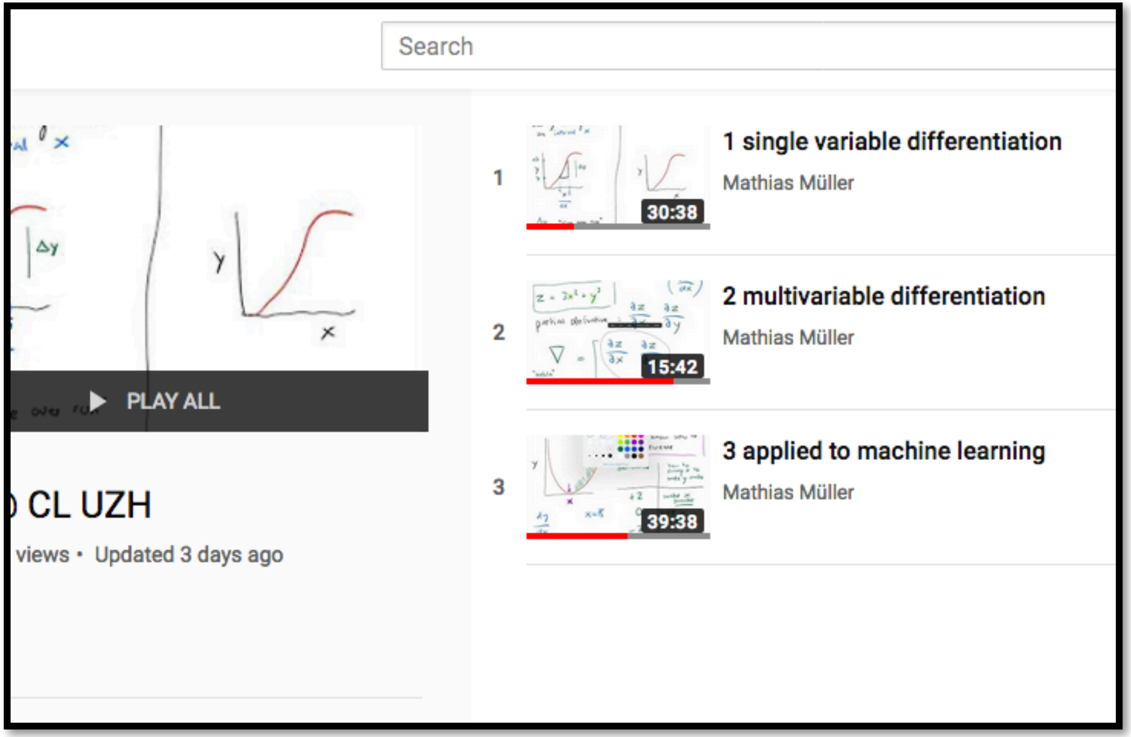
2×3 $\qquad$ 3

$$M\vec{a} = \begin{bmatrix} (m_1)^T \cdot \vec{a} \\ (m_2)^T \cdot \vec{a} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\ \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \end{bmatrix}$$

$$\begin{bmatrix} \underbrace{1 \times 1}_{1} + \underbrace{2 \times 2}_{4} + \underbrace{3 \times 3}_{9} \\ \underbrace{0 \times 1}_{0} + \underbrace{(-1)2}_{-2} + \underbrace{1 \times 3}_{3} \end{bmatrix}$$

$$\left\{ \begin{array}{l} \cancel{1} \\ 1 \end{array} \right. \quad \begin{bmatrix} 14 \\ 1 \end{bmatrix}$$

Started my Youtube career



Topics of today

- learn about a class of machine learning algorithms: linear models
- specific instances of linear models:
  - linear regression
  - logistic regression

Why those topics

- NMT systems are built with neural networks
- neural networks are **logistic regression** with a twist

nn = several nested logistic regressions

- **logistic regression** is **linear regression** with a twist

adding a non-linear function

How we represent data for ML problems

"good" 1

"bad" 0

$$X \qquad\qquad\qquad y$$

$$\begin{array}{cc} \text{sugar} & \text{vitamin C} \\ \downarrow & \downarrow \end{array}$$

$$\left.\begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 6 & 7 \end{bmatrix}\right\} \text{3 observations/samples} \qquad \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Regression vs. classification problems

linear regression

logistic regression

sugar	vitamin C	beverage
67.0	0.01	“Coke”
0.2	0.00	“Water”
4.0	0.98	“Milk”

train

sugar	vitamin C	beverage
55.0	0.04	?

test set

sugar	vitamin C	heart failures / year
67.0	0.01	1234
0.2	0.00	1
4.0	0.98	3

sugar	vitamin C	heart failures / year
67.0	0.01	?

Linear Regression

Regression Problems

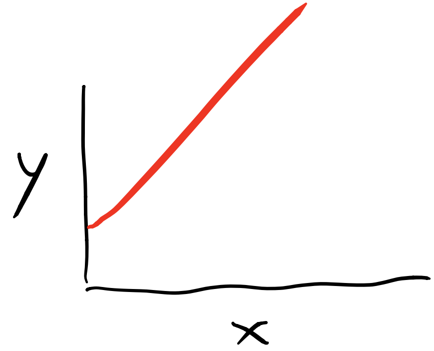
- Assumption: data-generating process is a function
- fitting a regression model: approximating this unknown function

- fitting a regression model: 1) decide on a class of functions, 2) set all parameters that fully describe the function

Classes of functions

linear functions

$$y = 2x + 1$$



polynomial functions

$$y = 3x^2 + 4x + 3$$

exponential functions

$$y = e^x$$

Parameters that describe functions

intercept

(bias)

$$y = -2x + 3z + 4$$

coefficients

(weights)

Linear Regression

3D

plane

2D

>4D

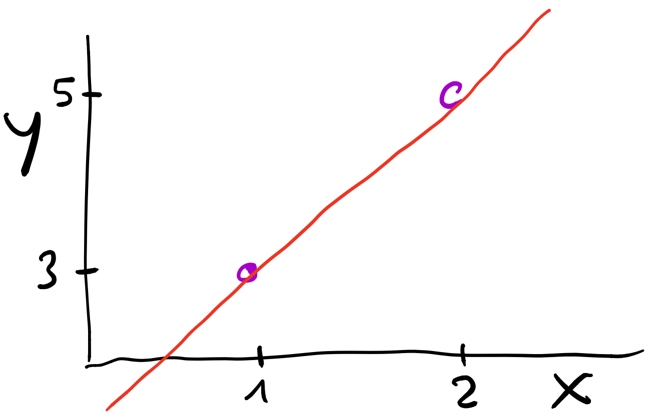
- function class: **linear**
- linear functions describe **lines or hyperplanes**
- parameters to be learned: 1 weight for each feature in X, optionally 1 **intercept**

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{array}{l} \text{sugar} \\ \text{vitamin C} \end{array} \qquad y = c_1 * x_1 + c_2 * x_2 + \bar{c}$$

Line or Hyperplane?

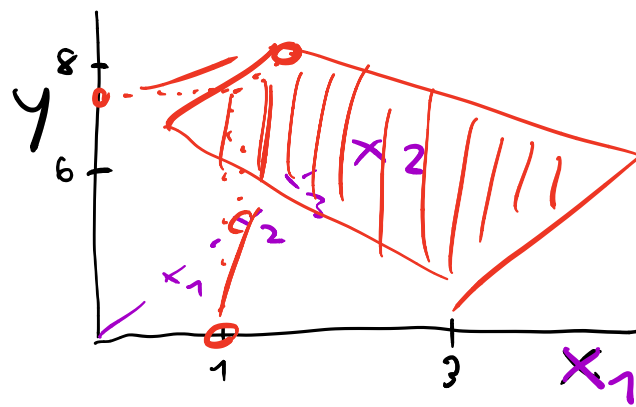
x	y
1	3
2	5

$$y = 2x + 1$$



x	y
$\begin{bmatrix}1\\2\end{bmatrix}\begin{matrix}x_1\\x_2\end{matrix}$	7
$\begin{bmatrix}3\\2\end{bmatrix}\begin{matrix}x_1\\x_2\end{matrix}$	13

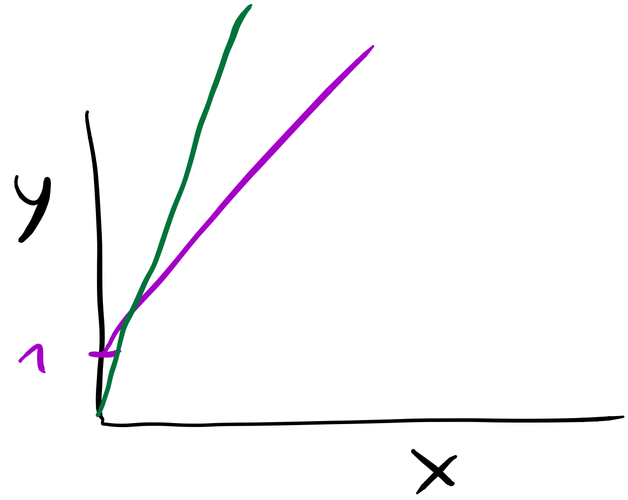
$$y = 3x_1 + 2x_2$$



Equation of a line

$$y = mx + b$$

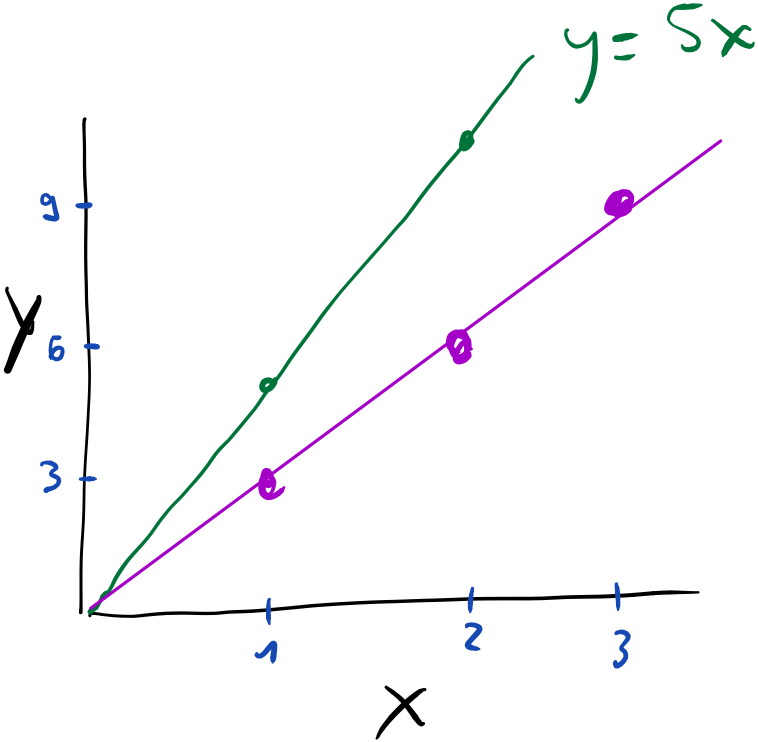
$$y = m_1 x_1 + m_2 x_2 + b$$



$$y = 2x + 1$$

$$y = 17x$$

Simple linear regression problem: one feature, one target variable, no intercept



x	y
1	3.1
2	6.2
3	9.01

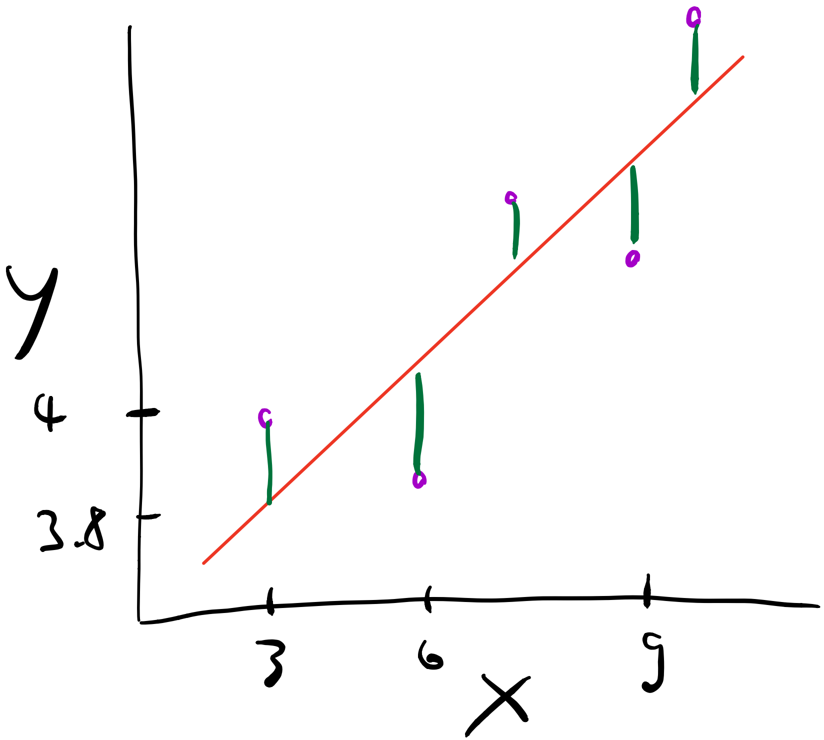
training data

$$y = \underline{5}\,x$$

$$y = \underline{3}\,x$$

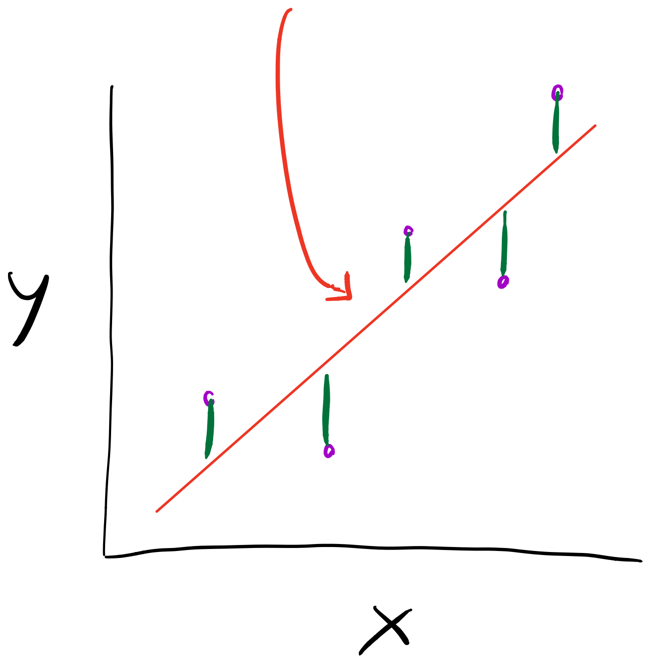
Residuals

∘ = training example



Goodness of fit: sum of squared residuals

$f(x) = 3x$



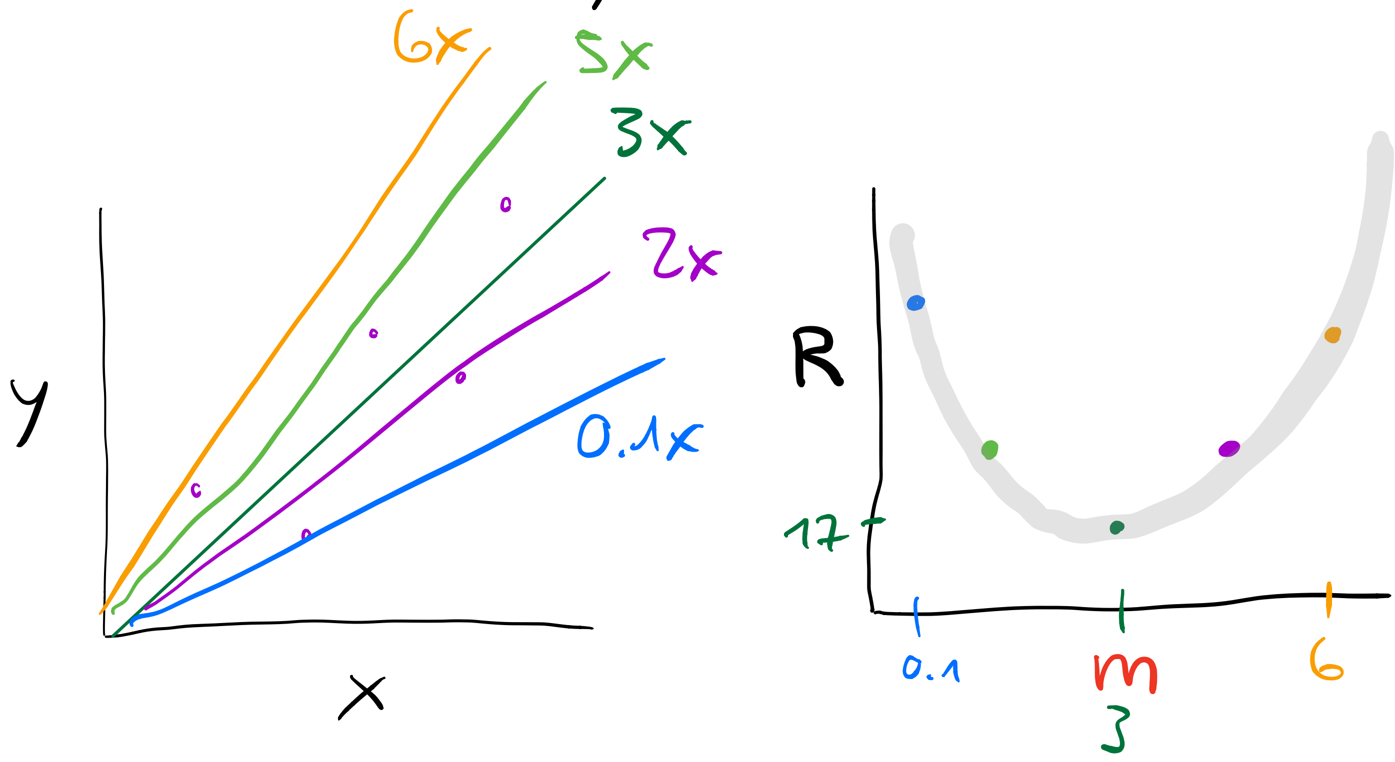
$y - f(x)$

$$R = \sum_{x,y} (y - f(x))^2$$

$$= \underline{17.}$$

How to find best line? Let's analyze sum of squared residuals

$y = mx$



Least squares solution

- closed form, analytical solution for linear regression
- solution is called **normal equations**

$R(m)$ $\quad$ $R'(m) = 0$

$m =$

$$\boxed{\theta = (X^T X)^{-1} X^T y}$$

Summary Linear Regression

- Regression approximates functions that generated the data
- functions are defined by their parameters

- linear regression approximates with linear functions
- linear functions are lines or hyperplanes

- model fitting means finding parameters that minimize sum of squared residuals, with a least squares solution

Logistic Regression

for classification.

Vector notation for linear regression

intercept/bias

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \qquad b$$

non-vector $\quad y = c_1 x_1 + c_2 x_2 + b$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ b \end{bmatrix}$$

vector + absorb bias $\quad y = \vec{x} \cdot \vec{c}$

How about linear regression for classification?

$y = \vec{c} \cdot \vec{x}$

training	data
x	y
$\begin{bmatrix} 2 \\ 3 \end{bmatrix}$	"bad"
$\begin{bmatrix} 7 \\ 1 \end{bmatrix}$	"good"

if $y < 0.5 \longrightarrow$ "bad"

if $y > 0.5 \longrightarrow$ "good"

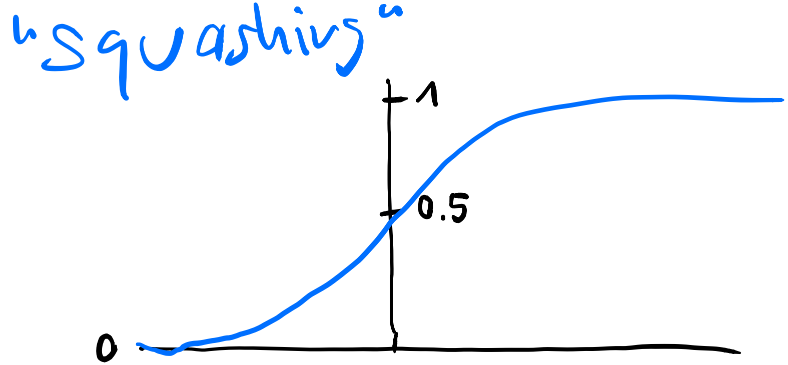
$\dfrac{-1347}{0.5}$

Logistic Regression

$$x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \qquad c = \begin{bmatrix} c_1 \\ c_2 \\ b \end{bmatrix}$$

sigmoid

$$y = \sigma\left(\vec{c} \cdot \vec{x}\right)$$



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

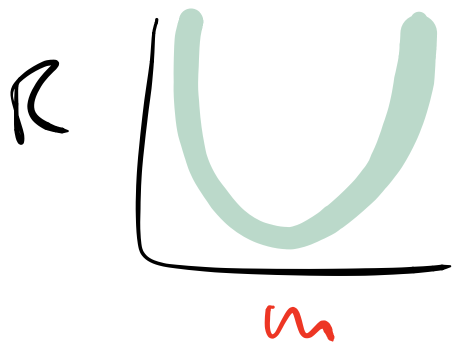
<u>Interpretation:</u> y is <u>the probability</u> of <u>the positive class</u>

"good" + 0.79

0.21

Optimizing Logistic Regression

- Logistic regression does not have a closed form solution
- but it is a **convex optimization** problem



Summary

- **linear models** are algorithms that apply only linear transformations to input features
- **linear regression** solves a regression problem in closed form
- **logistic regression** solves a classification problem with convex optimization

Next time

Termin	Thema
19.02.	Einführung; regelbasierte vs. datengetriebene Modelle
26.02.	Evaluation
05.03.	Trainingsdaten, Vor- und Nachverarbeitung
12.03.	N-Gramm-Sprachmodelle, statistische Maschinelle Übersetzung
19.03.	Grundlagen Lineare Algebra und Analysis, Numpy
26.03.	Lineare Modelle: lineare Regression, logistische Regression
02.04.	Neuronale Netzwerke: MLPs, Backpropagation, Gradient Descent
09.04.	Word Embeddings, Recurrent neural networks
16.04.	Tensorflow und Google Cloud Platform
30.04.	Encoder-Decoder-Modell
07.05.	Decoding-Strategien
14.05.	Attention-Mechanismus, bidirektionales Encoding, Byte Pair Encoding
21.05.	Maschinelle Übersetzung in der Praxis (Anwendungen)
28.05.	Zusammenfassung, Q&A Prüfung
Eventuell: Gastvortrag Prof. Artem Sokolov	
04.06., Raum TBA, 16:15 bis 18:00 Uhr	
Prüfung (schriftlich)	
18.06., AND-2-48, 16.15 bis 18:00 Uhr	

EVALUATION

TRAINING DATA

SMT

NMT

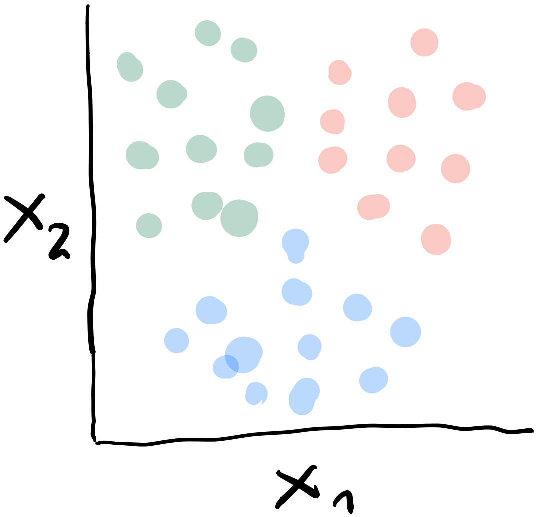
this is kinda important

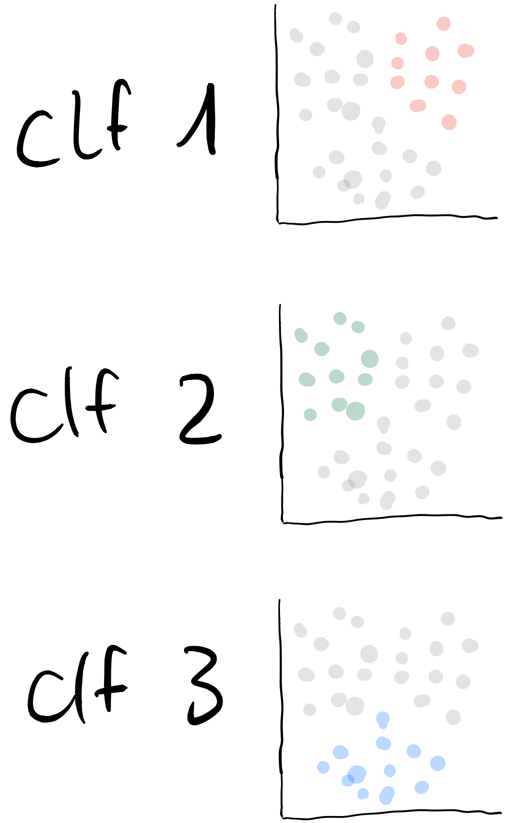
Bonus Material: Logistic Regression for multiclass problems

2 ways to extend binary logistic regression

1) One-versus-all logistic regression
2) Softmax regression

One-versus-all multi-class logistic regression





Softmax regression (= Maximum Entropy)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ 1 \end{bmatrix} \qquad C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ & & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}$$

$$y = s(Cx) \qquad \boxed{s = \frac{e^{z_j}}{\sum_{k=1} e^{z_k}}}$$

output without softmax

$$Cx = \begin{bmatrix} 117 \\ -3 \\ 0.001 \end{bmatrix}$$

output with softmax

$$s(Cx) =$$