

Stochastic Machine Learning

Chapter 01 - Introduction

Thorsten Schmidt

Abteilung für Mathematische Stochastik

www.stochastik.uni-freiburg.de
thorsten.schmidt@stochastik.uni-freiburg.de

SS 2024

Motivation

- ▶ **Machine Learning** is nowadays used at many places (Google, Amazon, etc.) with a great variety of applications.
- ▶ It is a great job opportunity ! It needs maths and probability !
- ▶ Many applications are surprisingly successful (speech / face recognition / robotic / autonomous cars / medicine / chemistry / chat GPT) and currently people are seeking further applications
- ▶ Here we want to learn about the foundations, discuss implications and what can be done by ML and what not.

Topics include:

- ▶ Foundations and deeper understanding
- ▶ Uncertainty Quantification
- ▶ Optimization under Uncertainty
- ▶ Regulation and Fairness
- ▶ Risk quantification (for example in transfer learning)
- ▶ Advancement of fundamental theories (LEAN)

Organization

Slides and code will be available on github. But - proofs and theorems will mostly remain on blackboard only.

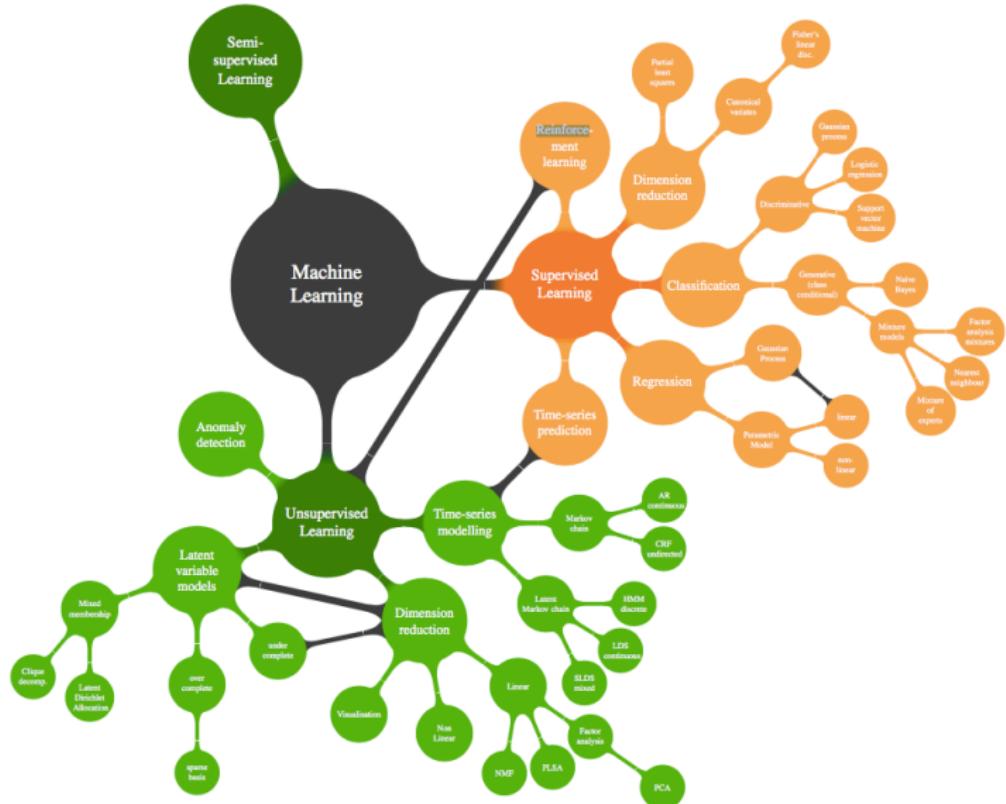
- ▶ The lectures will mix python implementation with theory - so it is now a good time for you to start learning python. Every 4th lecture will focus on implementation and projects.
- ▶ Homework in the 2nd part will be done by projects. You can choose a topic which interests you, and we will provide topics. Groups up to 6 people work on a project, more than one group can also work on the same project.
- ▶ We provide a shared repository for all projects, such that you share your current work and can profit from the work from others.
- ▶ **Moritz Ritter** is organizing the projects - please contact him for questions.
- ▶ You need 50% of the points for the oral exam at the end. The oral exam will cover all **theoretical** parts of the lecture.

Overview¹

- ▶ Artificial intelligence is the field where computers solve problems.
- ▶ It is easy for a computer to solve tasks which can be described formally (Chess, Tic-Tac-Toe). The challenge is to solve tasks which are hard to describe formally (but are easy for humans: walk, drive a car, speak, recognize people ...)
- ▶ The solution is to allow computers to learn from experience and to understand the world by a hierarchy of concepts, each concept defined in terms of its relation to simpler concepts.
- ▶ A fixed knowledge-base would be somehow limiting such that we are interested in such attempts where the systems acquire their own knowledge, which we call **Machine Learning**.

¹This introduction follows somehow Goodfellow et.al. (2016) and my previous lectures.

- ▶ First examples of machine learning are **logistic regression** or **naive Bayes** → standard statistical procedures (E.g. the recognition of spam, more examples to follow)
- ▶ Problems become simpler with a nice representation. Of course it would be nice if the system itself could find such a representation, which we call **representation learning**.
- ▶ An example is the so-called **auto-encoder**. This is a combination of an encoder and a decoder. The encoder converts the input to a certain representation and the decoder converts it back again, such that the result has nice properties.
- ▶ Speech for example might be influenced by many factors of variation (age, sex, origin, ...) and it needs nearly human understanding to disentangle the variation from the content we are interested in.
- ▶ **Deep Learning** solves this problem by introducing hierarchical representations.
- ▶ This leads to the following hierarchy:
- ▶ AI → machine learning → representation learning → deep learning.



Source: Barber (2012).

Examples of Machine Learning

Some of the most prominent examples:

- ▶ LeCun et.al.² recognition of handwritten digits. The MNIST Database³ provides 60.000 samples for testing algorithms. The NIST database is of increased size⁴
- ▶ The Viola & Jones face recognition,⁵. This path-breaking work proposed a procedure to combine existing tools with machine-learning algorithms. One key is the use of approx. 5000 learning pictures to train the routine. We will revisit this procedure shortly.
- ▶ Imagenet is an image database containing many images classified (cats, cars, etc.)⁶
- ▶ Various twitter datasets are available, for example for learning to detect hate speech.
- ▶ Kaggle⁷ is a platform where computational competitions are hosted. It also provides many many data examples with it.
- ▶ Datasets for machine-learning research on Wikipedia⁸.

²Y. LeCun et al. (1998). „Gradient-based learning applied to document recognition“. In: Proceedings of the IEEE 86.11, pp. 2278–2324.

³<http://yann.lecun.com/exdb/mnist/>

⁴<https://www.nist.gov/srd/nist-special-database-19>

⁵P. Viola and M. Jones (2001). „Robust Real-time Object Detection“. In: International Journal of Computer Vision. Vol. 4. 34–47.

⁶<http://image-net.org>

⁷www.kaggle.com

⁸https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research

MNIST

We will see this data set in more detail.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

As an extension we also have the EMNIST data set - with letters and more challenging.

I UXEB1W Rk99aD HgP AV BORH fBh
15 HGh b1CD84h eF V UP Q9 b345Qm
hae56aKGh1P5/9m4u16H8/ hem
nwvD0nt25XB60V20ueHmt+nCA
J3EEZWSfWw2b4J TjB PzD a2V0B
VfQJdEfab38GJT SJ XRYM qby7d
06dPzA50Af4/1b507wa0mTBNBP
+9hAKZ/EE n1.MKFOBHXVaV fGWA
h3#NqRf4WCZF dAW465Nq844GA
4?REGt+LcYKPCgFUR98HZ60TSNx
n2HREB9R5QfY9BPJNM1zPrSS
41J0ZGRZY28GD46SB38A+t147
WFHB40U1EYC9HOUND1Y4N23H2
h6bPN9453n92E4W5AdYB4Pa14
9FQ9Cn3C6dNs6TTe16DOP3F35

- ▶ Speech recognition has long been a difficult problem for computers (first works date to the 50's) and only recently been solved with high computer power. It may seem surprising, that mathematical tools are at the core of these solutions. Let us quote Hinton et.al.⁹

Most current speech recognition systems use hidden Markov models (HMMs) to deal with the temporal variability of speech and Gaussian mixture models (GMMs) to determine how well each state of each HMM fits a frame or a short window of frames of coefficients that represents the acoustic input. (...)

Deep neural networks (DNNs) that have many hidden layers and are trained using new methods have been shown to outperform GMMs on a variety of speech recognition benchmarks, sometimes by a large margin

So, one of our tasks will be to develop a little bit of mathematical tools which we will need later. Most notably, some of the mathematical parts can be replaced by deep learning, which will be of high interest to us.

⁹Geoffrey Hinton et al. (2012). „Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups“. In: IEEE Signal Processing Magazine 29.6, pp. 82–97.

nature

View all journals Search Log in

Explore content About the journal Publish with us Subscribe Sign up for alerts RSS feed

[nature](#) > [news](#) > [article](#)

NEWS | 15 April 2024

AI now beats humans at basic tasks – new benchmarks are needed, says major report

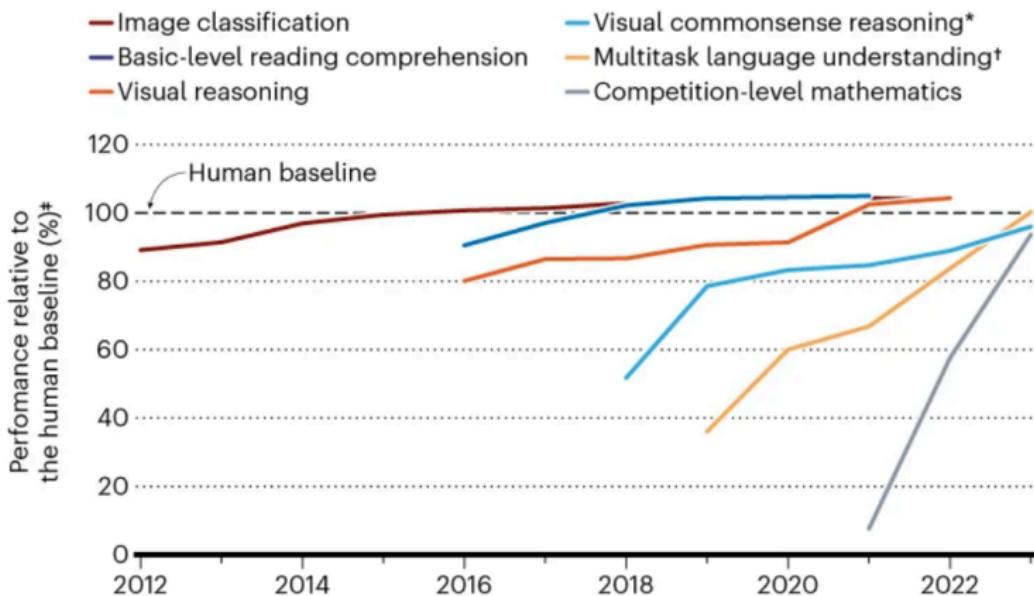
Stanford University's 2024 AI Index charts the meteoric rise of artificial-intelligence tools.

By [Nicola Jones](#)



SPEEDY ADVANCES

In the past several years, some AI systems have surpassed human performance on certain benchmark tests, and others have made rapid progress.



*Requires an AI system to answer questions about an image and provide a rationale for why its answers are true.

†Tests an AI model's knowledge and problem-solving ability with regard to 57 subjects, including broader topics such as mathematics and history, and narrower areas such as law and ethics.

[#]Data indicate the best performance of an AI model that year.

©nature

Cost of business

As performance is skyrocketing, so are costs. GPT-4 – the LLM that powers ChatGPT and that was released in March 2023 by San Francisco-based firm OpenAI – reportedly cost US\$78 million to train. Google's chatbot Gemini Ultra, launched in December, cost \$191 million. Many people are concerned about the energy use of these systems, as well as the amount of water needed to cool the data centres that help to run them². "These systems are impressive, but they're also very inefficient," Maslej says.

- ▶ What kind of consequences do we draw from this ?

The AI Index Report 2024

1. AI beats humans on some tasks, but not on all.

AI has surpassed human performance on several benchmarks, including some in image classification, visual reasoning, and English understanding. Yet it trails behind on more complex tasks like competition-level mathematics, visual commonsense reasoning and planning.

2. Industry continues to dominate frontier AI research.

In 2023, industry produced 51 notable machine learning models, while academia contributed only 15. There were also 21 notable models resulting from industry-academia collaborations in 2023, a new high.

3. Frontier models get way more expensive.

According to AI Index estimates, the training costs of state-of-the-art AI models have reached unprecedented levels. For example, OpenAI's GPT-4 used an estimated \$78 million worth of compute to train, while Google's Gemini Ultra cost \$191 million for compute.

4. The United States leads China, the EU, and the U.K. as the leading source of top AI models.

In 2023, 61 notable AI models originated from U.S.-based institutions, far outpacing the European Union's 21 and China's 15.

5. Robust and standardized evaluations for LLM responsibility are seriously lacking.

New research from the AI Index reveals a significant lack of standardization in responsible AI reporting. Leading developers, including OpenAI, Google, and Anthropic, primarily test their models against different responsible AI benchmarks. This practice complicates efforts to systematically compare the risks and limitations of top AI models.

6. Generative AI investment skyrockets.

Despite a decline in overall AI private investment last year, funding for generative AI surged, nearly octupling from 2022 to reach \$25.2 billion. Major players in the generative AI space, including OpenAI, Anthropic, Hugging Face, and Inflection, reported substantial fundraising rounds.

7. The data is in: AI makes workers more productive and leads to higher quality work.

In 2023, several studies assessed AI's impact on labor, suggesting that AI enables workers to complete tasks more quickly and to improve the quality of their output. These studies also demonstrated AI's potential to bridge the skill gap between low- and high-skilled

8. Scientific progress accelerates even further, thanks to AI.

In 2022, AI began to advance scientific discovery. 2023, however, saw the launch of even more significant science-related AI applications—from AlphaDev, which makes algorithmic sorting more efficient, to GnoME, which facilitates the process of materials discovery.

9. The number of AI regulations in the United States sharply increases.

The number of AI-related regulations in the U.S. has risen significantly in the past year and over the last five years. In 2023, there were 25 AI-related regulations, up from just one in 2016. Last year alone, the total number of AI-related regulations grew by 56.3%.

Questions

We repeatedly state questions after some slides. These allow you to reflect on the content and also invite you to research / experiment with some topics yourself.

- ▶ Was is artificial intelligence ?
- ▶ Was is machine learning ?
- ▶ Do you know what a neural network is (look for the history in the internet)?
- ▶ What are shallow / deep networks ?
- ▶ What are the applications which you find most exciting ?
- ▶ What are the applications that you think will have the largest impact on our future?
- ▶ Research a bit yourself: look for datasets, look for latest applications etc.

1. Introduction → Machine learning basics

Types of machine learning:

- ▶ **Supervised learning:** The data consists of datapoints and associated labels, i.e. we start from the dataset

$$(x_i, y_i)_{i \in I}.$$

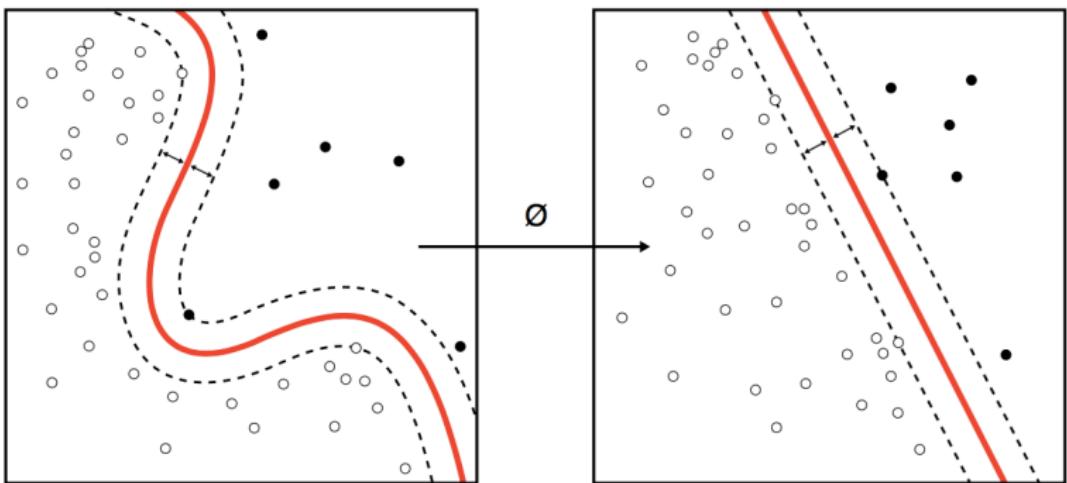
We give some examples:

- ▶ **Image recognition** (face recognition) where the images come with labels, i.e. cats / dogs or the person to which the image is associated to.
- ▶ **Spam filter** the training set contains emails together with the label spam / no spam.
- ▶ **Speech recognition** here sample speech files comes together with the content of the sentences. It is clear, that some sort of grammar understanding helps to break up the sentences into smaller pieces, i.e. words.
- ▶ **Ratings** here, to a creditor we assign the credit quality (AAA, ...) A typical finance application.
- ▶ **Language Models** here the idea is to predict the next letter and train it on a large dataset.

- **Unsupervised learning:** In this case the data just comes at it is, i.e.

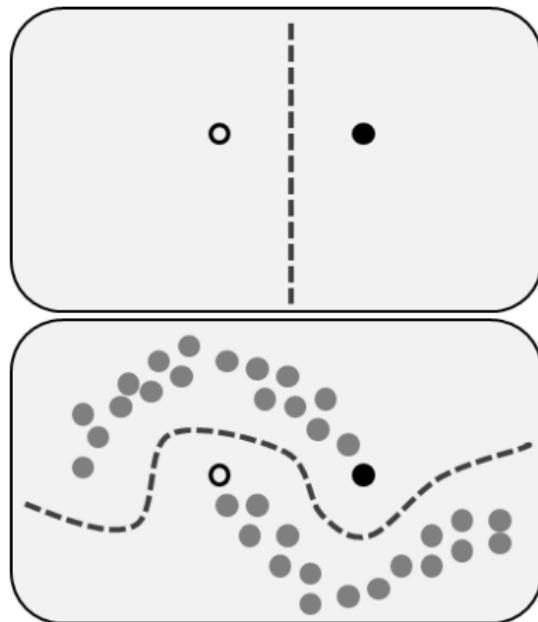
$$(x_i)_{i \in I}$$

and one goal would be to identify a certain structure from the data itself. In this sense the machine learning algorithm shall itself find a characteristics which divides the data into suitable subsets.



Picture by: Alisneaky, svg version by User:Zirguezi - Own work,
CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=47868867>

- ▶ **Semi-supervised learning:** only a few data are labelled and many are unlabelled.
- ▶ Labelling typically is quite expensive and the additional use of unlabelled data might improve the performance. However, some assumptions need to be made, such that this procedure works through.



Picture by: Techerin - Own work,
CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=19514958>

Questions

- ▶ What is (semi-/un-/supervised) learning ?
- ▶ Give examples
- ▶ The examples you have been choosing in the first part, please classify them

Dynamic contexts

- ▶ It is apparent, that the above questions have been static
- ▶ Many applications are dynamic !
- ▶ To drive a car
- ▶ To manage a portfolio
- ▶ To predict future evolutions from a time-series

This will require different methods which we will meet in the course. We will use Markov processes when we study **Reinforcement learning** in greater detail.

A quite related area is **Statistical Learning**. This new area of statistics is quite related to machine learning and we will study a number of relevant problems¹⁰.

- ▶ Formally, we have an observation given by pairs (x_i, y_i) , $i \in I$ and randomness is modelled with an (unknown) probability distribution
- ▶ The task is to predict y based on x .
- ▶ From all functions f in some set \mathcal{H} we want to choose f so that the *expected risk*

$$E[L(f(X), Y)]$$

is minimal. Here, L is some chosen loss function.

- ▶ because the probability is unknown, one estimates the expected risk with the *empirical risk*

$$\frac{1}{n} \sum_{i=1}^n L(f(x_i), y_i).$$

Popular and well-known examples are

- ▶ **Regression** in the simple least-squares regression, $f(x) = m + nx$ and $L(\cdot) = \cdot^2$
- ▶ **Classification** also falls into this framework: here Y takes only finitely many values, like $\{A, B, C, \dots\}$ and possibly a step-function is chosen as loss function.

¹⁰There is a lot of interesting literature in this area: e.g. T. Hastie, R. Tibshirani, and J. Friedman (2009). **The Elements of Statistical Learning**. Springer Series in Statistics. Springer New York Inc. URL: <https://statweb.stanford.edu/~tibs/ElemStatLearn/>, Vladimir Vapnik (2013). **The nature of statistical learning theory**. Springer science & business media.

Definition

A computer program learns from experience E with respect to tasks T , if its performance P improves with experience E .

This quite vague definition allows us to develop some intuition about the situation.

- ▶ **Experience** is given by an increasing sequence of observations, for example X_1, X_2, \dots, X_t could represent the information at time t . This is typically decoded in a **filtration**: a filtration is an increasing sequence of sub- σ -fields $(\mathcal{F}_t)_{t \in T}$.
- ▶ The performance is often measured in terms of an **utility function**. For example the utility at time t could be given by $U(X_t)$ with a function U . U could of course depend on more variables. One could also look for the accumulated utility

$$\sum_{t=1}^T U(X_t).$$

One very simple learning algorithm is linear regression, a classical statistical concept. Here it arises as an example of **supervised learning**.

Example (Linear Regression)

Suppose we observe pairs $(x_i, y_i)_{i=1, \dots, n}$ and want to predict y on basis of x . Linear regression requires

$$\hat{y}(x) = \beta x$$

with some weight $\beta \in \mathbb{R}$. We specify a loss function¹¹

$$\text{RSS}(\beta) := \sum_{i=1}^n (y_i - \hat{y}(x_i))^2$$

and minimize over β .

One could choose $-\text{MSE}$ as utility function. So how does the system **learn**?

¹¹ Given by the Residual Sum of Squares here.

The system learns by maximizing the utility, i.e. minimizing the MSE for each n . And additional data will lead to a better prediction. We will later see that this is in a certain sense indeed optimal.

We use the **first-order condition** to derive the solution letting $\mathbf{x} = (x_1, \dots, x_n)$ and similar for \mathbf{y} ,

$$\begin{aligned} 0 &= \partial_\beta (\mathbf{y} - \beta \mathbf{x})^2 = \partial_\beta (\mathbf{y}^2 - 2\mathbf{y}^\top \beta \mathbf{x} + \beta^2 \mathbf{x}^\top \mathbf{x}) \\ \Leftrightarrow 0 &= -2\mathbf{x}^\top \mathbf{y} + 2\beta \mathbf{x}^\top \mathbf{x} \end{aligned}$$

such that we obtain

$$\hat{\beta} = (\mathbf{x}^\top \mathbf{x})^{-1} \mathbf{x}^\top \mathbf{y}.$$

Note that typically one considers affine functions of x without mentioning, i.e. one looks at functions $y = \alpha + \beta x$. This can simply be achieved with the linear approach by augmenting \mathbf{x} by an additional entry 1.

- ▶ Of course many generalizations are possible:
- ▶ To higher dimensions: consider data vectors $(\mathbf{x}_i, \mathbf{y}_i)$, $i = 1, \dots, n$,
- ▶ To nonlinear functions: include x_i^1, \dots, x_i^p into the covariates
- ▶ and many more.

Let us consider a linear regression in python (see jupyter notebook on github).

```
import yfinance as yf
import matplotlib.pyplot as plt

DAX = yf.Ticker('%5Egداخی')
DAX_History = DAX.history(start="2020-01-01", end="2020-10-26")

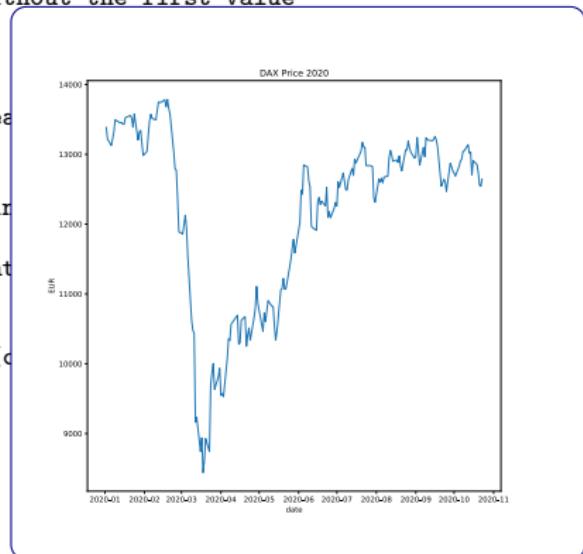
plt.figure(figsize=(10,10))
plt.plot(DAX_History.index, DAX_History['Close'])

# Linear Regression example: regress tomorrow on today
x = DAX_History['Close'][:-1]      # without the last value
y = DAX_History['Close'][1:]       # without the first value

import numpy as np
from numpy import array
from sklearn.linear_model import LinearRegression

model = LinearRegression()
x = array(x).reshape(-1,1)      # The linear model needs a 2D array
y = array(y).reshape(-1,1)
model.fit(x, y)    # values in model.inte

# Give a very sophisticated plot
import seaborn as sns; sns.set_theme(style="white")
ax = sns.regplot(x=x, y=y)
plt.show()
```



Could we improve this ? Suggestions ?

Let us consider a linear regression in python (see jupyter notebook on github).

```
import yfinance as yf
import matplotlib.pyplot as plt

DAX = yf.Ticker('%5Egداخی')
DAX_History = DAX.history(start="2020-01-01", end="2020-10-26")

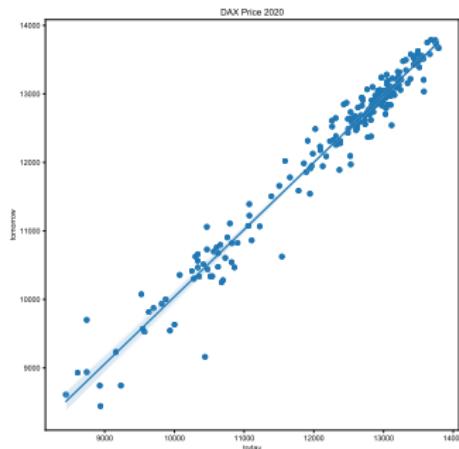
plt.figure(figsize=(10,10))
plt.plot(DAX_History.index, DAX_History['Close'])

# Linear Regression example: regress tomorrow on today
x = DAX_History['Close'][:-1]      # without the last value
y = DAX_History['Close'][1:]       # without the first value

import numpy as np
from numpy import array
from sklearn.linear_model import LinearRegression

model = LinearRegression()
x = array(x).reshape(-1,1)      # The linear model needs a 2D array
y = array(y).reshape(-1,1)
model.fit(x, y)    # values in model.intercept_ and model.coef_

# Give a very sophisticated plot
import seaborn as sns; sns.set_theme(style="white")
ax = sns.regplot(x=x, y=y)
plt.show()
```



Could we improve this ? Suggestions ?

Difference to statistics

- ▶ In a statistical approach we start with a **parametric model**:

$$Y_i = \alpha + \beta x_i + \epsilon_i, \quad i = 1, \dots, n$$

- ▶ and assume that $\epsilon_1, \dots, \epsilon_n$ have a certain structure (for example, i.i.d. and $\mathcal{N}(0, \sigma^2)$).
- ▶ Then one can derive (see, e.g. Czado & Schmidt (2011)) **optimal estimators** for α and β . One can also relax the assumptions and gets weaker results.
- ▶ So what? What are the advantages of the statistical approach?
- ▶ One particular outcome is that we are able to provide **confidence intervals, predictive intervals** and **test hypotheses**.

Questions

- ▶ What is the definition of Machine Learning?
- ▶ Give examples
- ▶ Give surprising examples
- ▶ Derive the main equation of linear regression
- ▶ (do it in 1 dimension first - this goes back to Gauss)
- ▶ Write your own python code, providing a linear regression on your favourite stock
- ▶ Do this with your least favourite stock
- ▶ Can you regress two stocks on each other ?
- ▶ Can you predict better the value of the stock tomorrow ?
(You can also research on this ...)

Generalized Linear Models

We already saw that transforming the input variables suitable might be helpful. This is the idea of a generalized linear model (GLM), see Casella & Berger (2002).

Definition

A GLM consists of three components:

1. Response variables (random) Y_1, \dots, Y_n ,
2. a systematic component of the form $\alpha + \beta^\top \mathbf{x}_i$, $i = 1, \dots, n$,
3. a link function g satisfying

$$\mathbb{E}[Y_i] = g(\alpha + \beta^\top \mathbf{x}_i), \quad i = 1, \dots, n.$$

Regularization of multiple linear regression

- ▶ One problem in practice is parsimony of a linear regression: suppose you have many covariates and you want to include only those which are relevant.
- ▶ It would be possible to iteratively throw out those parameters which are not significant. This procedure, however is not optimal. Many others have been proposed.
- ▶ We concentrate on **continuous** subset selection methods: it is better to introduce a penalty for including too many parameters, which we call regularization. This is moreover a standard procedure for ill-posed problems. We will consider a famous example: the **LASSO** introduced in [R. Tibshirani \(1996\). „Regression Shrinkage and Selection via the Lasso“](#). In: [Journal of the Royal Statistical Society. Series B \(Methodological\)](#) 58.1, pp. 267–288.

- ▶ The least absolute shrinkage and selection operator minimizes the following function

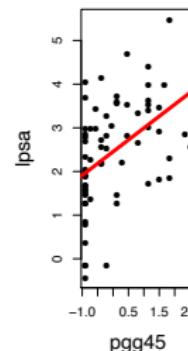
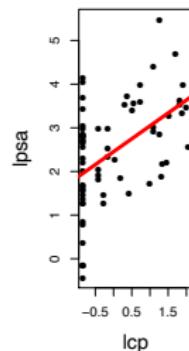
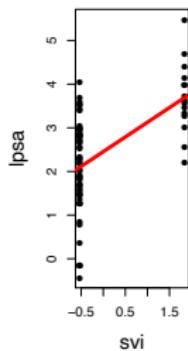
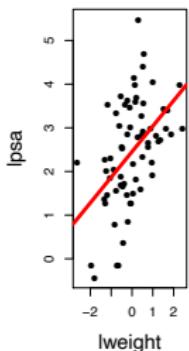
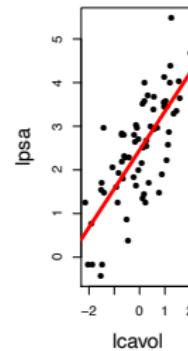
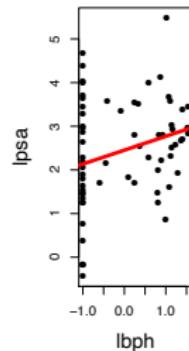
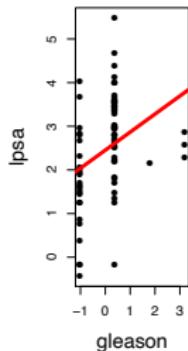
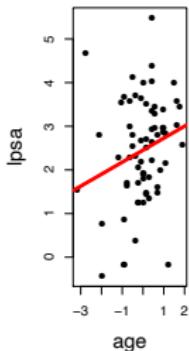
$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \| \mathbf{Y} - \mathbf{x}\beta \|_2^2 + \lambda \| \beta \|_1 \right\}.$$

The parameter λ has to be chosen and allows to vary the level of regularization. Clearly this model prefers to set non-significant parameters to zero.

- ▶ Let us illustrate the lasso with an example taken from Chris Franck, <http://www.lisa.stat.vt.edu/?q=node/5969>. The data stems from Stamey et.al.¹².
- ▶ The data describes clinical measures from 97 men about to undergo radical prostatectomy. It is of interest to estimate the relation between the clinical measures and the prostate specific antigen (measures are: lcavol - log (cancer volume), lweight - log(prostate weight volume), age, lbph - log (benign prostatic hyperplasia), svi - seminal vesicle invasion, lcp - log(capsular penetration), Gleason (score), ppg45 - percent Gleason scores 4 or 5, Y =lpsa - log(prostate specific antigen))

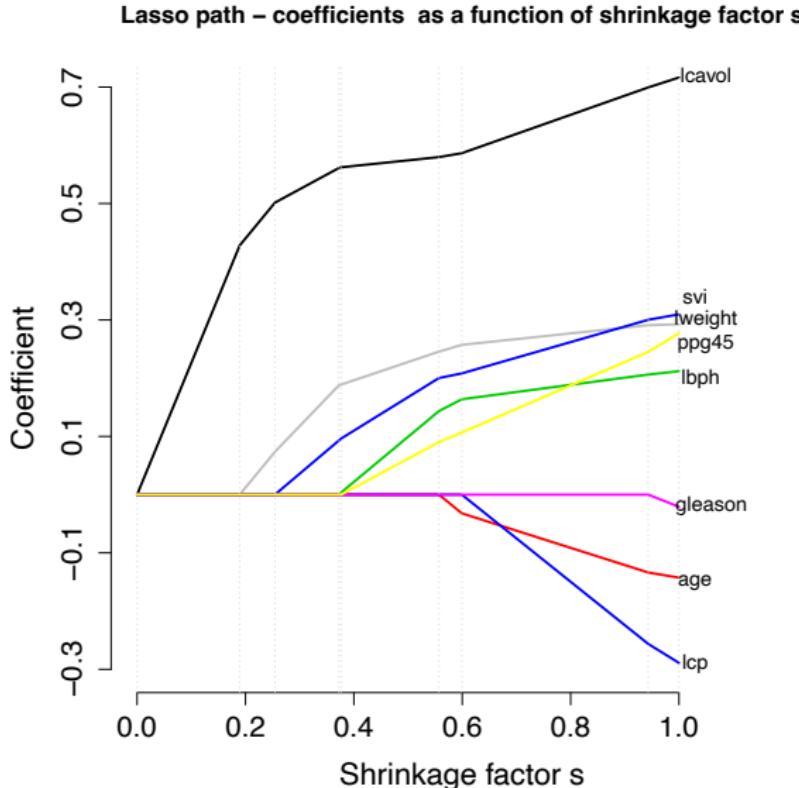
¹²T. A. Stamey et al. (1989). „Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients.“. In: *The Journal of urology* 141.5, pp. 1076–1083.

We start by examining bi-variate regressions.

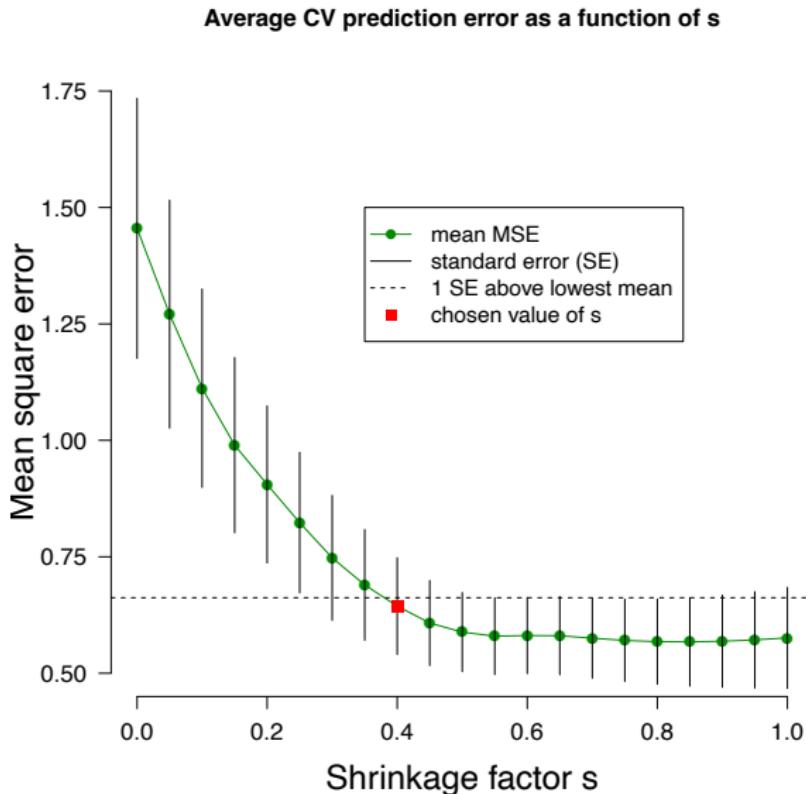


- ▶ It is obvious that some variables have fewer impact and some others seem to be more important. The question is how to effectively select those.
- ▶ We illustrate how cross-validation may be used in this case. This means we separate the data into a training set and a validation set. The tuning parameter λ is chosen based on the training set and validated on the validation set.
- ▶ We use a 10-fold cross validation, ie. the set is split into 10 pieces. Iteratively, each piece is chosen as the validation set while the remaining 9 sets are used to estimate the model.

This is the so-called lasso path. The shrinkage factor is antiproportional to λ .



This is the cross-validation result. A rule of thumb is to select that value of s that is within 1 standard error of the lowest value.



Remarks and Questions

- ▶ We see that the optimal choice of λ is far from trivial. Alternative approaches are at hand, compare the recent results by Johannes Lederer and coauthors, [J. Lederer and C. Müller \(Apr. 2014\)](#). „Don't Fall for Tuning Parameters: Tuning-Free Variable Selection in High Dimensions With the TREX“. In: ArXiv e-prints. eprint: [1404.0541 \(stat.ME\)](#).
- ▶ What is a generalized linear model? Where are the differences to a linear model?
- ▶ What is the LASSO ?
- ▶ What are the differences to simple least squares ?
- ▶ What is an ill-posed problem ? Why do you regulate this ? Why is linear regression an ill-posed problem ?
- ▶ What is cross-validation ?

Please note that I encourage you to do research in the internet on words you don't know. Use the references, use google, google scholar, use the katalog at uni freiburg to find online resources for books and literature, use Wikipedia, use the mathematical encyclopdia or discuss with chatGPT ...

Logistic regression

- ▶ One important regression approach for **classification** is logistic regression.
- ▶ We start by considering **simple** logistic regression, i.e. the classification into **two** classes. In this case, the response is always binary.
- ▶ One therefore needs to transform the whole real line to $[0, 1]$ and two approaches are common: first, via the logistic function

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

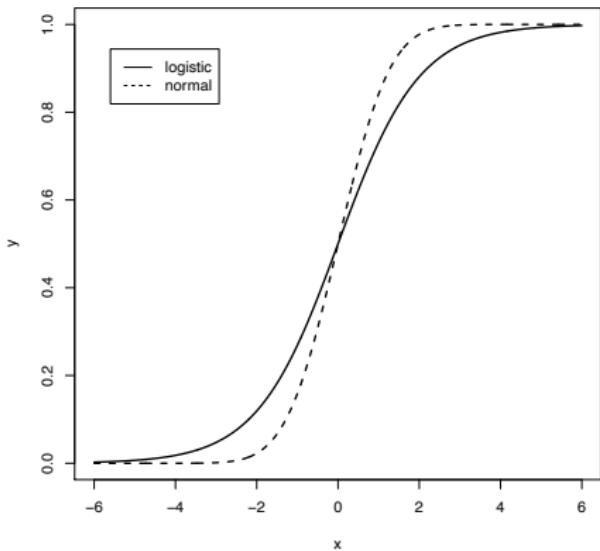
The most common way is to transform y via

$$\sigma^{-1}(p) = \text{logit}(p) = \log \frac{p}{1 - p},$$

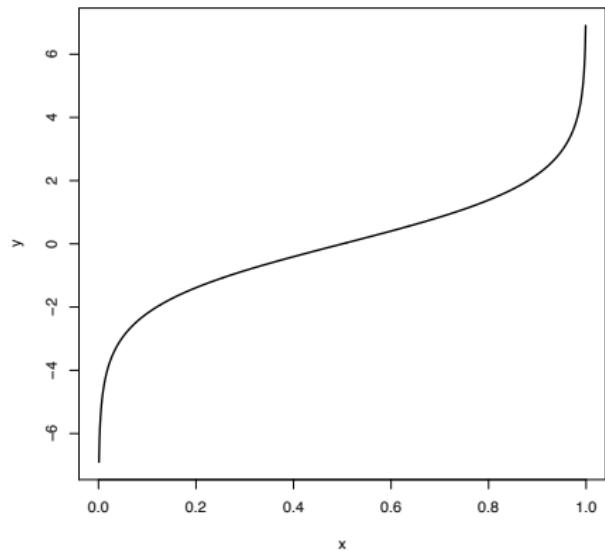
the so-called **logit** function.

- ▶ Second, by a cumulative distribution function (when this is Φ - standard normal - this approach is called **probit** model).

Logistic function



Logit function



Definition (Logistic regression)

A logistic regression is the generalized linear model where

$$\text{logit}(p_i) = \alpha + \boldsymbol{\beta}^\top \mathbf{x}_i, \quad i = 1, \dots, n.$$

Note that this model is equivalent to

$$p_i = \frac{\exp(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i)}{1 + \exp(\alpha + \boldsymbol{\beta}^\top \mathbf{x}_i)}.$$

The observations y_1, \dots, y_n are binary, hence take values in $\{0, 1\}$ and are assumed to be i.i.d. Bernoulli with $P(y_i = 1) = p_i = p_i(\mathbf{x}_i)$.

A nice source explaining the depth of logistic regression and their various applications is¹³.

¹³Ronald Christensen (2006). **Log-linear models and logistic regression**. Springer Science & Business Media.

The most common estimation method used is **maximum-likelihood**. We take a small detour towards this exciting statistical concept going back to Sir Ronald Fisher.

Maximum-likelihood

- ▶ A **statistical model** is given by a family of probability measures $(P_\theta)_{\theta \in \Theta}$ on a common measurable space (Ω, \mathcal{F}) . It is typically called **parametric**, if Θ is of finite dimension.
- ▶ The **likelihood**-function for the observation E is given by

$$L(\theta) = P_\theta(E)$$

If $P_\theta(E) = 0$ for all $\theta \in \Theta$ one proceeds via the density: assume $P_\theta \ll P^*$ for all $\theta \in \Theta$ and denote the densities by $f_\theta := dP_\theta/dP^*$. Then, for the observation x ,

$$L(\theta) = f_\theta(x).$$

- ▶ This looks complicated, but is in most cases quite simple: consider i.i.d. random variables X_1, \dots, X_n with common density f_θ . Then P^* is clearly the Lebesgue-measure. Due to the i.i.d.-property,

$$L(\theta) = \prod_{i=1}^n f_\theta(x_i).$$

Definition

Any maximizer $\hat{\theta}$ of the likelihood-function is called maximum-likelihood estimator for the model $(P_\theta)_{\theta \in \Theta}$.

In the above example, we need to maximize $\prod_{i=1}^n f_\theta(x_i)$, which is typically infeasible. One therefore considers the log-likelihood function

$$\ell(\theta) := \ln L(\theta)$$

which is often much easier to maximize. Typically one can apply first-order conditions or needs to solve numerically.

Example (ML for the normal distribution)

Consider $X_i \sim \mathcal{N}(\mu, 1)$. Then the density is

$$f_\theta(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x - \mu)^2\right).$$

We obtain the log-likelihood function

$$l(\theta) = \text{const.} - \frac{1}{2} \sum_{i=1}^n (x_i - \mu)^2.$$

The first derivative is

$$\partial_\mu l(\theta) = \sum_{i=1}^n x_i - n\mu \stackrel{!}{=} 0$$

and we obtain the maximum-likelihood estimator (second derivative is < 0)

$$\hat{\mu} = \bar{x} = \frac{\sum_{i=1}^n x_i}{n}.$$

Exercise: compute the ML estimator for σ ! Read Czado & Schmidt (2011) on ML-estimation and further estimation procedures.

Maximum-Likelihood for the logistic regression

- ▶ For the logistic regression, where y_1, \dots, y_n are Bernoulli, we obtain the likelihood function

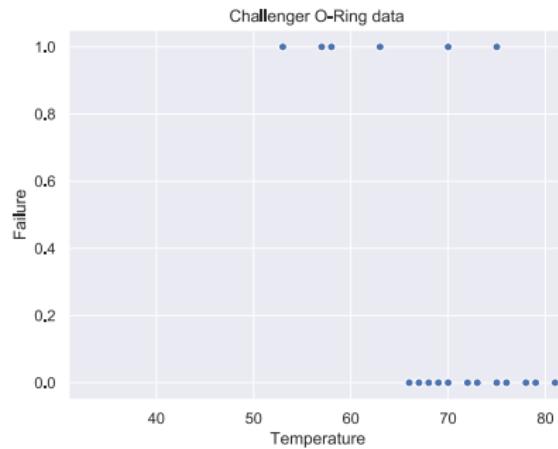
$$L(\mathbf{y}) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i}.$$

- ▶ Maximization has to be done numerically, eg. by gradient descent or by weighted least squares.
- ▶ Asymptotic distributions are available, such that we can test approximately several hypothesis, like for example $\beta_i = 0$ or $\alpha = 0$.

Back to logistic regression. We look at the by now infamous Challenger¹⁴ O-ring data set (taken from Casella & Berger (2002))

1 53	1 57	1 58	1 63	0 66	0 67	0 67	0 67	0 68	0 69	0 70	0 70	1 70	1 70	0 72	0 73	0 75	1 75
---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

The table reports failures with associated temperature.



¹⁴See https://en.wikipedia.org/wiki/Space_Shuttle_Challenger_disaster.

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression

x = np.array([53,57,58,63,66,67,67,68,69,70,70,70,70,72,73,
              75,75,76,76,78,79,81]).reshape(-1, 1)
y = np.array([1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
              0, 1, 0, 0, 0, 0, 0]).reshape(-1, 1)

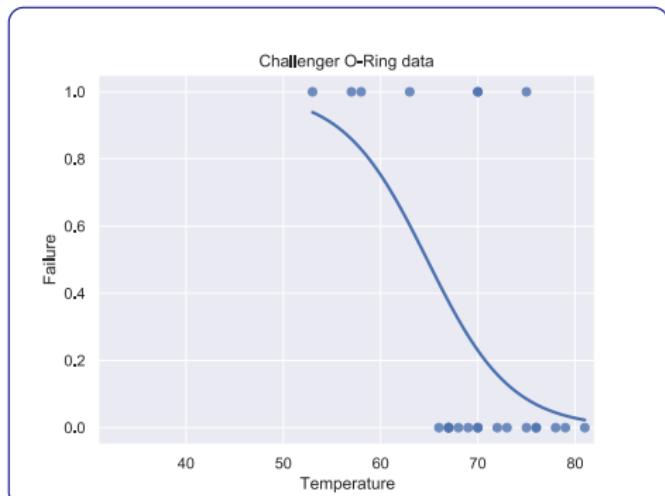
# logistic regression model
logreg = LogisticRegression().fit(x, y)

print(logreg.intercept_, logreg.coef_[0])
#[0.52055518] [-0.02100215]

import seaborn as sns
sns.set_theme(color_codes=True)
sns.regplot(x=x, y=y, logistic=True)
plt.show()

```

The estimated probability
for a failure at 31 degree is 0.9996088.

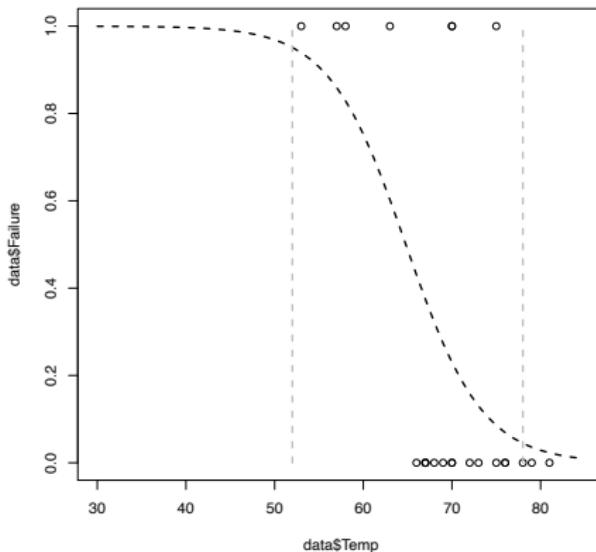


- ▶ Logistic regression naturally classifies the data into two fields: the ones with probability above 0.5, where we would optimally decide for outcome one and the ones with probability below 0.5, where we would decide for outcome 0.
- ▶ Hence, we obtain a **decision boundary**, given by the hyperplane

$$\alpha + \beta x = 0.$$

- ▶ If the decision boundary separates the two groups, then the data is called **linearly separable**. Note that this can not be achieved in the Challenger dataset.
- ▶ Note that the logistic regression also provides probabilities of false decisions: at the boundary this is 50/50, but further out the probability of a false decision decrease. **Significant decisions** requires the probability of a false decision to be below a significance level, e.g. $\alpha = 0.05$ or $\alpha = 0.01$.

With significance level $\alpha = 0.05$ obtained decision boundaries.



Load the python example¹⁵ from the homepage and revisit the above steps. Try your own examples.

- ▶ The likelihood-function has to be maximized numerically.
- ▶ A first-order iterative scheme is the **gradient-descent** algorithm. Look this algorithm up and recall its properties and functionality.

¹⁵Called 01_05_logistic_regression.py

Questions

- ▶ What is the difference between logistic regression and regression?
- ▶ What are the logit and probit functions ?
- ▶ What is maximum-likelihood?
- ▶ Compute the maximum-likelihood estimator for an exponential distribution.
- ▶ Look up the challenger catastrophe and watch Richard Feynman's famous speech.

The imitation game - der Turing Test

- ▶ Ursprünglich 1960 von Turin in komplexerer Form (imitation game) eingeführt, gilt er heute als einer der Tests auf künstliche Intelligenz
- ▶ Der einfache Turing Test ist eine Kommunikation mit zwei Partnern A und B und man muss am Anschluss entscheiden wer die Maschine ist. Typischerweise würde man viele Personen befragen und erwarten dass man nicht in der Lage ist die beiden Personen zu unterscheiden, die Wahrscheinlichkeit die Maschine zu identifizieren also nicht über 50% betragen sollte.
- ▶ Es gibt einige berühmte Beispiele. Starten wir mit
- ▶ ELIZA (1960s - Joseph Weizenbaum) ein Chatbot der im Emacs integriert ist. Probieren wir diesen doch einmal aus.



I am the psychotherapist. Please, describe your problems. Each time you are finished talking, type RET twice.

i am bad at maths

Why do you say you are bad at maths?

my teacher tells me so

Tell me about your teacher.

oh - he is crazy. even if he is a probabilist, he is teaching us machine learning. he thinks computers are intelligent

You have your mind on computers, it seems.

a little bit.

Is it because of your plans that you say a little bit?

yes, my plan is to become a famous mathematician

Maybe your life has something to do with this.

my life ? do you think i am not doing enough maths ? should i do more homework ?

What makes you believe that?

because my teacher tells me i should do all the homework in the class. but it is so much.

Is it because your teacher tells you you should do all the homework in the class but it is so much that you came to me?

because it is so much.

Does it bother you that it is so much?

yes. it is tough to do all the homework on my own

Earlier you said your teacher tells you you should do all the homework in the class but it is so much?

yes - that is right. the teacher gives a lot of homework and it is so much. i want to go out and do skateboard

When did you first know that is right the teacher gives a lot of

Weitere Beispiele

- ▶ In den 1980s wurde Cleverbot entwickelt, der erste Tests gut überstand
- ▶ 2017 gab es auf der ART Basel ein Projekt in welchem KI-generierte Bilder mit denjenigen von zeitgenössischen Künstlern ausgestellt wurden. Ein großer Teil der Zuschauer:innen bevorzugte die KI-generierten Bilder (durch ein Creative Adversarial Network erzeugt).
- ▶ Schauen Sie auch einmal an meinem Büro vorbei ...



N=4

Artificial Art

Telmo
D'Amico

T. D'Amico

Chat GPT

- ▶ Nun können wir natürlich auch mit Chat GPT einen Turing Test machen. Los gehts.