Statistisches Data Mining (StDM)
Woche 6

Oliver Dürr

Institut für Datenanalyse und Prozessdesign

Zürcher Hochschule für Angewandte Wissenschaften

oliver.duerr@zhaw.ch

Winterthur, 25 Oktober 2016

No laptops,
no phones, no problems

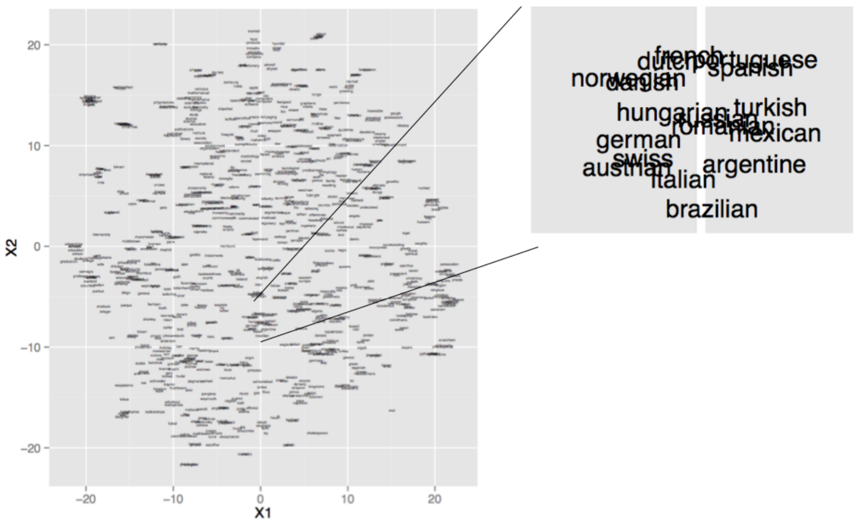
Multitasking senkt Lerneffizienz:

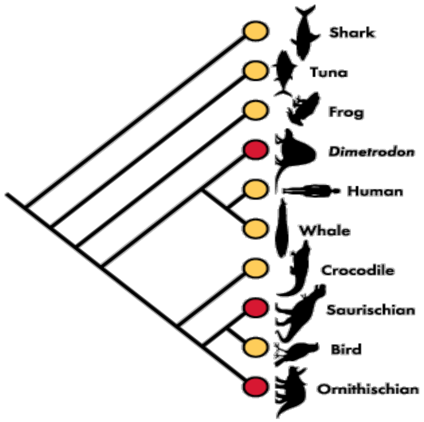
- **Keine Laptops im Theorie-Unterricht Deckel zu oder fast zu (Sleep modus)**

Overview of the semester

Part I (Unsupervised Learning)

- Dimension Reduction
    - PCA
- Similarities, Distance between objects
    - Euclidian, L-Norms, Gower,…
- Visualizing Similarities (in 2D)
    - MDS, t-SNE
- Clustering
    - K-Means
    - Hierarchical Clustering



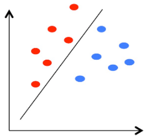


Part II (Supervised Learning)

- …

Overview of classification (until the end to the semester)

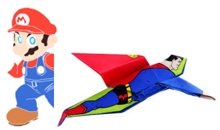
Classifiers



K-Nearest-Neighbors (KNN)
Logistic Regression
Linear discriminant analysis
Classification Trees
Support Vector Machine (SVM)
Neural networks NN
Deep Neural Networks (e.g. CNN, RNN)
...

Evaluation

Cross validation
Performance measures
ROC Analysis / Lift Charts

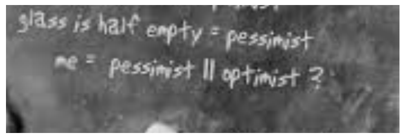


Combining classifiers

Bagging
Boosting
Random Forest

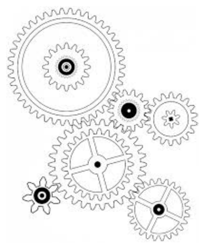
Theoretical Guidance / General Ideas

Bayes Classifier
Bias Variance Trade off (Overfitting)



Feature Engineering

Feature Extraction
Feature Selection



Principal Idea Classification

Klassifikatoren
- Neuronale Netze
- Entscheidungsbäum
- ...

Training Data

id	Type	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	setosa	5.1	3.5	1.4	0.2
2	setosa	4.9	3	1.4	0.2
3	virinica	3.3	3.2	1.6	0.5
4	setosa	5.1	3.5	1.4	0.2
...	...	...	...	...	...
150	virinica	4.9	3	1.4	0.2

Learn a classifier → Classifier

Predict

Unknown data / Test data

d	Type	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	?	3.1	3.5	1.4	0.2
2	?	4.9	3	1.4	0.2
3	?	3.3	3.2	1.6	0.5
4	?	5.1	3.5	31.4	0.2

→ Classifier → Type

Note:
To evaluate the performance a part of the labelled data not used to train the classifier but left aside to check the performance of the classifier to new data.

Examples of Classification Task

- Is a given text e.g. tweet about a product positive, negative or neutral. *Sentiment Analysis*

  “The movie XXX actually neither that funny, nor super witty” → Negative

- Churn in Marketing: Predict which customer wants to quit and offer them a discount

- Spam Detection

- Face detection. Image (array of pixels) ➔ John

…

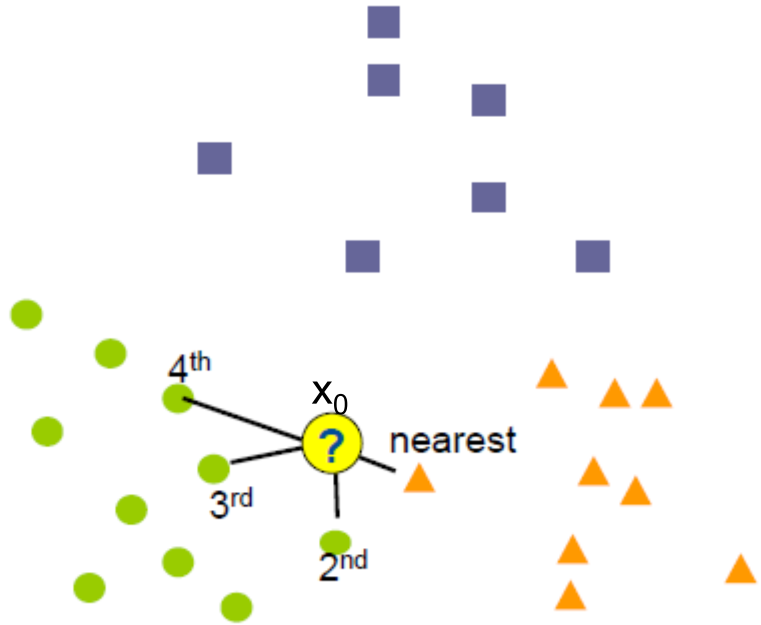
K-Nearest-Neighbors in a nutshell

Idea of knn classification:

- Start with an observation x_0 with unknown class label

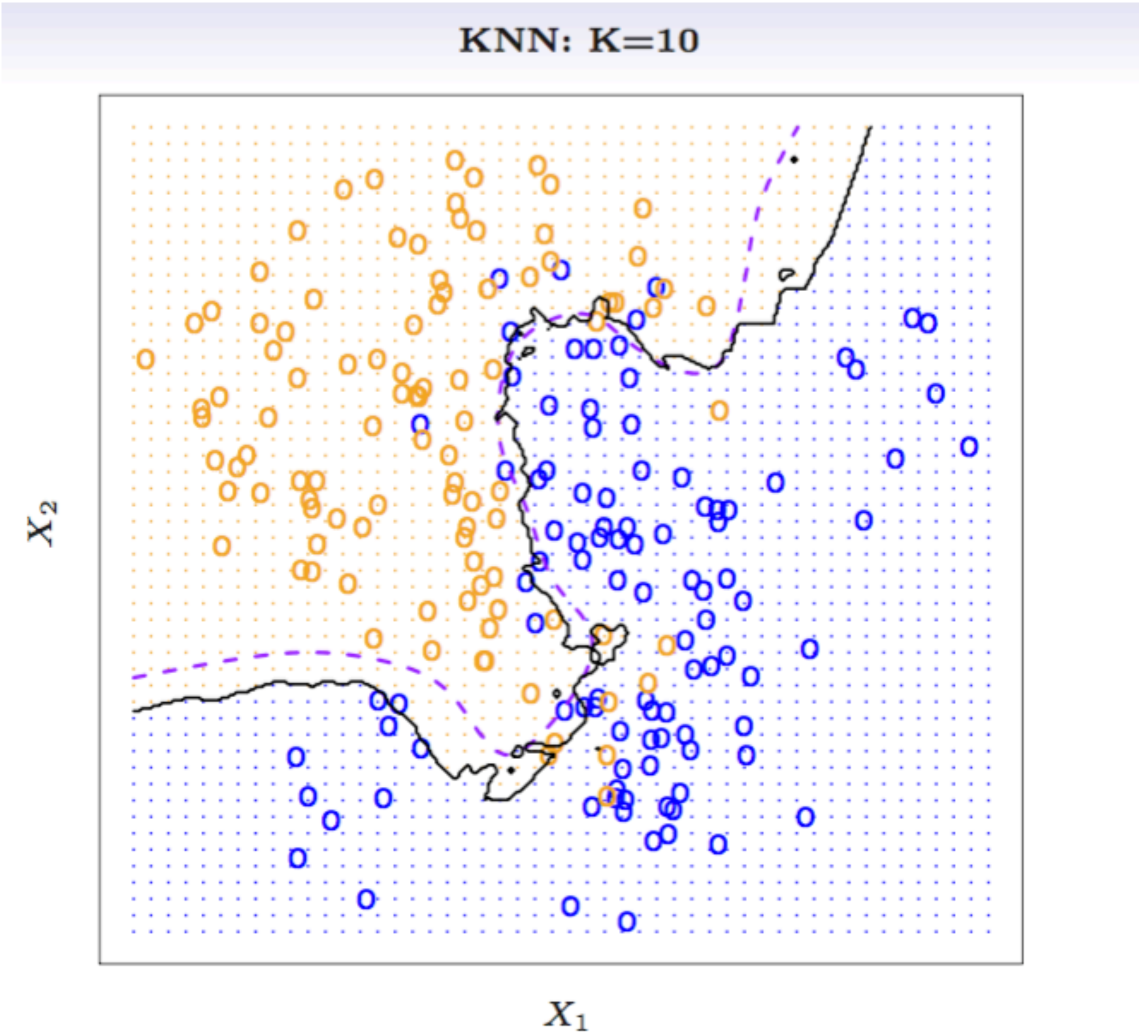
- Find the k training observations, that have the smallest distance to x_0

- Use the majority class among the k neighbors as class label for x_0

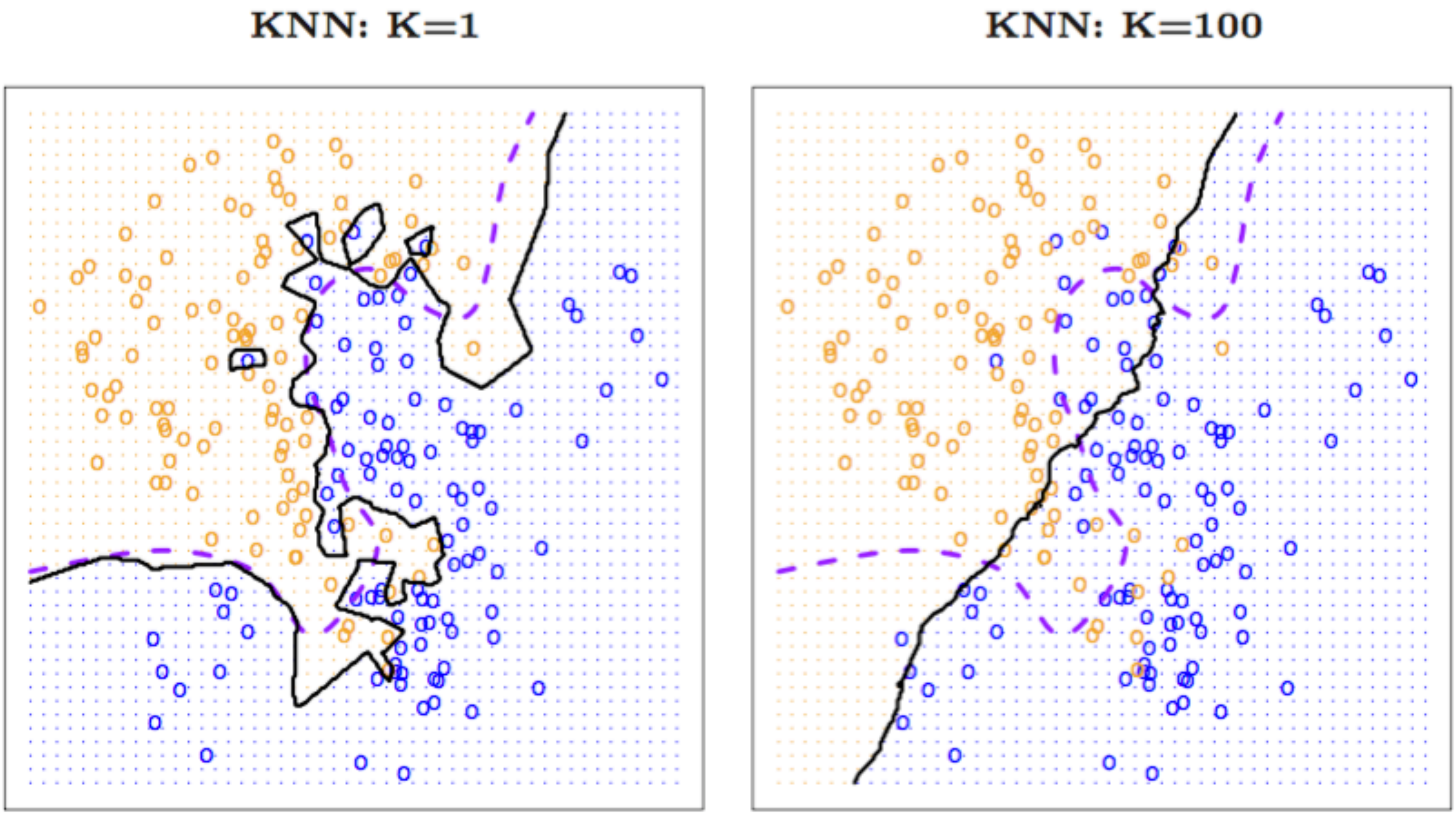


R functions to know
- From package “class”: “knn”

The effect of K



The effect of K



Which k to use? Let's quantify the error / accuracy.

Accuracy as performance measure

Evaluate prediction accuracy on data

Confusion Matrix:

	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

For an ideal classifier the off-diagonal entries should be zero: c=0, b=0, or Accuracy=1

a: TP (true positive)

b: FN (false negative)

c: FP (false positive)

d: TN (true negative)

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Simply count the # correct / all

Source: Tan, Steinbach, Kumar

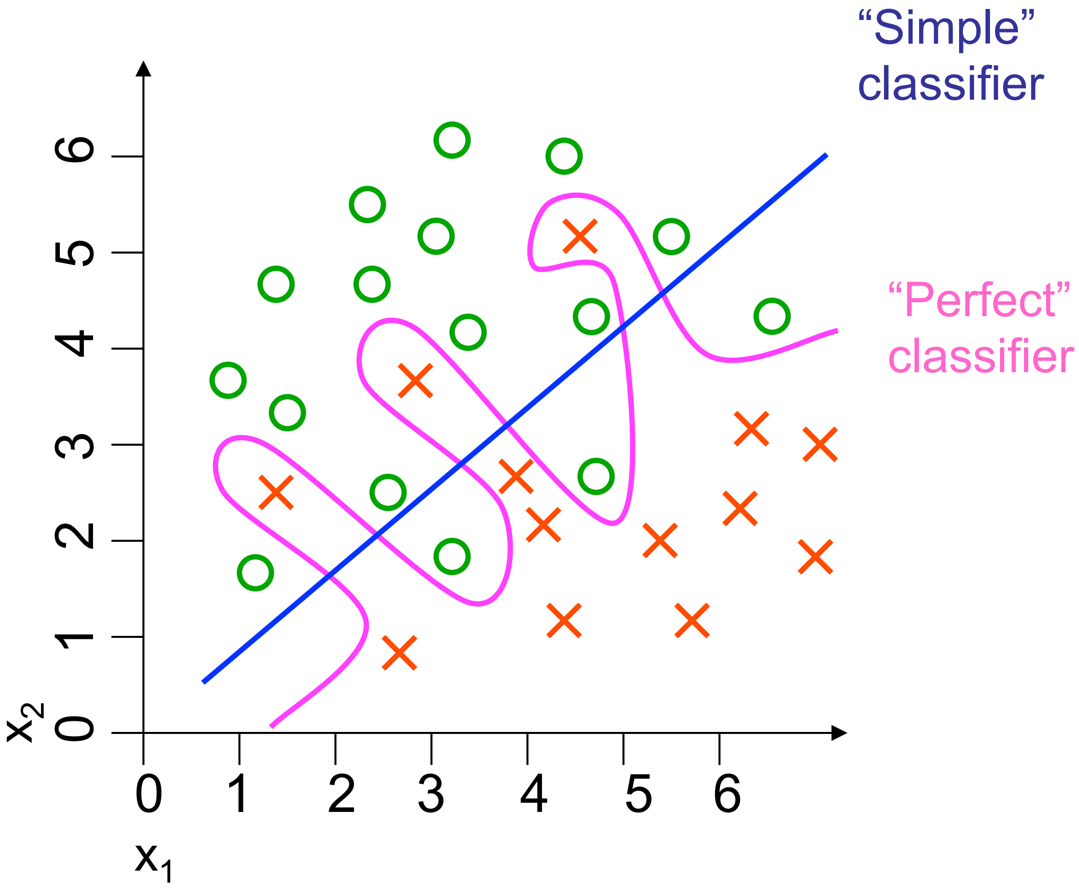
Types of Errors

- **Training error or in-sample-error:**
  - Error on data that have been used to train the model
- **Test error or out-of-sample-error**
  - Error on previously unseen records (out of sample)

Overfitting phenomenon:

- Model fits the training data well (small training error) but shows high generalization error

"Perfect" Vs. "Simple" classifier



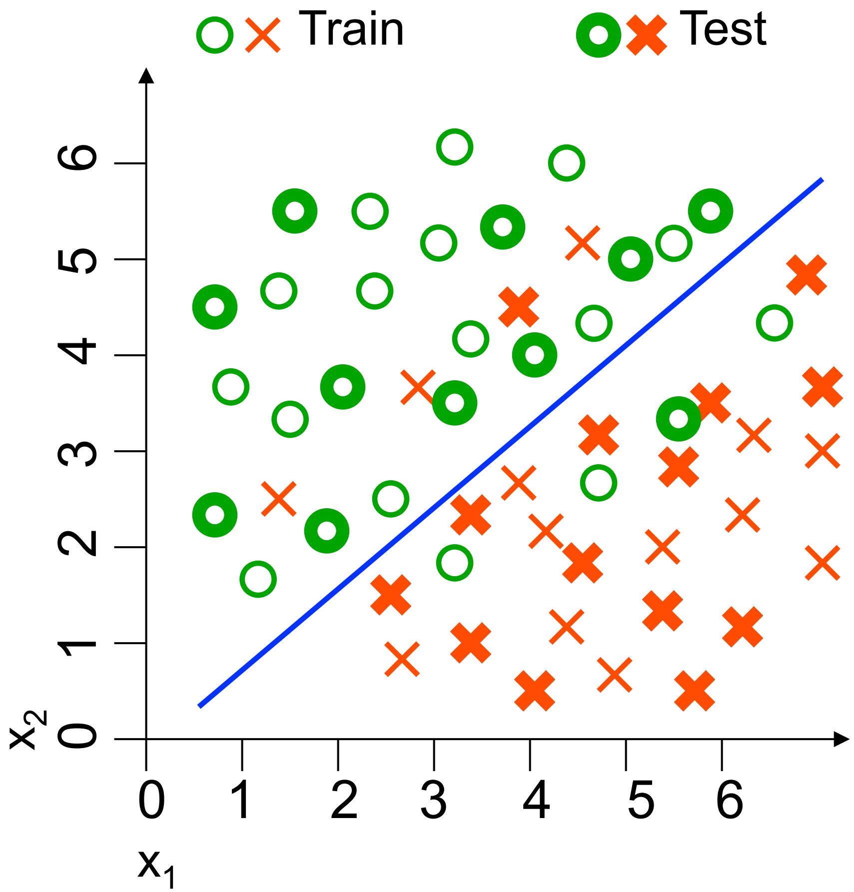
Which is better?

Check on a test-set (don't use all you labeled data to train)

Cross validation of the “simple” classifier

Training set:
6/29=20%
misclassification

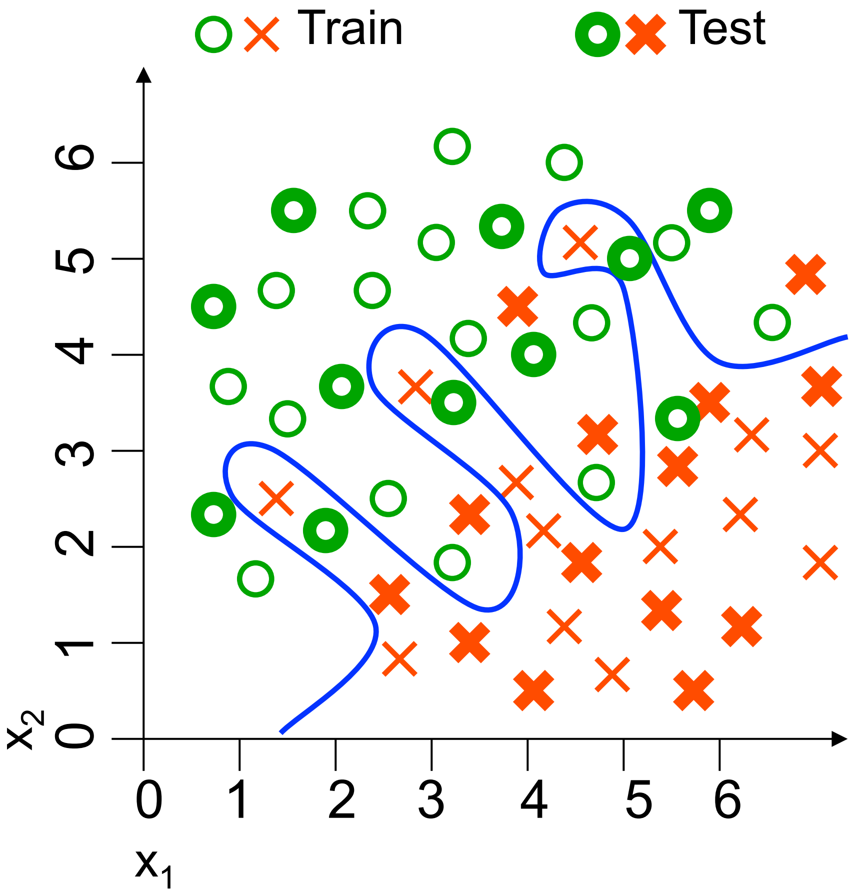
Test set:
2/25=8%
misclassification



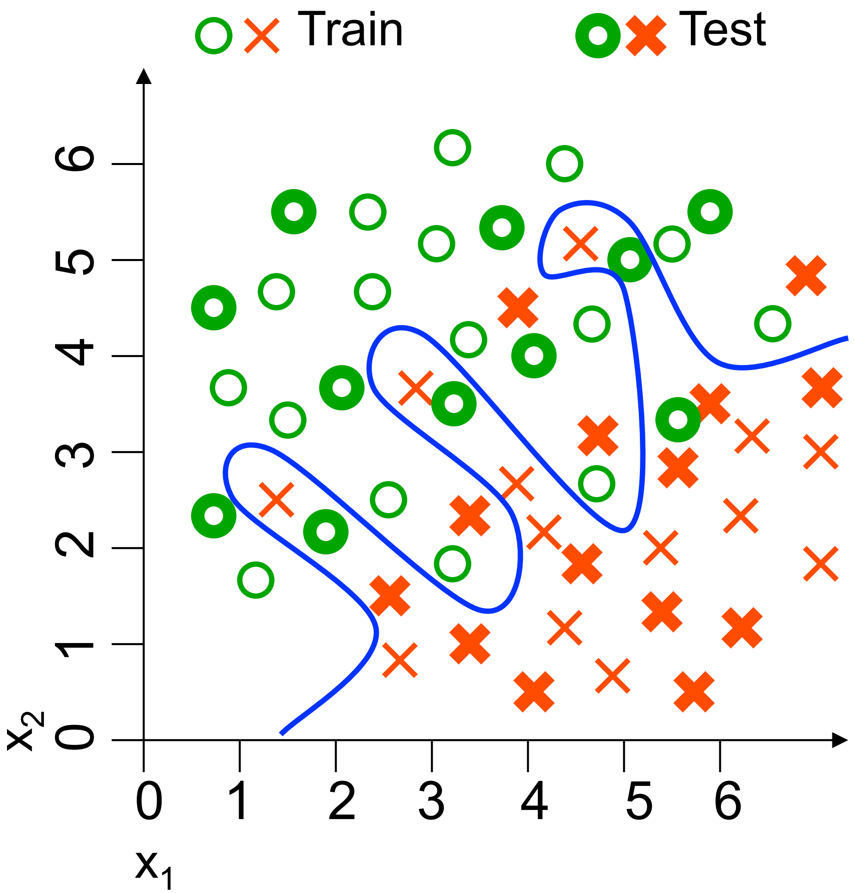
Cross validation of the "Perfect" classifier

Training set:
0%misclassification

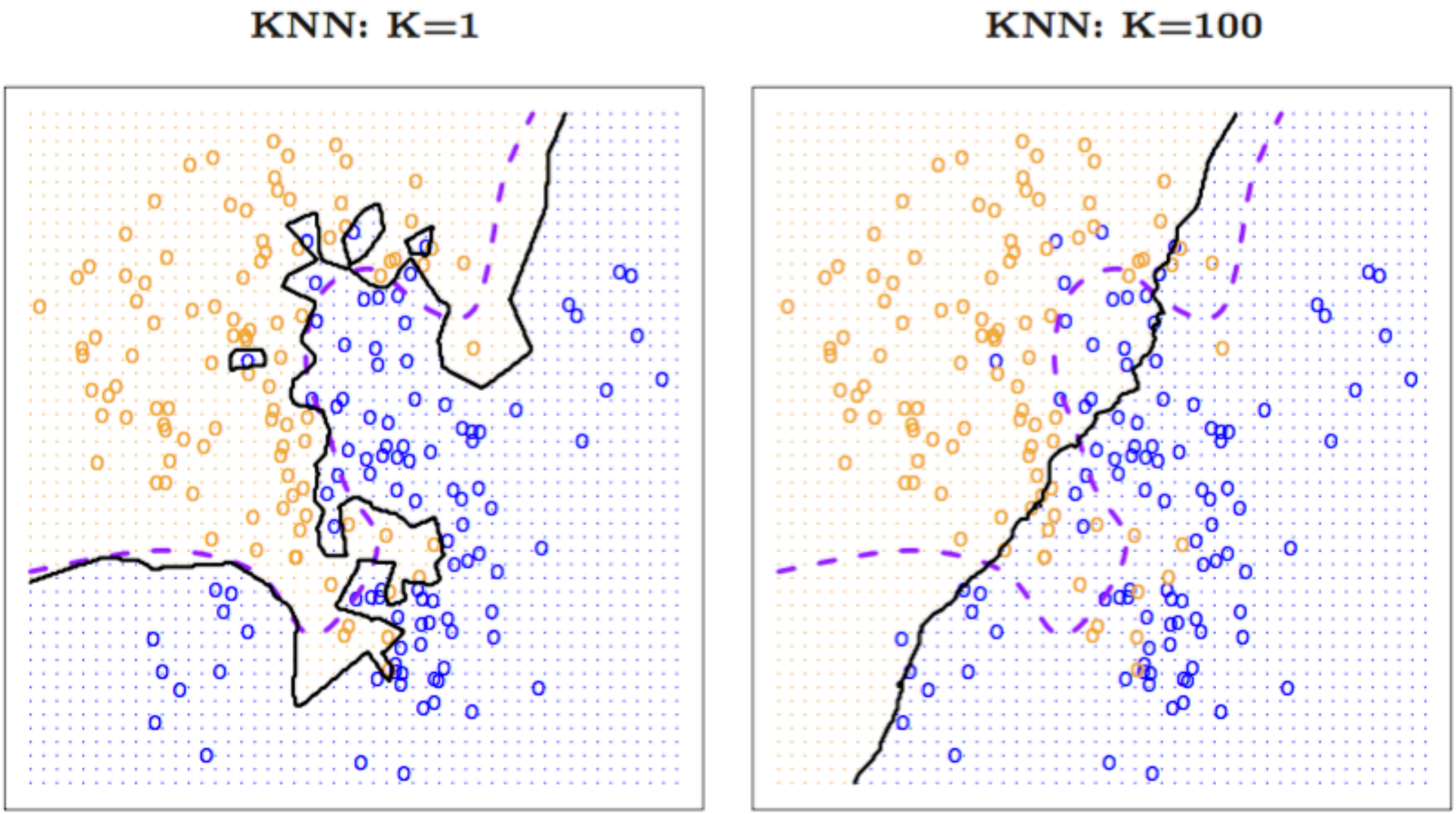
Test set:
8/25=24%
misclassification



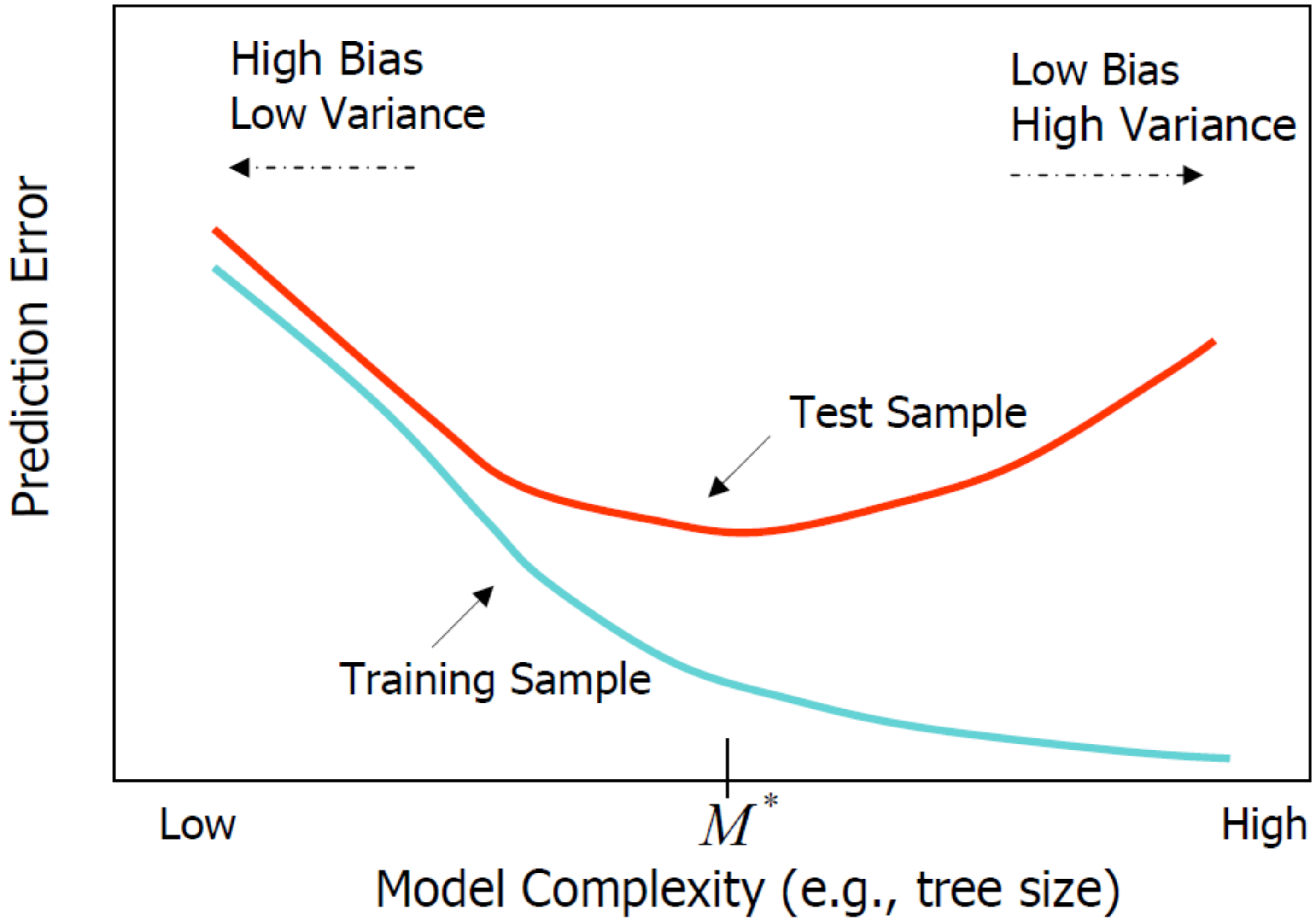
Cross validation of the "Perfect" classifier



Which one to use?

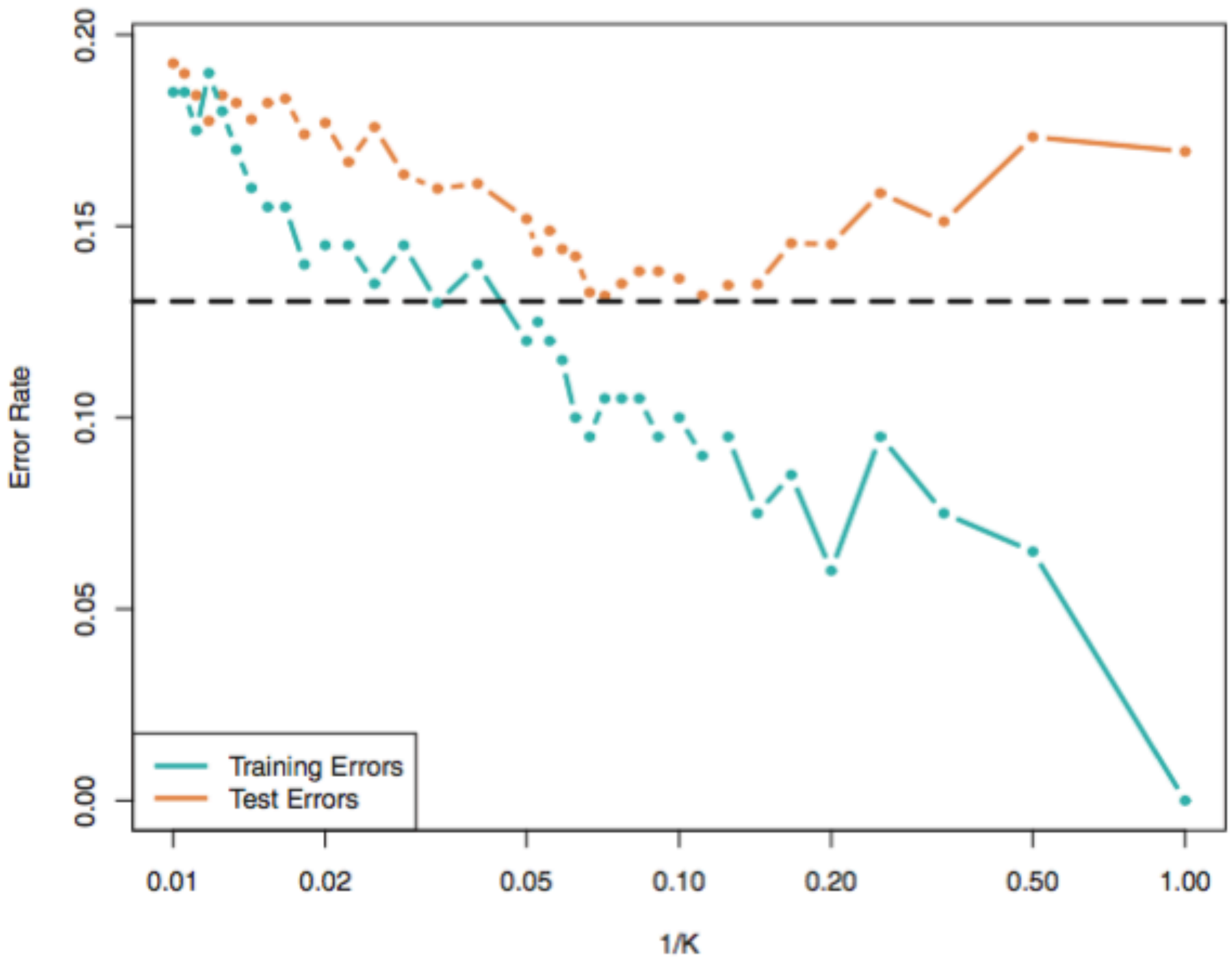


What is the right level of complexity



1/k

What is the right level of complexity



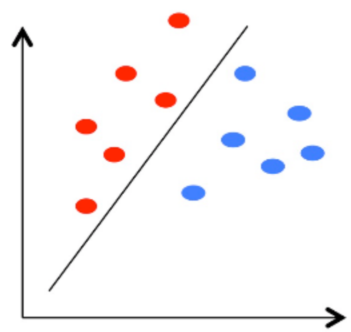
Example from ILSR

Occam's razor

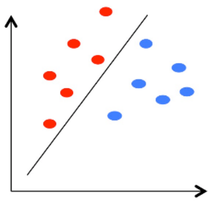
- A more complex model performs always better on training data than a simpler model.
- Models should be evaluated on test data to determine the generalization error.
  If comparing performance on training data the model complexity should be taken into account (penalize for complexity).
- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model

Make everything as simple as possible, but not simpler.
A. Einstein

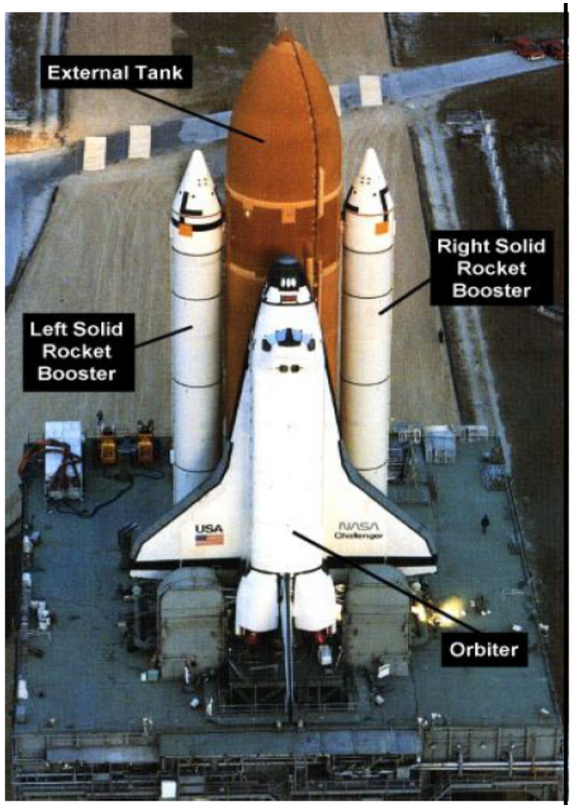
Logistic Regression



Logistic Regression



- See also: ISLR chapter 4.3
- RKST chapter 6.2
- Example

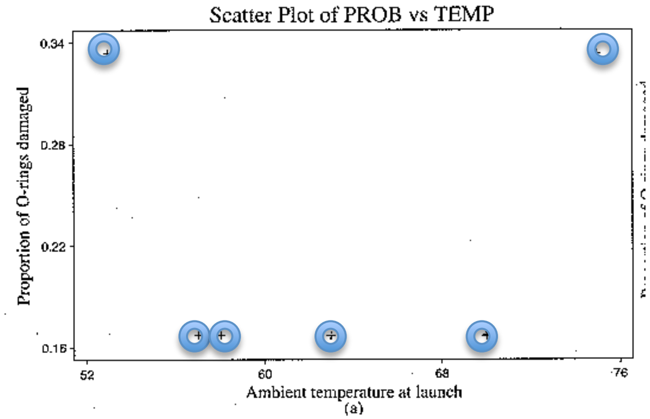


Die bemannte Raumfähre Challenger explodierte 1986 nach dem Start, weil die Dichtungsringe an den Boostern nicht dicht hielten.

Statistik & Challenger Desaster

- Am Tage des Starts war es kalt, 31°F.
- Bei den 23 bisherigen Flügen, gab es bei 7 Probleme mit den Dichtungen.

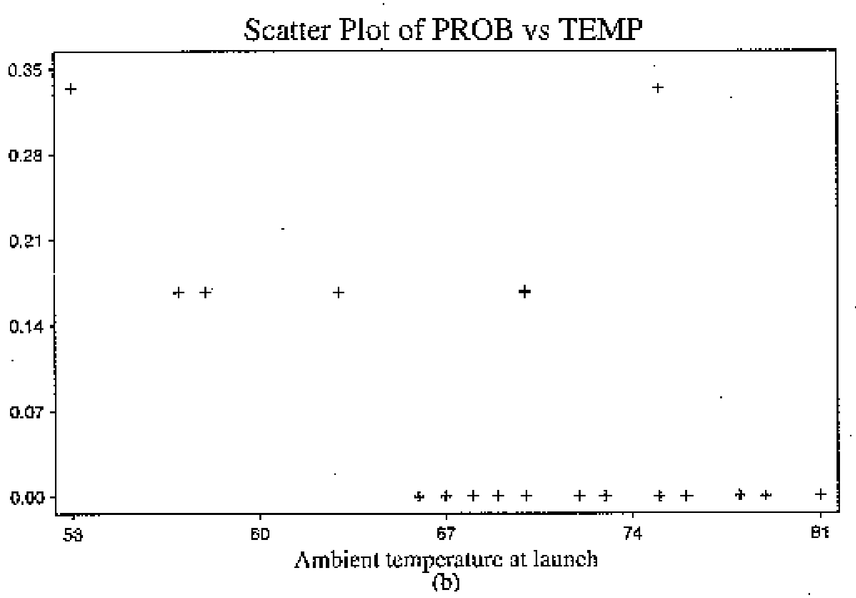
Ambient temperature	Number of O–rings damaged	$\hat{p}$
53°	2	.333
57°	1	.167
58°	1	.167
63°	1	.167
70°	1	.167
70°	1	.167
75°	2	.333

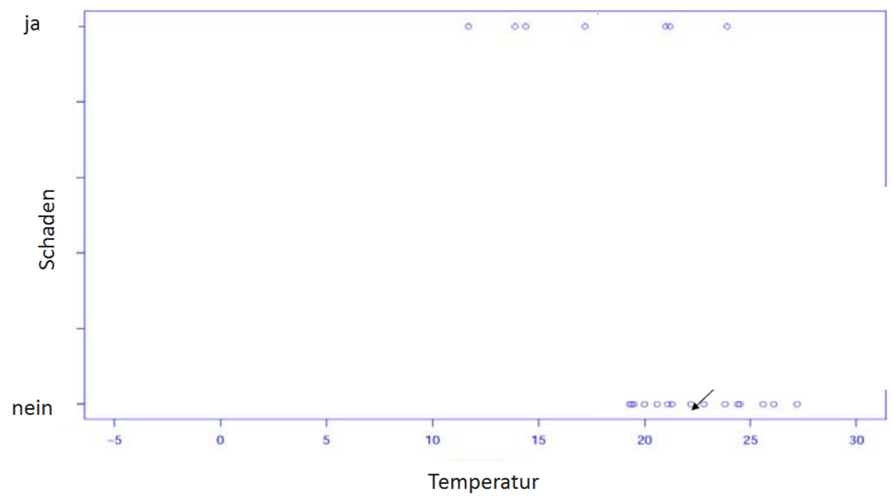


- Erhöhtes Risiko bei kleiner Temperatur?
- Starten ja oder nein? Was ist Ihre Meinung?

Statistik & Challenger Desaster

- Die erfolgreichen Flüge enthalten auch Information.

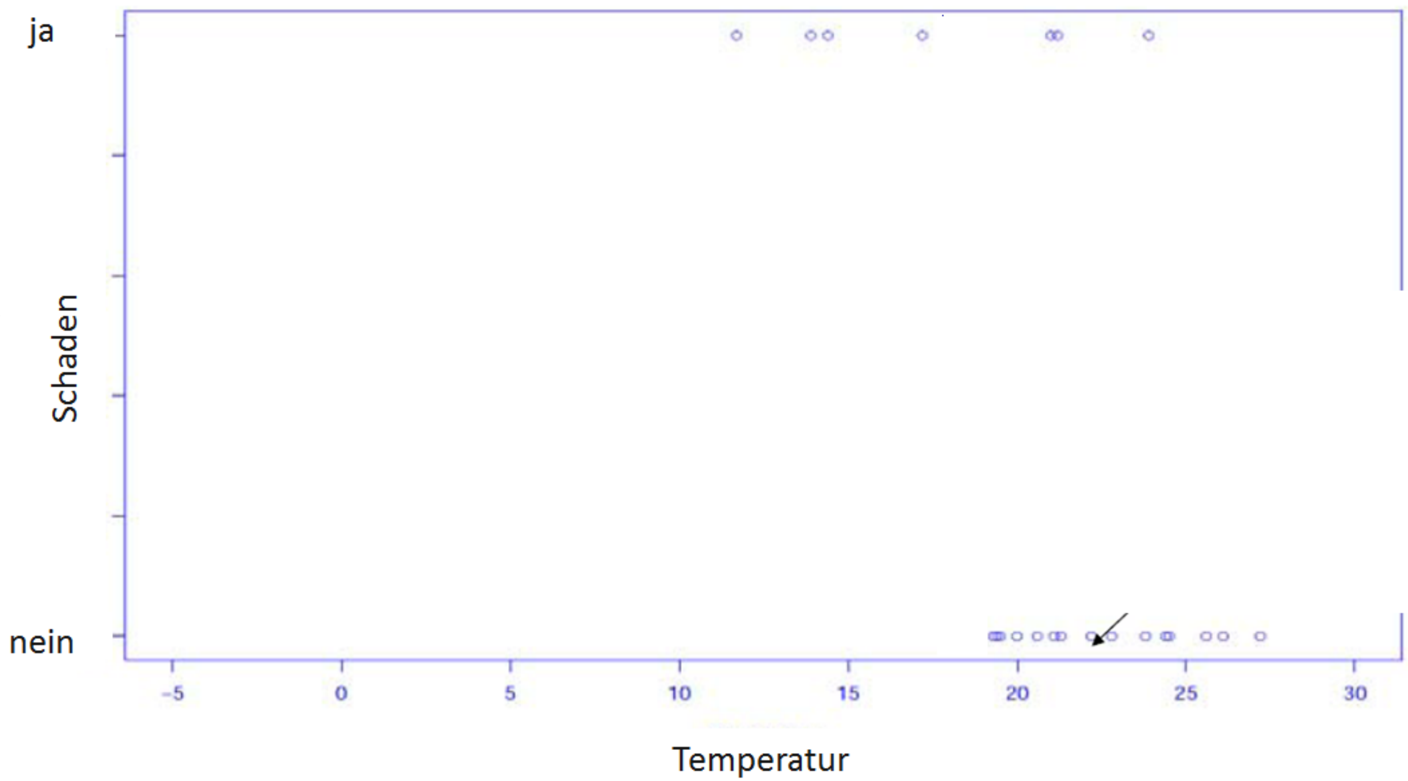




Modelling logistic regression

$p(X) = Pr(Y = 1|X)$ Prob. for a O-ring to be defect at a given temperature X

p(x)

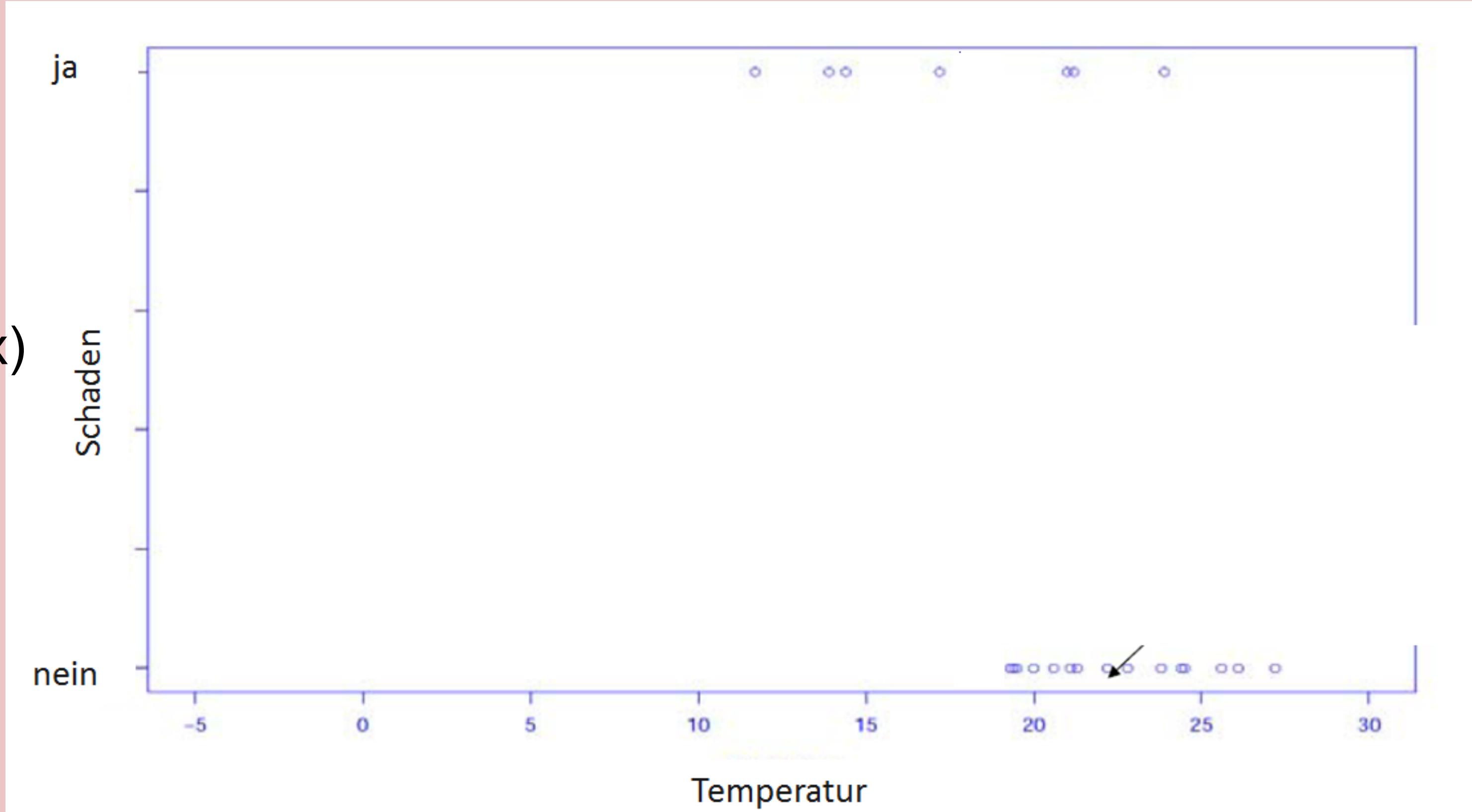


Question: Why is $p(X) = \beta_0 + \beta_1 X$ wrong?

Modelling logistic regression

p(X) = Pr(Y = 1|X) Prob. for a O-ring to be defect at a given temperature X

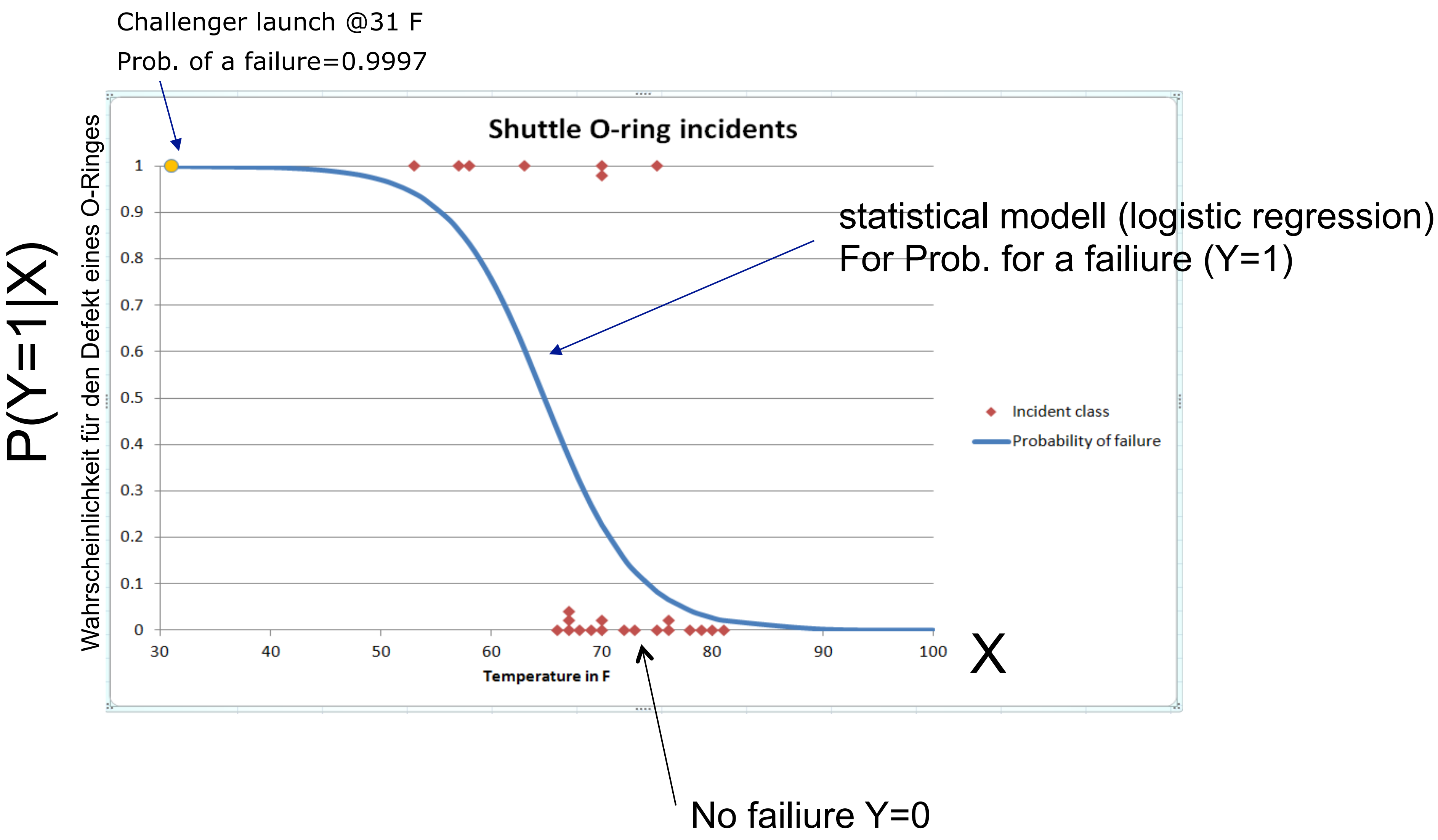
p(x)



Frage: $z = \beta_0 + \beta_1 X$ mit $\beta_1 < 0$ und $\beta_0 = 0$ wie verläuft z und wie $p(X) = [1+\exp(-z)]^{-1}$ für den ganzen Bereich (einzeichnen)

Logistic Regression: Example challenger O-rings

Predict if O-Ring is broken, depending on temperature

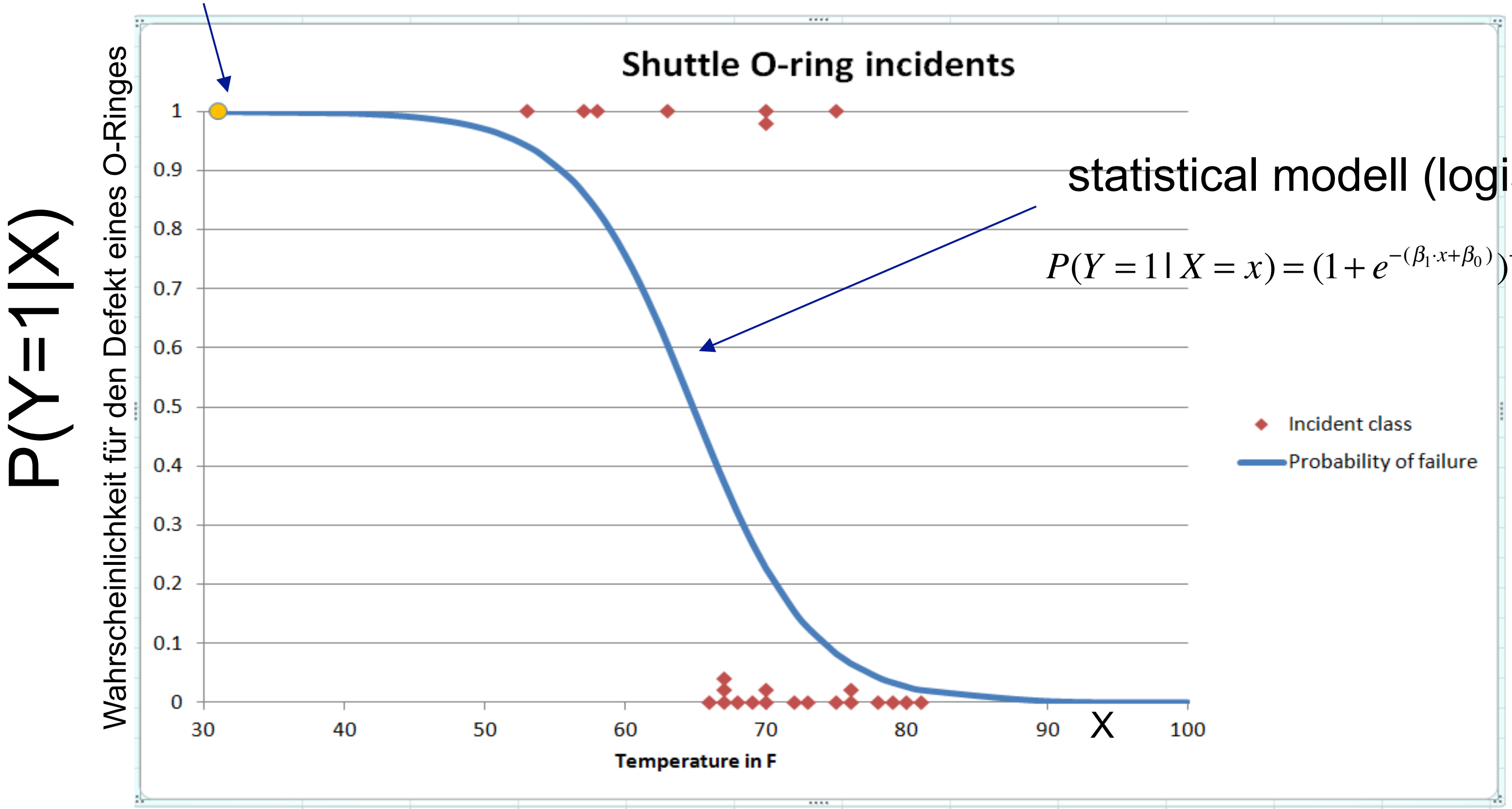


Logistic Regression (recap)

Predict if O-Ring is broken depending on temperature

Challenger launch @31 F

Prob. of a failure=0.9997



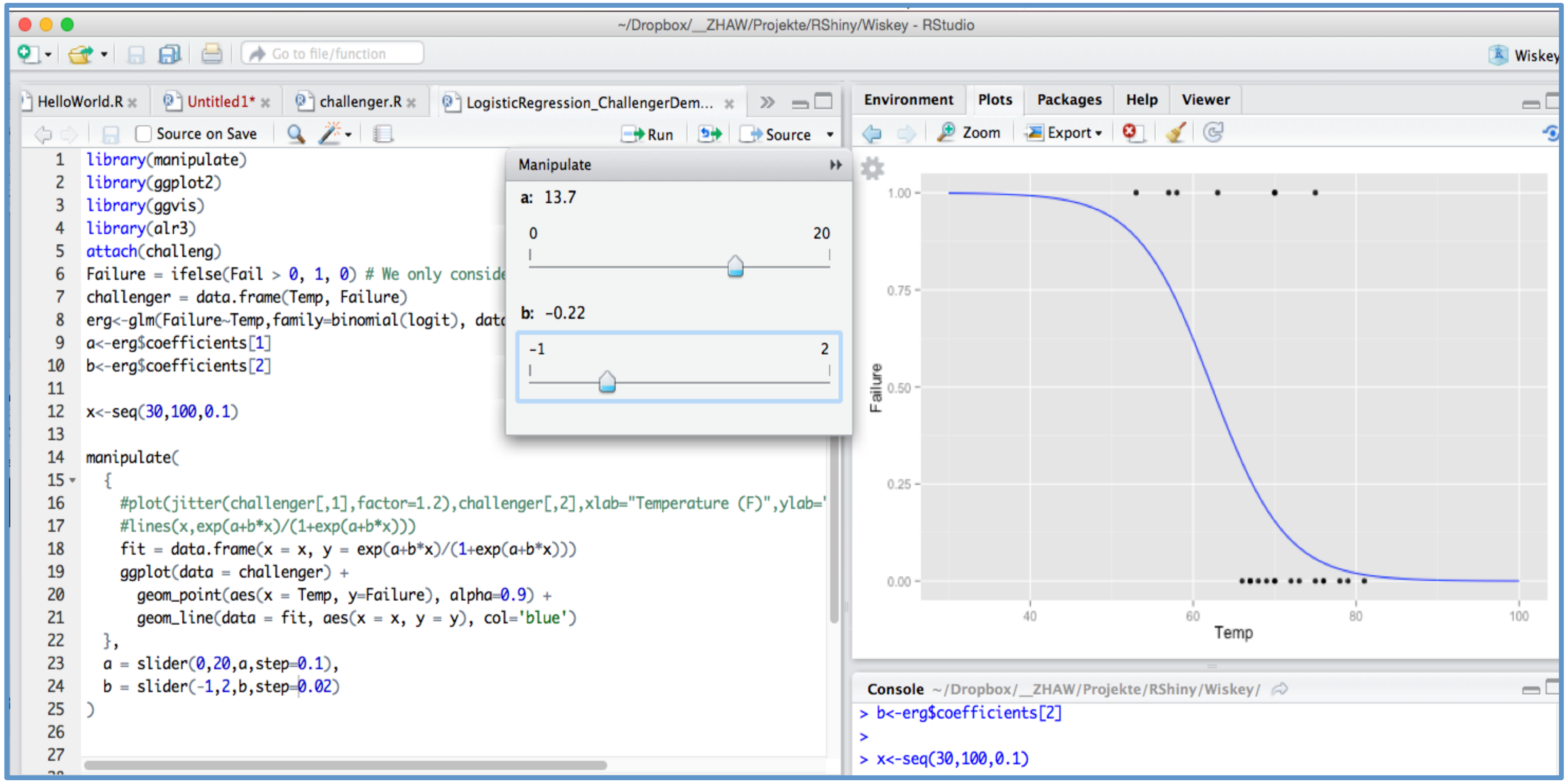
statistical modell (logistic regression)

$$P(Y=1 \mid X=x) = (1+e^{-(\beta_1 \cdot x+\beta_0)})^{-1} = (1+e^{-z})^{-1} = f(x)$$

$$z \in]-\infty,\infty[\rightarrow [0,1]$$

How do we determine the parameters β of the model? M(β)

Determination to the parameters



Live Demo with RStudio

Maximum Likelihood
(one of the most beautiful ideas in statistics)

$$M(\beta) \xrightarrow{\text{Likelihood / “probability“ (often known)}} \text{Data}$$

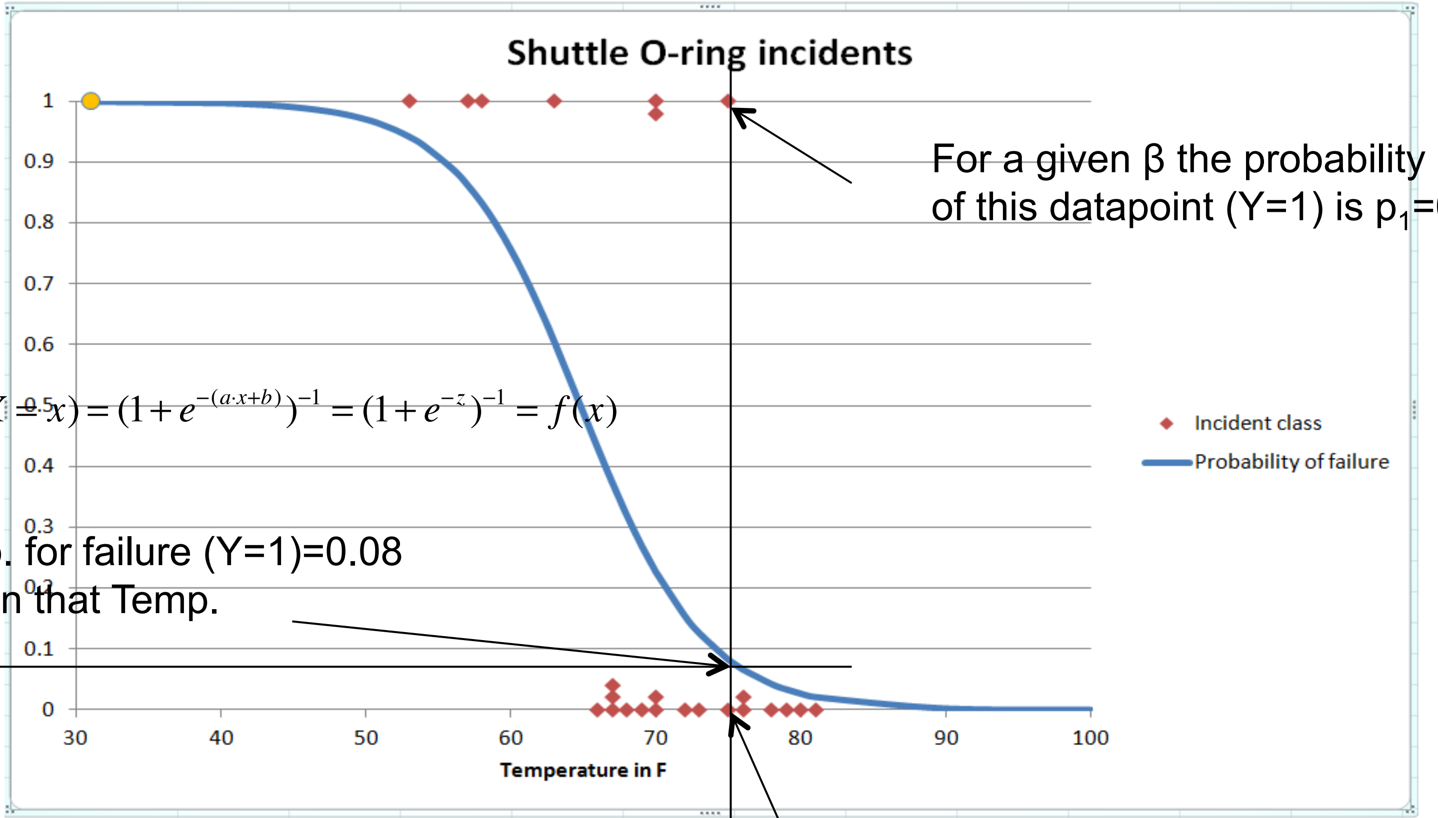
Tune the parameter(s) β of the model M so that (observed) data is most likely

What‘s the likelihood of the data for log. regression...

Ableitung Likelihood Funktion Tafel

Likelihood: Probability of a single observation

Two data points Y=1 (failure) and Y=0 (OK)



For a given β the probability of this datapoint (Y=1) is p_1=0.08

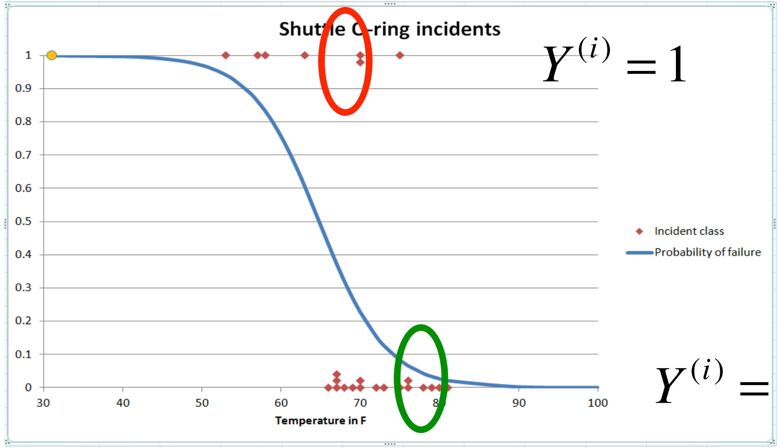
$$P(Y=1 \mid X=x) = (1+e^{-(a\cdot x+b)})^{-1} = (1+e^{-z})^{-1} = f(x)$$

Prob. for failure (Y=1)=0.08
Given that Temp.

For a given β the probability of this datapoint (Y=0) is
1 - 0.08 = 92%

Prob. of all data points is the product of the individual data points, (if iid).

Likelihood: Probability of the training set



$Y^{(i)} = 1$

$Y^{(i)} = 0$

Training Data i = 1...N

$X^{(i)}, Y^{(i)}$

$$p_1(X) = P(Y = 1 \mid X) = (1 + e^{-(a \cdot x + b)})^{-1} = (1 + e^{-z})^{-1} = f(x)$$

Probability to find Y=1 for a given values X (single data point) and a, b

$p_0(X) = 1 - p_1(X)$ Probability to find Y=0 for a given value X (single data point)

Likelihood (probability⁺ of the training set given the parameters)

$$L(\beta_0, \beta_1) = \prod_{i \in All\ ones} p_1(x^{(i)}) \; * \prod_{i \notin All\ Zeros} p_0(x^{(j)})$$

Let‘s maximize this probability

Maximizing the Likelihood

Likelihood (prob of a given training set) want to maximized wrt. parameters

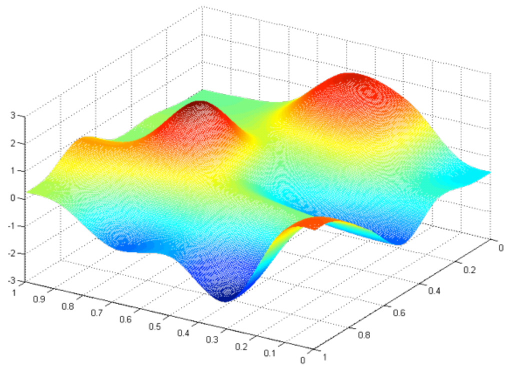
$$L(\beta_0,\beta_1) = \prod_{i \in All\ ones} p_1(x^{(i)}) \quad * \prod_{i \notin All\ Zeros} p_0(x^{(j)})$$

Taking log (maximum of log is at same position)

$$-nJ(\beta) = L(\beta) = L(\beta_0,\beta_1) = \sum_{i \in All\ ones} \log(p_1(x^{(i)})) \quad + \sum_{i \in All\ zeros} \log(p_0(x^{(i)})) = \sum_{i \in All\ Training} y_i \log(p_1(x^{(i)})) + (1-y_i)\log(p_0(x^{(i)}))$$

Gradient Descent for Minimum of J

$$\beta_i'' \leftarrow \beta_i' - \alpha \left. \frac{\partial J(\beta)}{\partial \beta_i} \right|_{\beta_i = \beta_i'}$$



Generalization

Verallgemeinerung

Mehr als eine Variable

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots \beta x_p$$

$$p(\vec{x}) = \left[1 + e^{-(x_0 \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p)}\right]^{-1}$$

Oft schreibt man auch als Vektor)

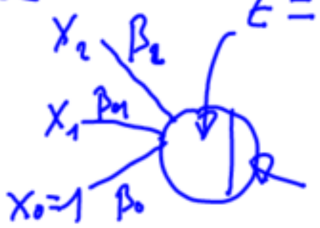
$$\vec{x} = (x_0, x_1, \ldots, x_p)'$$

$$\vec{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)'$$

dann $p(\vec{x}) = \left[1 + e^{-\vec{x}\vec{\beta}}\right]^{-1}$

Symbolisch

$$z = \vec{x} \cdot \vec{\beta}$$



$$\hat{y} = \mathrm{logit}(\vec{x} \cdot \vec{\beta})$$

Interpretation

Interpretation (Nach dieser Variable)

$$\frac{p(X)}{1-p(X)} = \frac{\text{Wahrs. das Ereignis stattfindet}}{\text{W'lkt das es nicht stattfindet}}$$

odds: Pferderennen

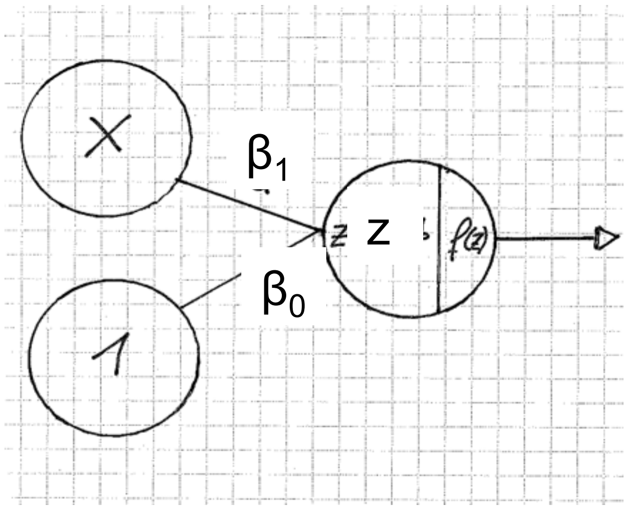
$$= e^{\beta_0 + \beta_1 X_1}$$

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots \beta_p x_p$$

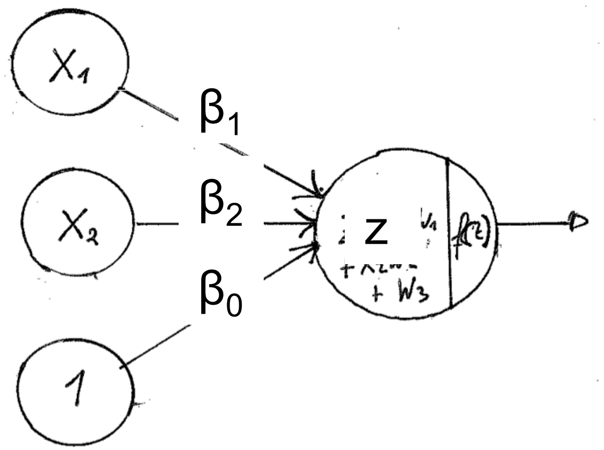
Je größer die Variable β_i desto größer ist der Effekt auf den log-odds

Logistic Regression the mother of all neural networks

1-D log Regression



Multivariate Log.-Regression



$$z = \beta_0 + \beta_1 x$$

$$z = \beta_0 + x_1\beta_1 + x_2\beta_2 = \beta^T x$$

$$p_1(x) = P(Y = 1 \mid X = x) = [1 + \exp(-\beta^T x)]^{-1} = \frac{\exp(\beta^T x)}{1 + \exp(\beta^T x)} = f(\beta^T x)$$

Logistic Regression in R (learning)

- Logistic regression in R using generalized linear models glm. Syntax like lm but need parameter **family=binomial.**
- default is factor

```
fit = glm(default ~ balance + income + student,
    family=binomial(logit), #Binomial for log-regression
    data = Default)
summary(fit)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
income       3.033e-06  8.203e-06   0.370  0.71152
studentYes  -6.468e-01  2.363e-01  -2.738  0.00619 **

## Alternatively using
library(nnet)
fit.mm = multinom(default ~ balance + income + student, data=Default)
```

Logistic Regression in R (prediction)

- Logistic regression in R using generalized linear models glm. Use type='response'

```
> #Balance sind Schulden!
> df = data.frame(balance=c(0,3000), income=c(1000,2000),
student=c("No", "Yes"))
> predict(fit, df, type='response')
           1            2
1.909602e-05 9.966644e-01
```

Result is the probability to belong to class 1 which is default = yes

Feature engineering: Categorical Features

- Example green, blue, red how to code?

```
> ############
> # Kategorical Variables
> y = c(0,0,0,1,1,1)
> x = c(0,1,2,0,1,2)
> fit = glm(y ~ x)
> model.matrix(fit)
  (Intercept) x
1           1 0
2           1 1
3           1 2
4           1 0
5           1 1
6           1 2
attr(,"assign")
[1] 0 1
>
> fit = glm(y ~ as.factor(x))
> model.matrix(fit)
  (Intercept) as.factor(x)1 as.factor(x)2
1           1             0             0
2           1             1             0
3           1             0             1
4           1             0             0
5           1             1             0
6           1             0             1
```

Normalisierung / Scaling

- Unterschiedliche Werte Bereiche
- Daten können Einheiten tragen

Person	Körper Gewicht [kg]	Hirngewicht [g]	Schuhgrösse	Körper Länge [cm]
1	75.1	1400	42	192
2	84.9	2029	47	189
...	...	...	...	
150	50	1780	39	173

- Beliebte Normierungen:
  – Z-Normierung: Danach einheitenlos, MW = 0, stddev = 1 (R: scale)
  – Quantil-Normalisierung: Alle Quantile der Verteilung gleich