Hochschule Konstanz
Fakultät Informatik

Prof. Dr. Oliver Dürr

KI Vorlesung: Machine Learning(II)

Remark: Much of the material have been developed together with Beate Sick (ZHAW/Uni Zurich).

Learning Objective

- Know what is overfitting
- Know cross-validation
- Understand logistic regression
- Understand the loss function for logistic regression
- Know who scale data and handle categorical data

Question

- Is linear regression good for my data

- Alternatives to linear regression:
  - Neural Networks (see next week)
  - Deep Learning (in 2 weeks)
  - Polynomial regression (next slide)

Non Linear Effect / Polynomial Data

- Want:
  - $\hat{y} = w_0 + w_1 \cdot x + w_2 \cdot x^2 + \cdots + w_p \cdot x^p$

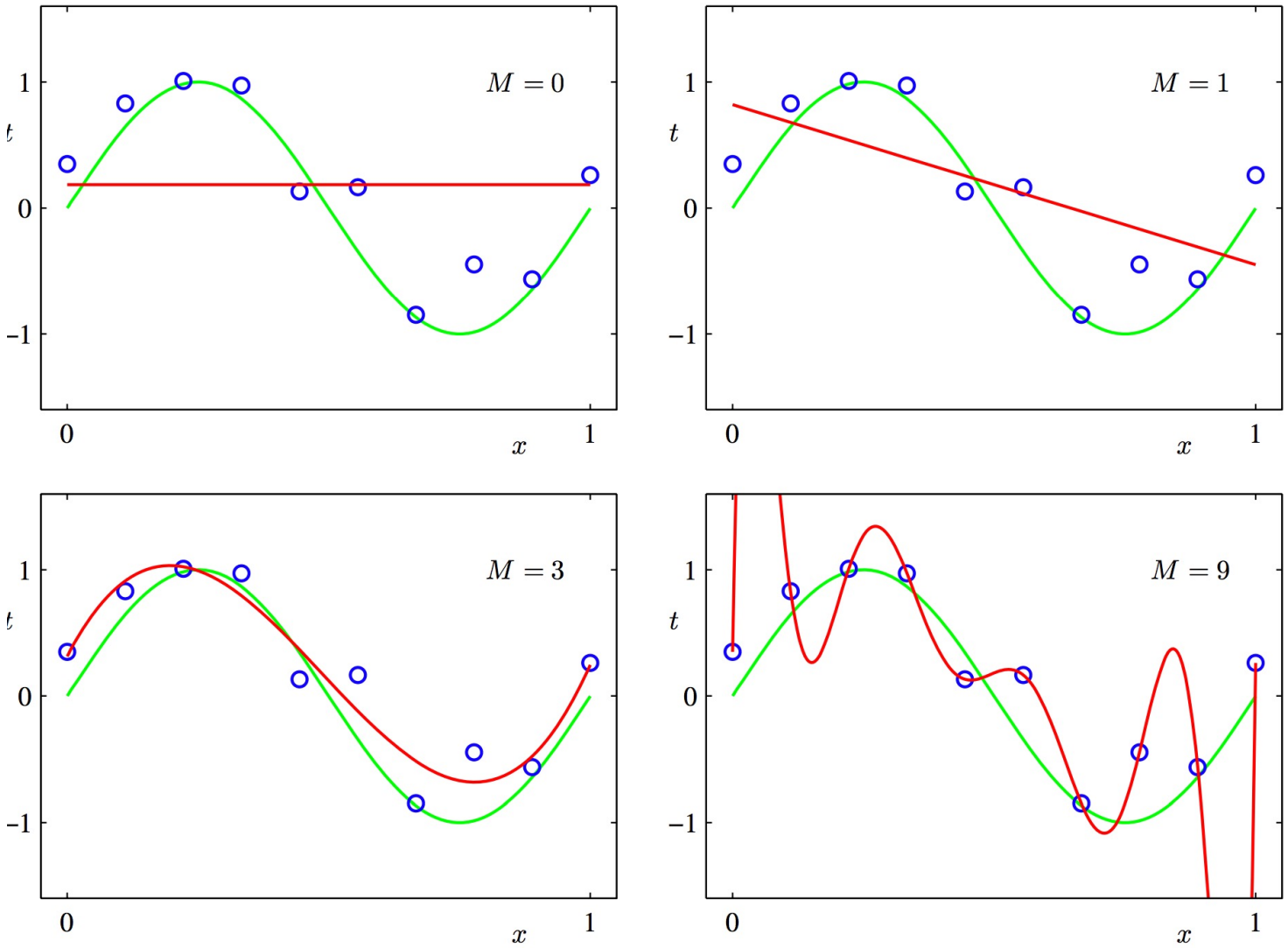
- Copy columns in design matrix **X** say $p = 3$

Intercept	x
1	2
1	3
…	
1	1

Intercept	x	x^2	x^3
1	2	4	8
1	3	9	27
…			
1	1	1	1

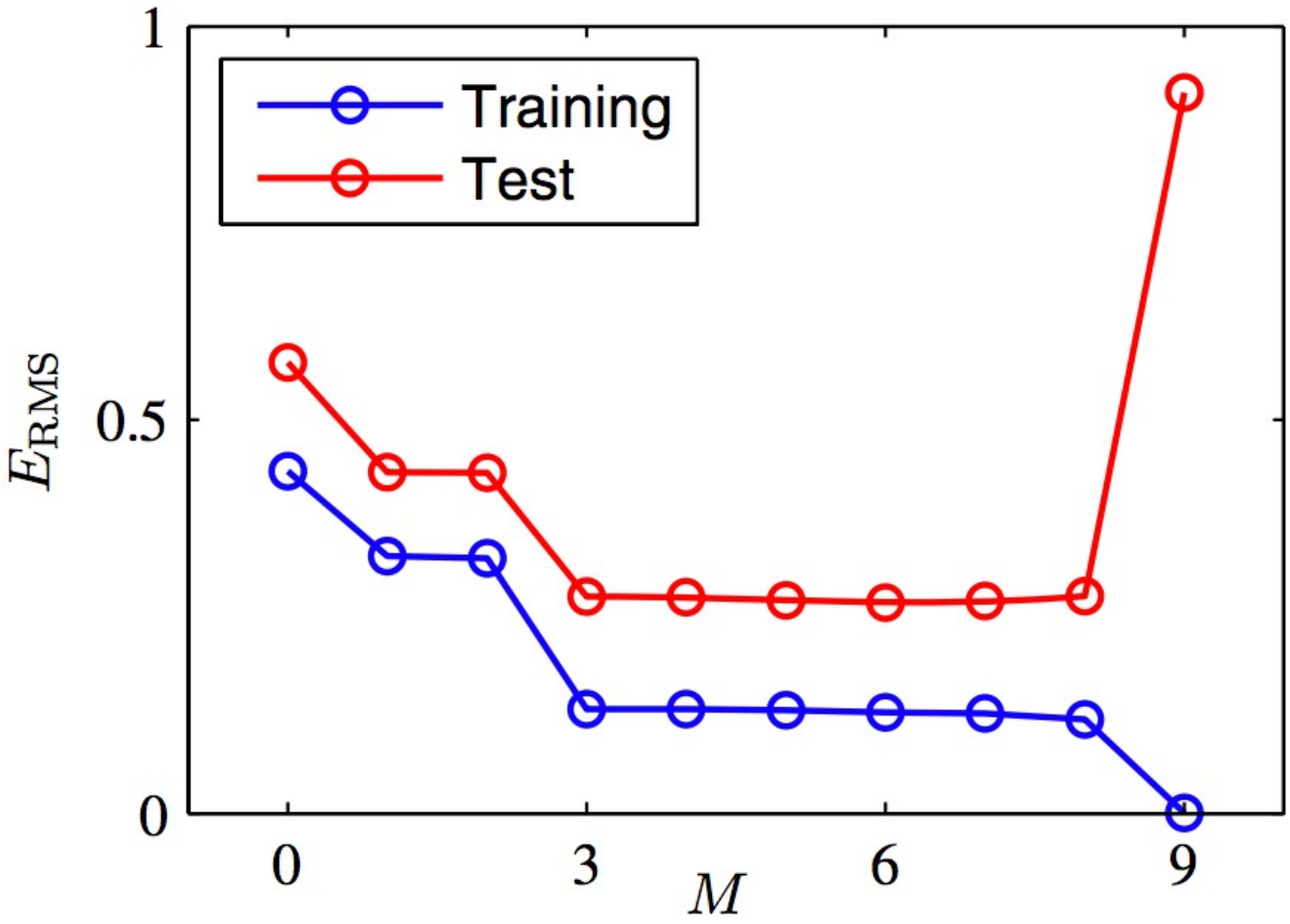
$$\hat{y} = X \cdot w^T$$

Overfitting: Data Generating Process



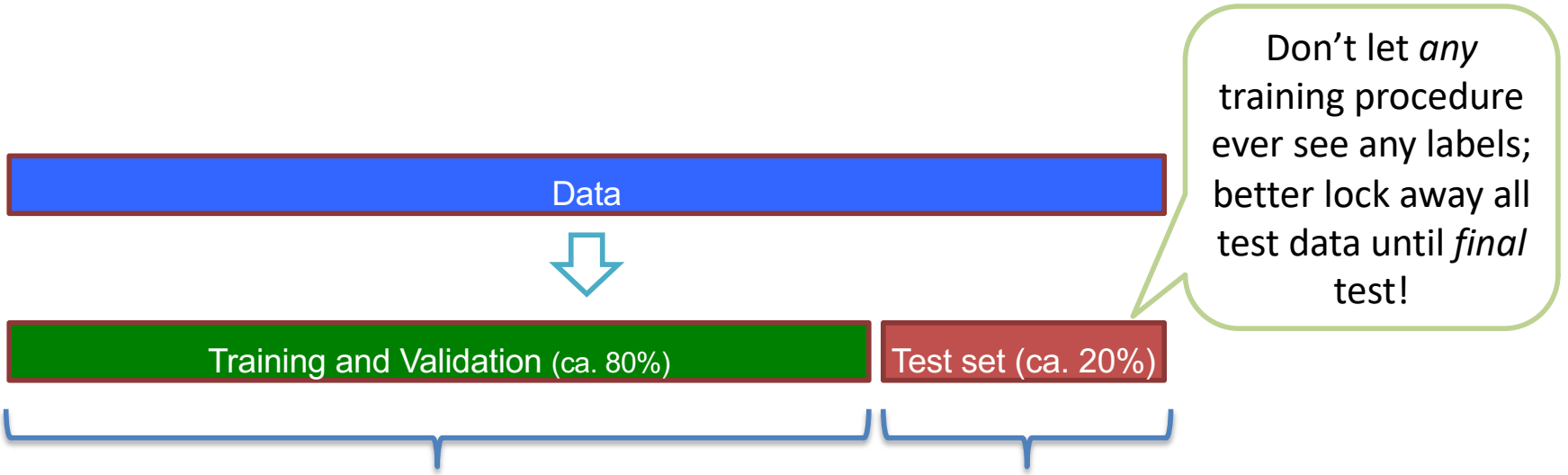
Expand the data matrix to columns $x^0, x^1 \ldots x^M$

Overfitting



How to be on the safe side

Typical strategy, spare some data for the final testing.

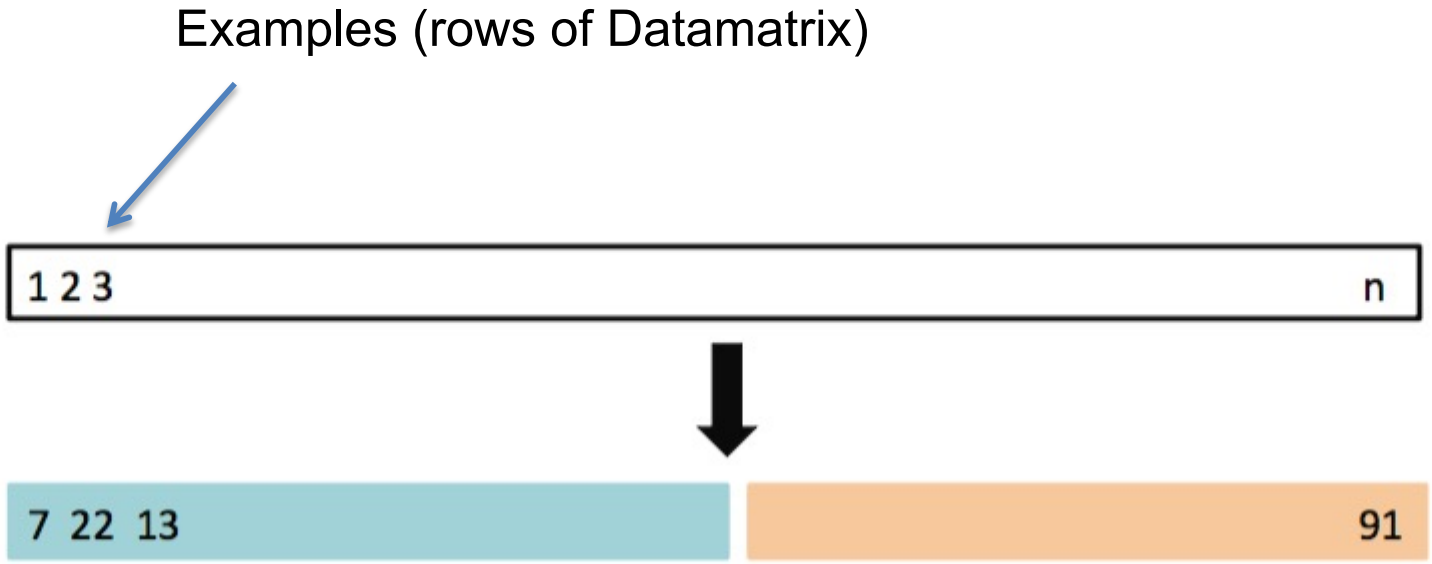


Use this to train model, select models and hyperparameters

Use this to estimate the performance.

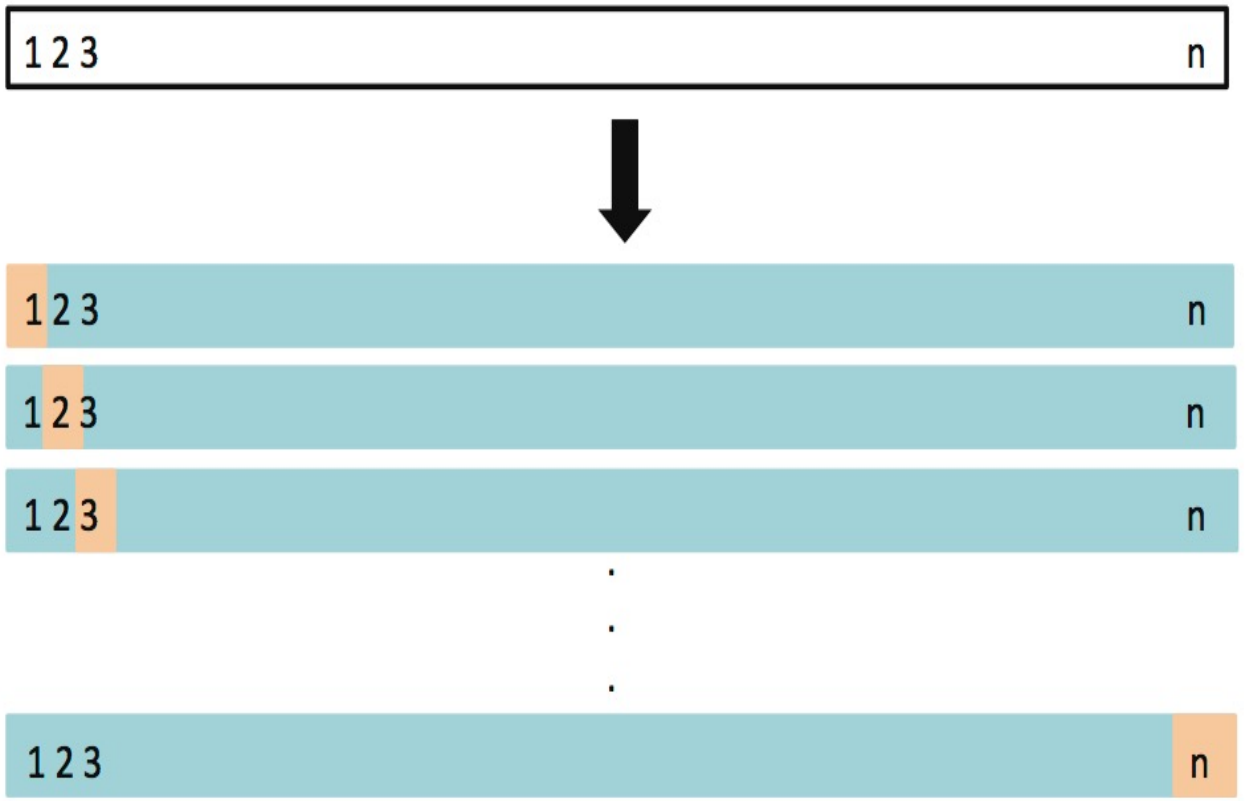
In the following, we are only talking about Training and Validation set.

The Validation Set Approach



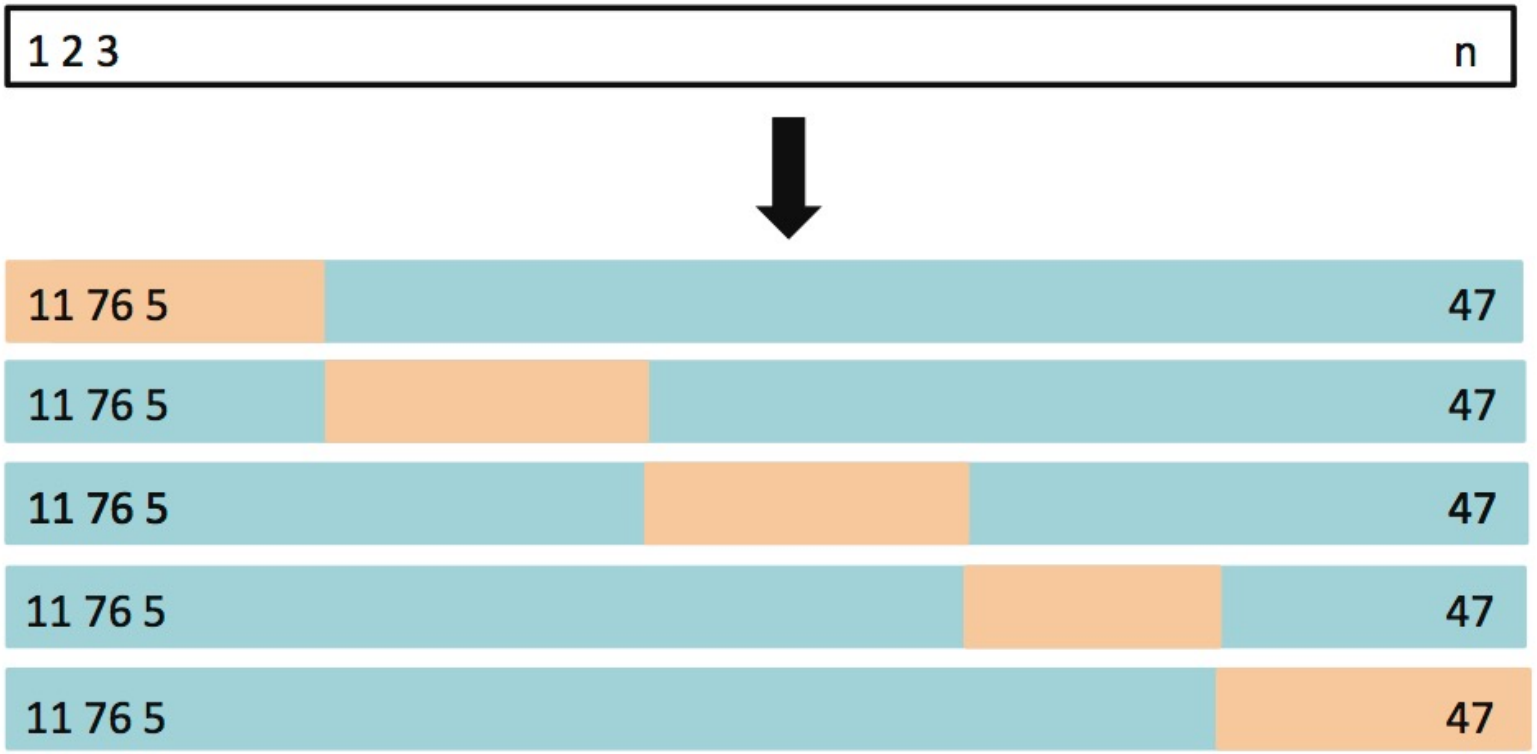
A random splitting into two halves: left part is training set, right part is validation set

Leave-One-Out Cross Validation (LOOCV)



Fit w/o red sample and predict the red sample. Average over all n repeats

K-fold Cross Validation



Fit w/o orange samples and predict the red samples. Average over all k repeats. Do a weighted average if folds do not have the same size.

Question: What happens if k=n, what if k=2?

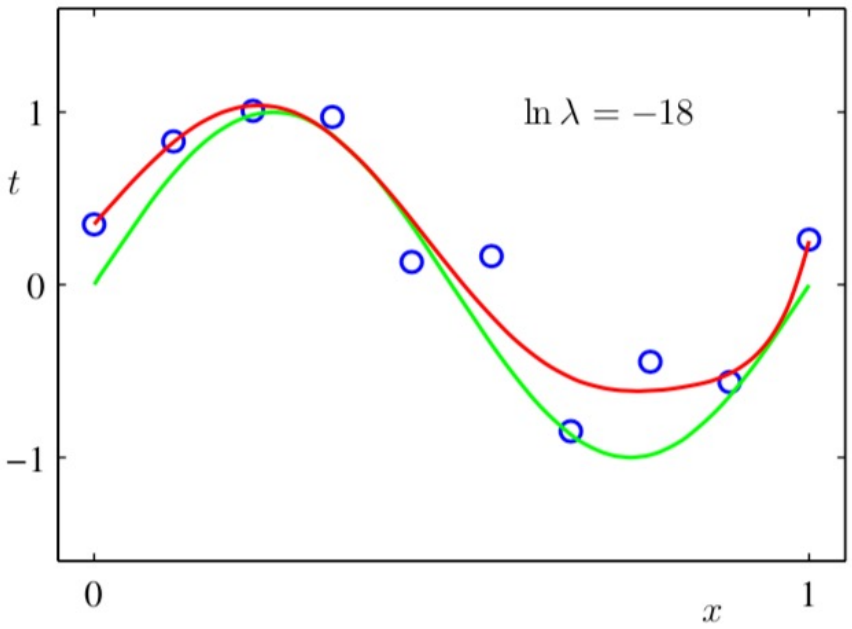
Regularization

How to avoid overfitting

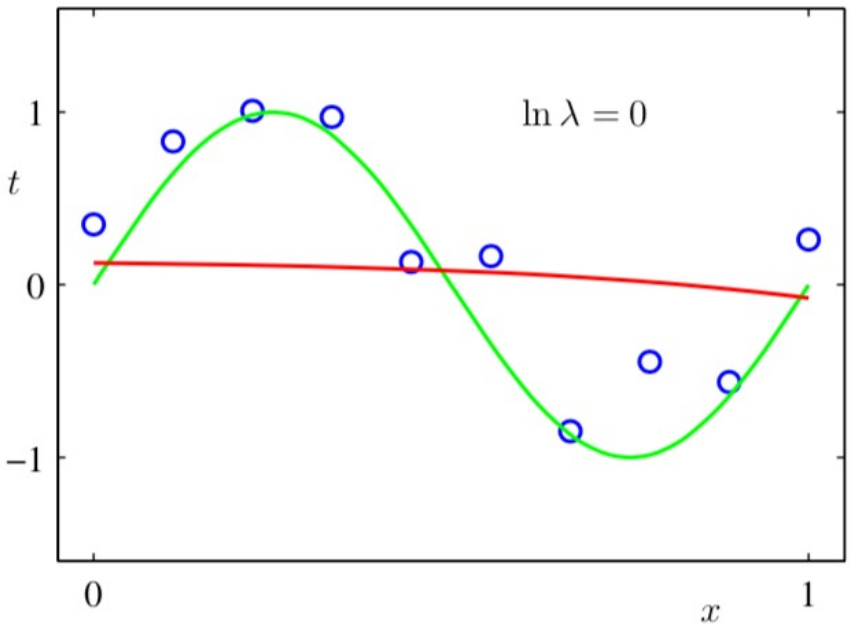
- Regression example
  - Simple models small M
  - Don’t use all features
- Penalty for too complex features
  - Ridge Regression (next slide)
- Many novel methods in Deep Learning
  - Weight Decay (ridge regression in disguise)
  - Training Data Augmentation

Ridge Regression

$\lambda = 0.000000015$



$\lambda = 1$



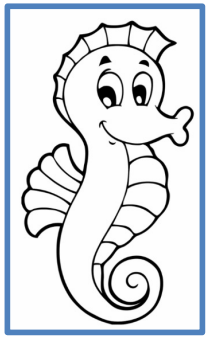
$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = \mathrm{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2,$$

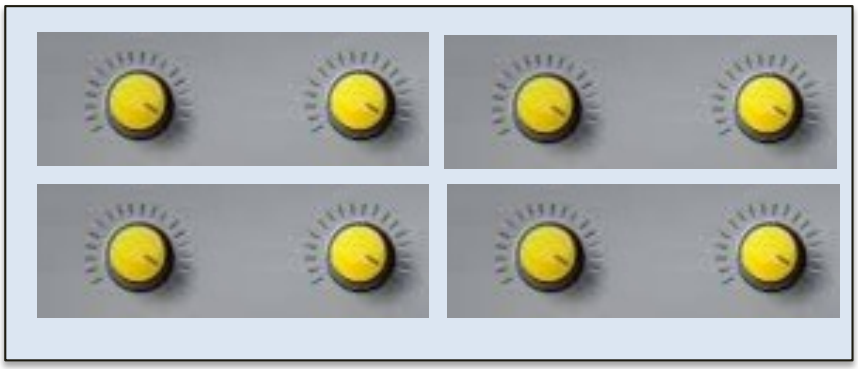
Additional loss term L2-Regularisation

- It costs to have non-zero parameters.
- Parameters are closer to zero (*shrinkage*)
- Ridge Regression reduces overfitting.
- Allows to fit linear regression to #parameters > # Data problems
- Also popular Lasso L1 regularisation

Classification

Recap Supervised Learning: Training (Image Classification)

x	True Class y		Predicted class $\hat{y}$
	Tiger	→	Seal 👎
	Tiger	→	Tiger 👍
	Seah orse	→	Seahorse 👍



...

Typical 1 Mio. Trainingsdata

Training Principle:

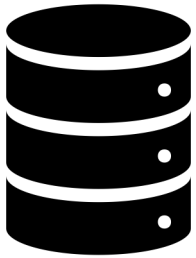
Parameter are adjusted so that training error is minimal.

Logistic Regression [motivation]

- Kreditscoring & logistische Regression

schufa

67,7 Mio. Personen und
6 Mio. Unternehmen



Anzahl und Art der Kreditaktivitäten

Zahlungsausfälle

Erfahrungen im Umgang mit Kreditgeschäften

„Das verwendete Verfahren wird als „logistische Regression“ bezeichnet und ist eine fundierte, seit langem praxiserprobte, mathematisch-statistische Methode zur Prognose von Risikowahrscheinlichkeiten.“

Source: Constanze Riedinger presentation Data Analytics / Auch hier

Logistic Regression

- Extends idea of linear regression to situation where outcome variable is categorical

- We focus on binary classification

  i.e. Y=0 or Y=1

- Is a probabilistic method, gives the probability for class Y=1

Logistic Regression [motivation, cont'd]

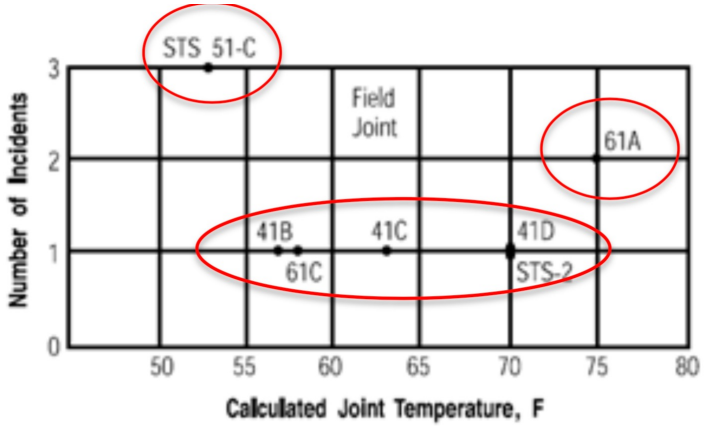
Some Background on probabilistic modelling

The challenger space shuttle exploded 73 seconds after the start in 1986. One of bearings in the booster has been broken.

Statistik & Challenger Desaster [side track]

- On the day of the challenger launch it was cold: 31°F.
- In 7 from 23 flights there have been problems with the booster bearings

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



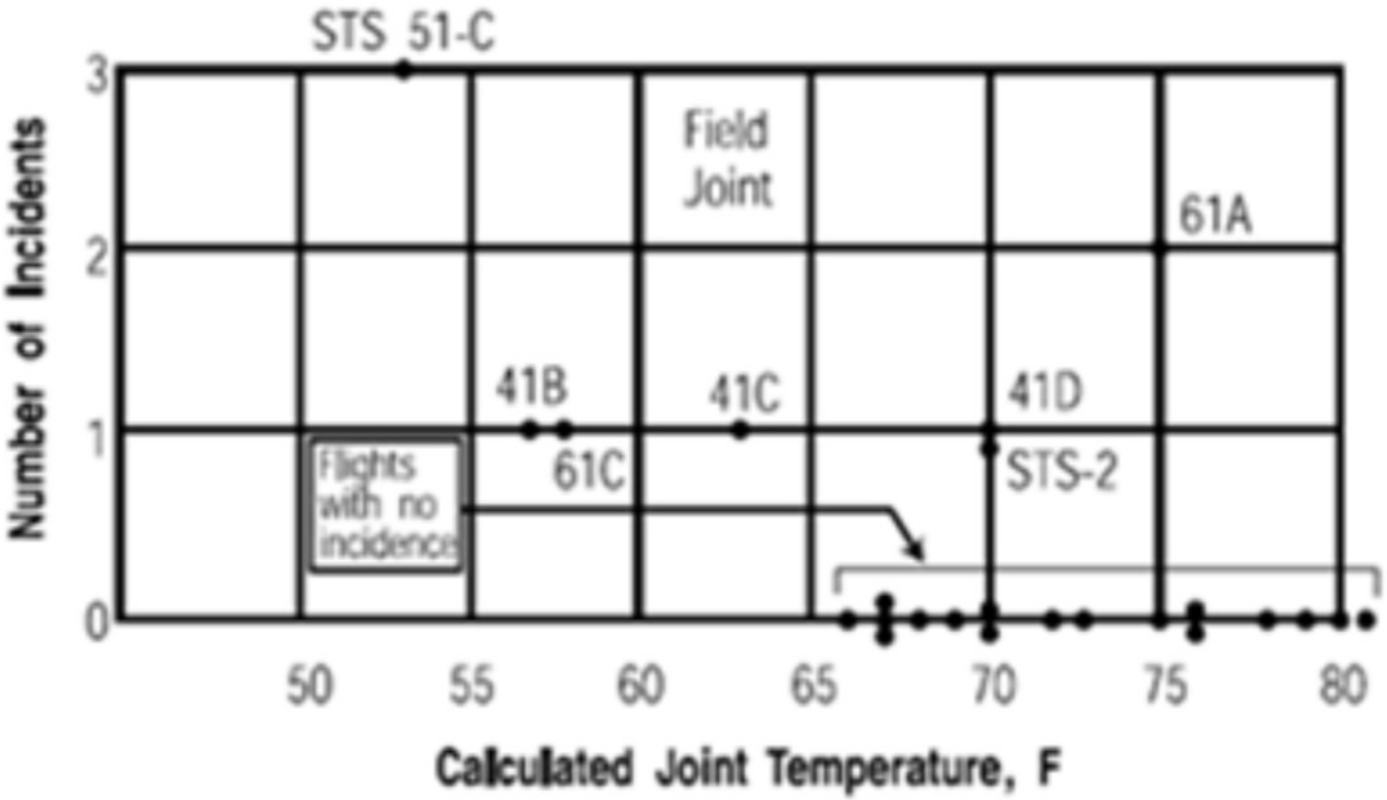
Plot of flights with incidents of O-ring thermal distress as function of temperature.

Figures from: [PRESIDENTIAL COMMISSION on the Space Shuttle Challenger Accident](https://history.nasa.gov/rogersrep/v4part3.htm) (https://history.nasa.gov/rogersrep/v4part3.htm)

- Is there an increased risk of failure at low temperatures?
  - NASA Engineer: „...I can't get a correlation between O-ring erosion, blow-by an O-ring, and temperature. “
- Would you launch (give reasons)?

Statistik & Challenger Desaster [side track]

- There is information in the successful flights



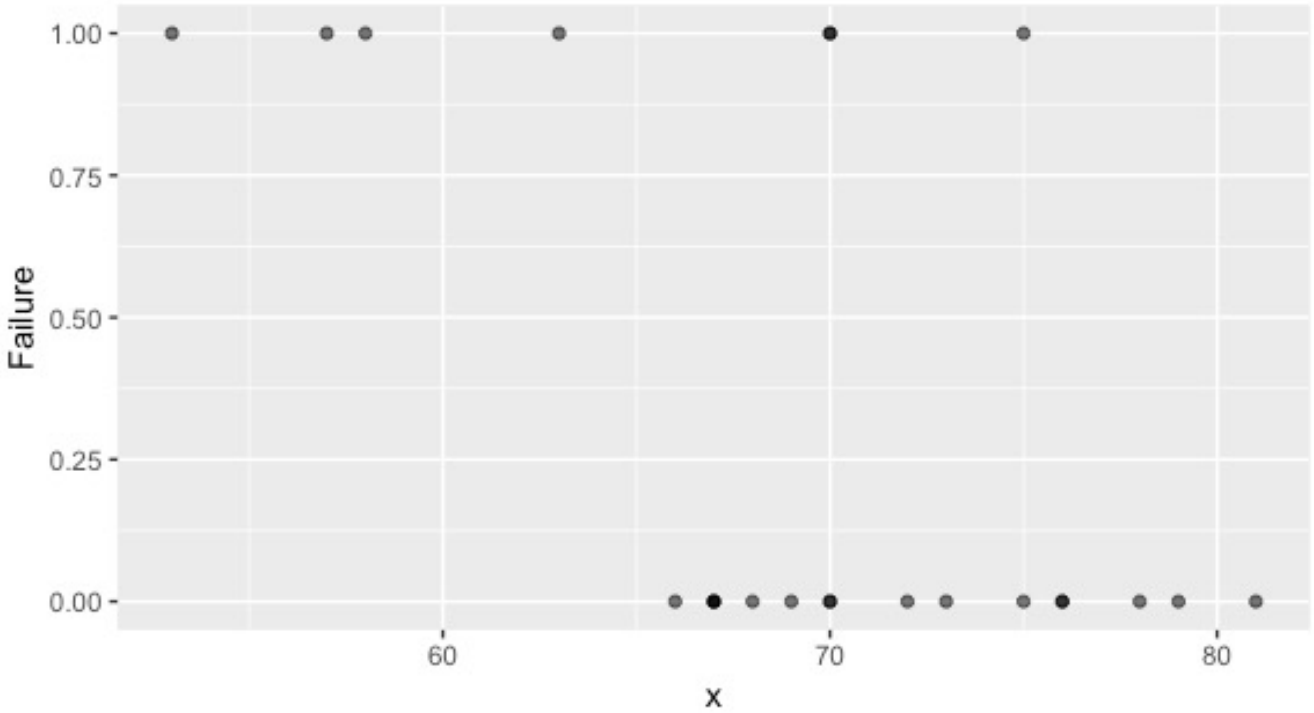
Plot of flights with and without incidents of O-ring thermal distress.

Modelling with logistic regression

Binarize to a zero / one classification

Want: $p(x) = Pr(Y = 1|x)$

Prob. for a O-ring to be defect Y=1 at a given temperature X



Question:

- **Guess curve?**
- **Why is p(X) = b +a x (linear regression) wrong?**

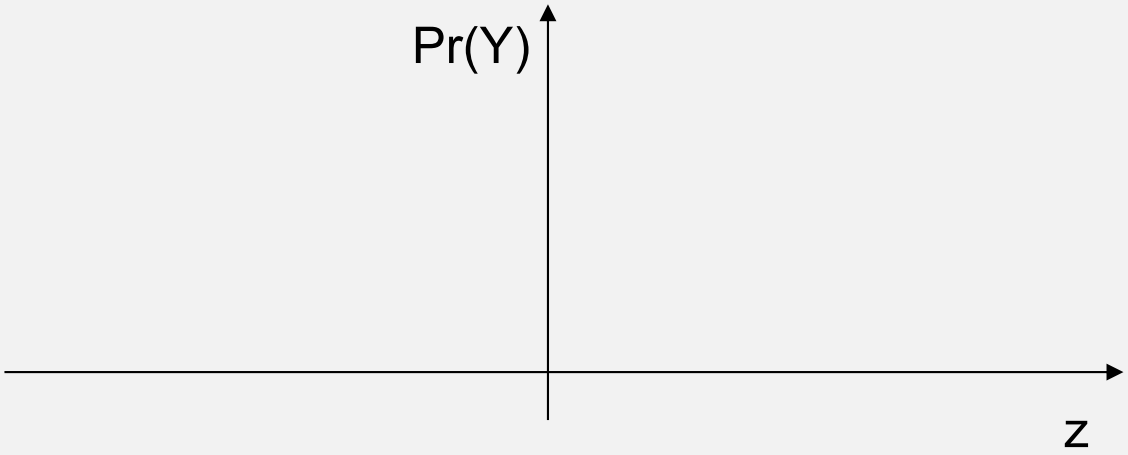
Find a suitable squeezing function

- Idea of logistic regression
  - –Take output of linear regression

$$z = a \cdot x + b \qquad z \in [-\infty, \infty]$$

  - –and squeeze it to [0 and 1]

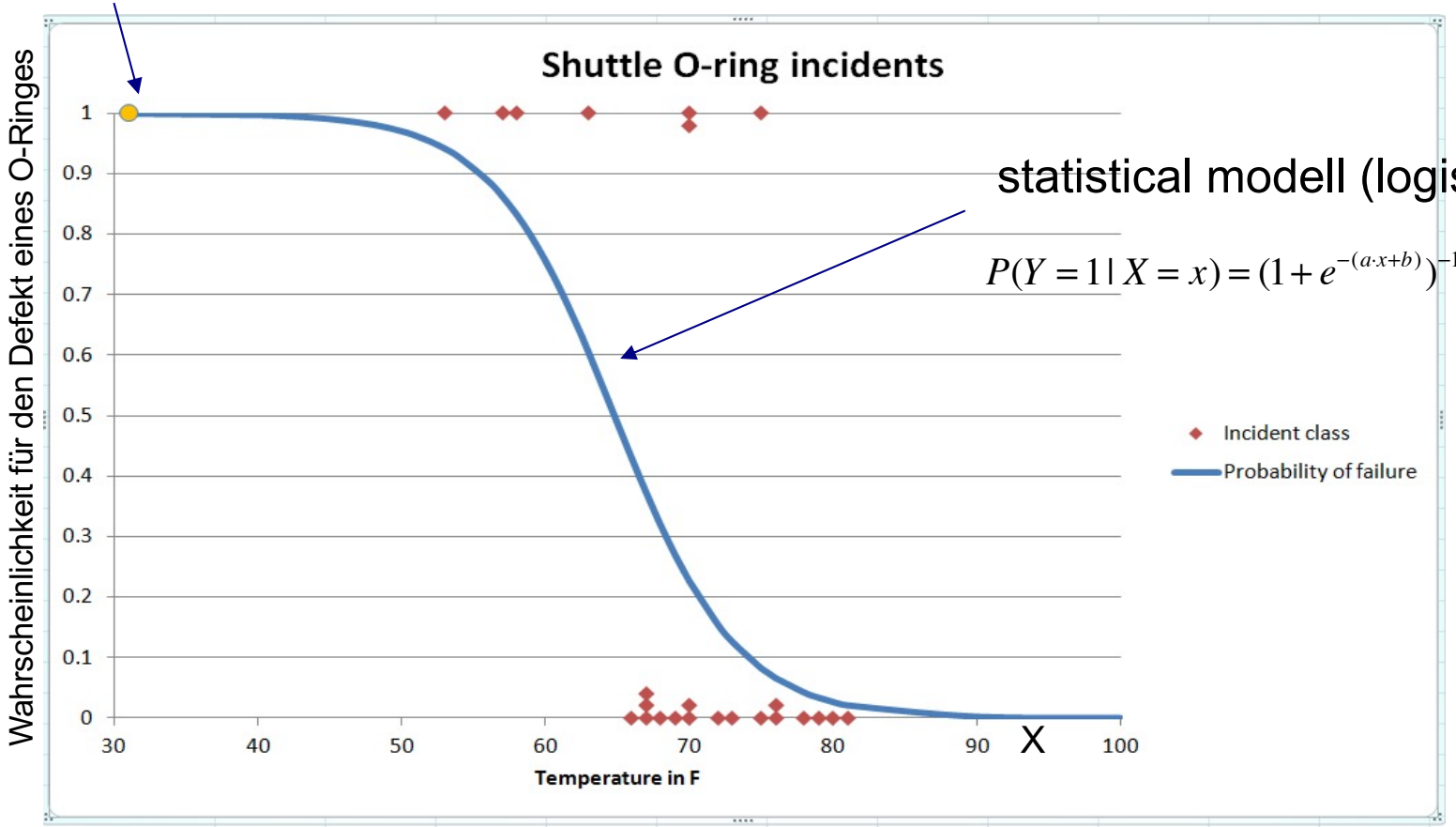
- **Task: Draw a function which could do that.**
  - Discuss with your neighbor



Logistic Regression

Predict if O-Ring is broken, depending on temperature

Challenger launch @31 F
Prob. of a failure=0.9997



statistical modell (logistic regression)

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

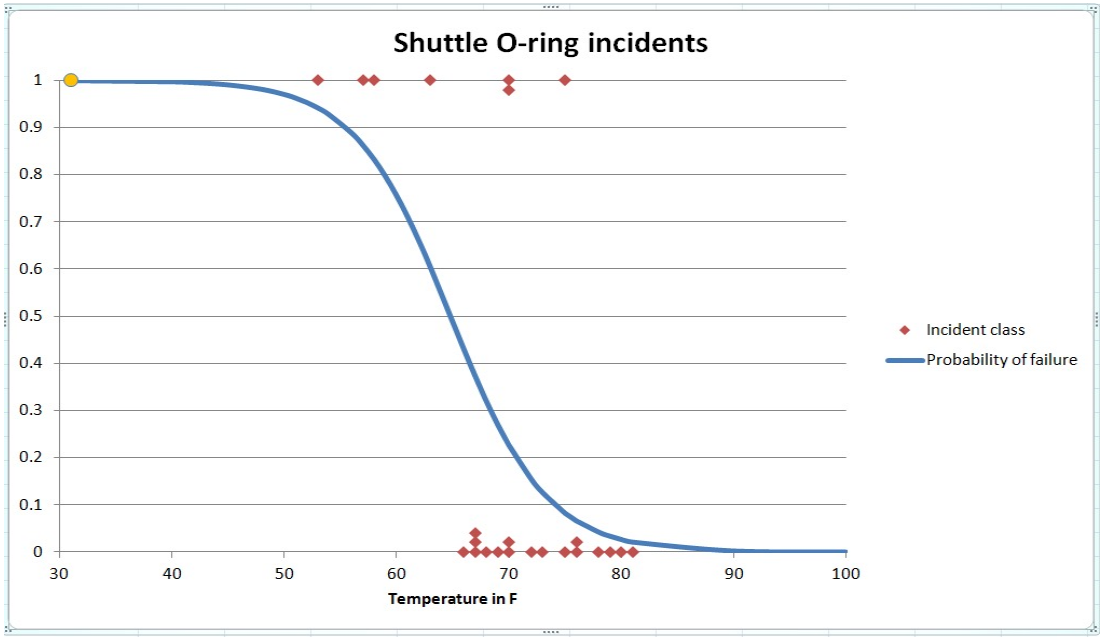
$$z \in]-\infty,\infty[\rightarrow [0,1]$$

Fitting of the parameters

- Linear Regression minimize RMSE
- RMSE in lineare regression can be seen as Maximum Likelihood Method
- Maximum Likelihood yields to

$$-NJ(\theta) = L(\theta) = L(a,b) = \sum_{i \in All\ ones} \log(p_1(x^{(i)})) + \sum_{i \in All\ zeros} \log(p_0(x^{(i)})) = \sum_{i \in All\ Training} y_i \log(p_1(x^{(i)})) + (1-y_i)\log(p_0(x^{(i)}))$$

- Optimization using an iterative approach



This curve depends on parameters *a,b.*

Choose them so that data is best fitted.

Logistic Regression as a neural network

The Single Cell: Biological Motivation

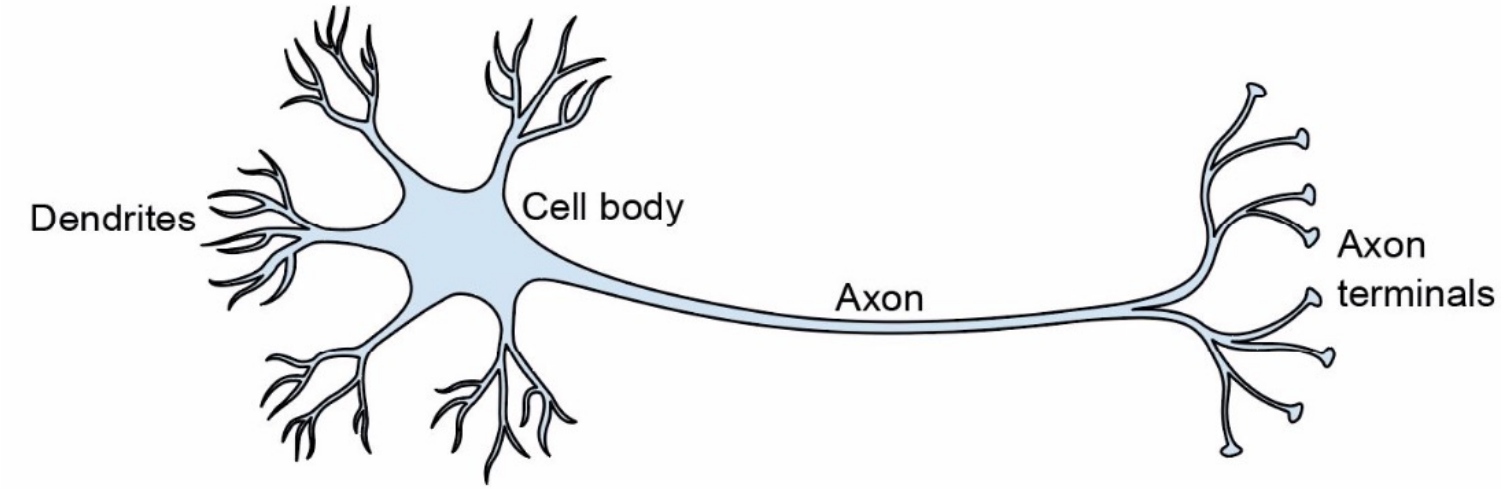


Figure 2.2 A single biological brain cell. The neuron receives the signal from other neurons via its dendrites shown on the left. If the cumulated signal exceeds a certain value, an impulse is sent via the axon to the axon terminals, which, in turn, couples to other neurons.

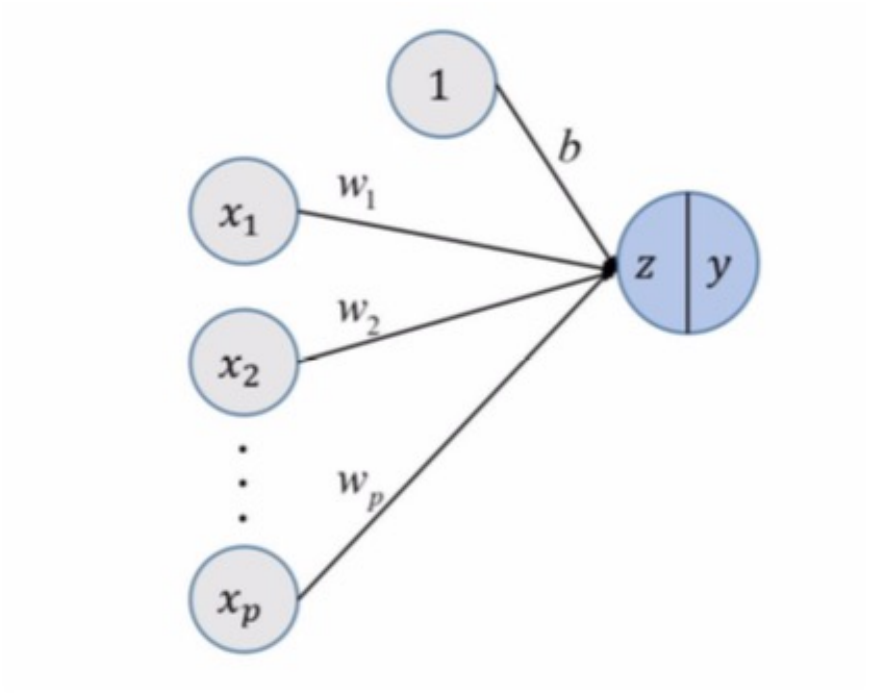
Neural networks are **loosely** inspired by how the brain works

The Single Cell: Mathematical Abstraction

$$z = b + x_1 \cdot w_1 + x_2 \cdot w_2 + \cdots x_p \cdot w_p$$

$$z = b + \sum x_i \cdot w_i = b + \boldsymbol{x} \cdot \boldsymbol{w}$$

Activation (many possibilities)



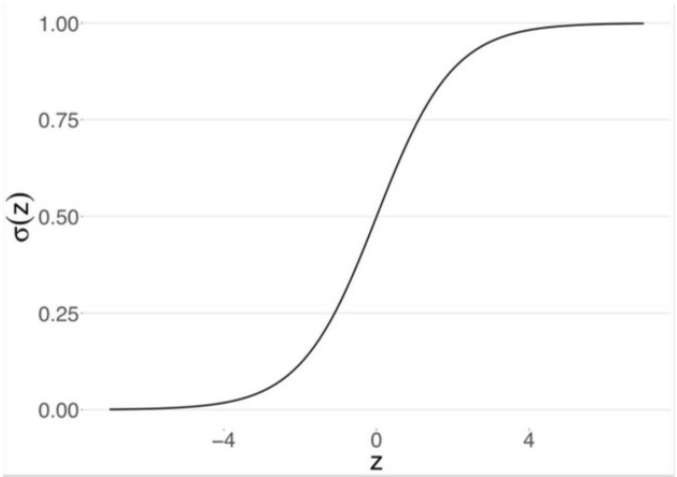
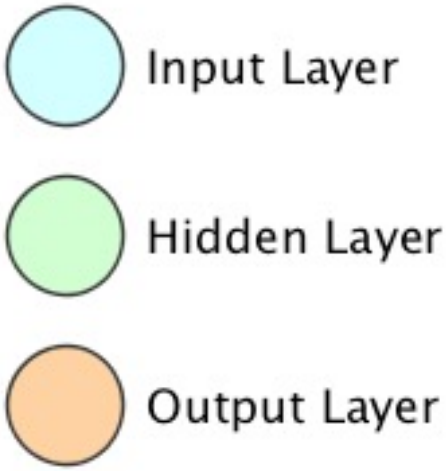


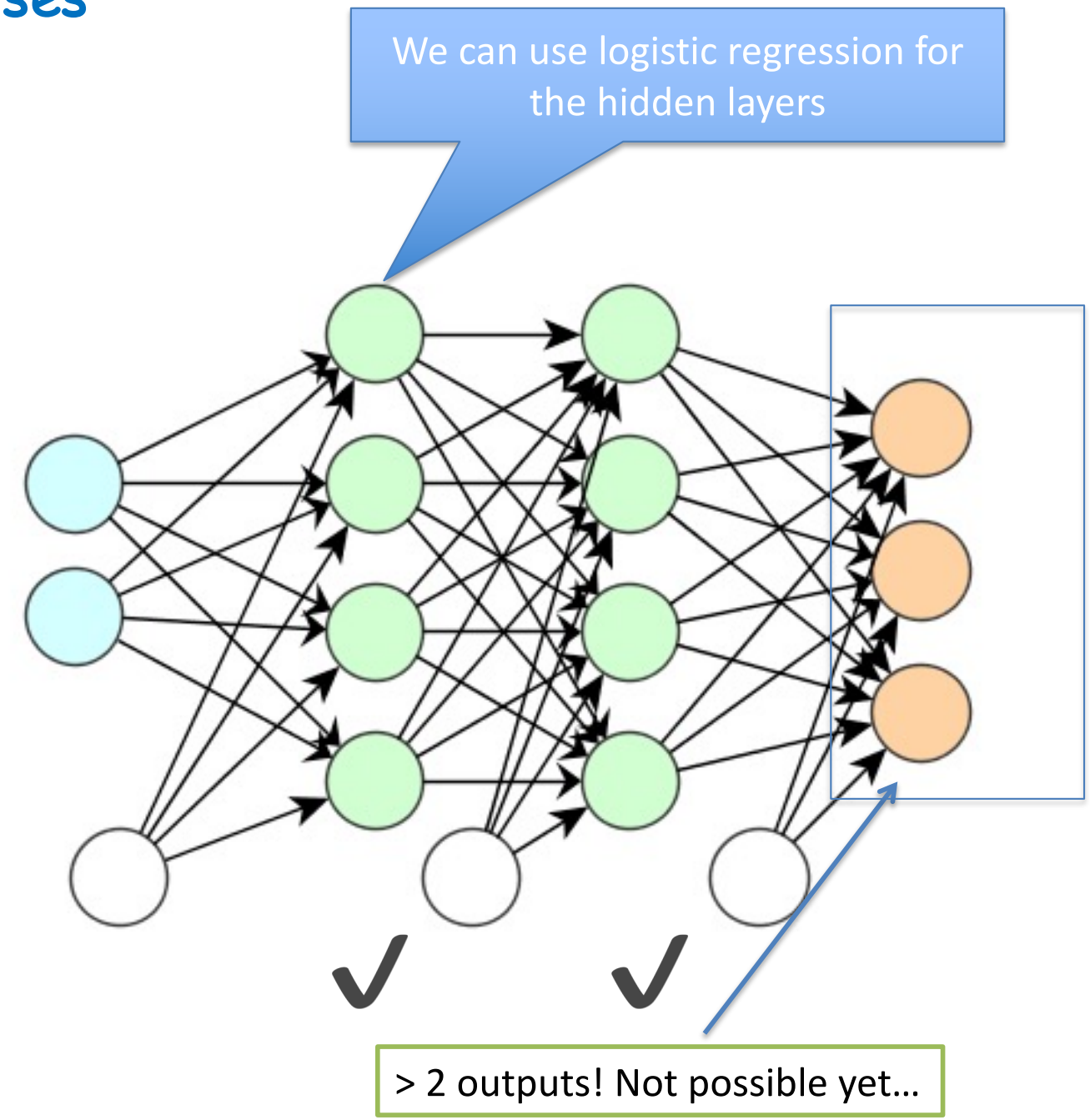
Figure 2.3 The mathematical abstraction of a brain cell (an artificial neuron). The value *z* is computed as the weighted sum of the *p* input values, *x*1 to *xp*, and a bias term *b* that shifts up or down the resulting weighted sum of the inputs. The value *y* is computed from *z* by applying an activation function.

```
# definition of the sigmoid function
def sigmoid(z):
    return (1 / (1 + np.exp(-z)))
```

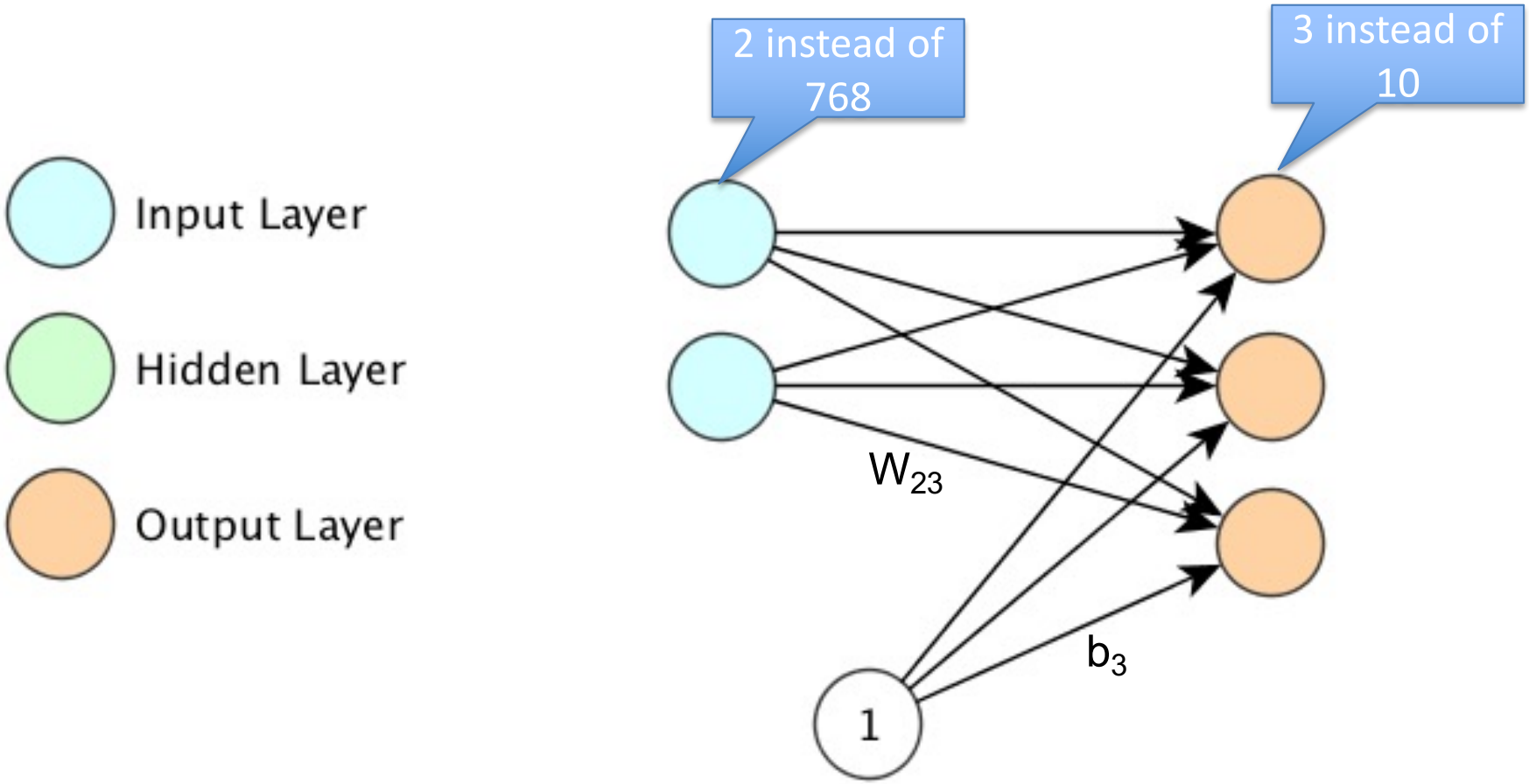
More than two classes

More than two classes



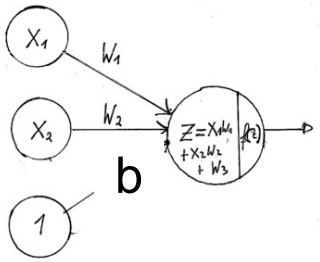


Multinomial Logistic Regression



Blackboard Example MNIST

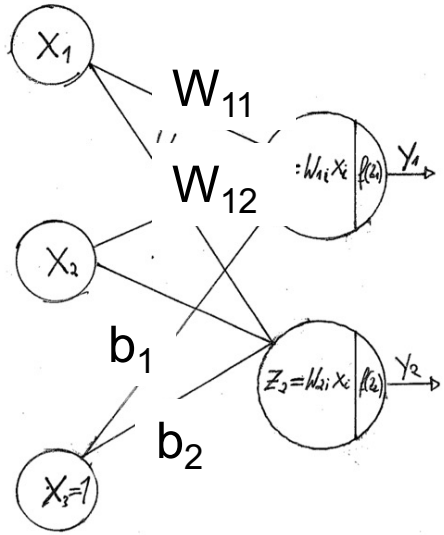
Multinominal Regression



Binary Case

$$P(Y=1 \mid X=x) = \frac{1}{1+\exp(-z)} = \frac{\exp(\sum_i x_i W_i)}{1+\exp(\sum_i x_i W_i)} \propto \exp(\sum_i x_i W_i)$$

W_{12} = reads „from node 2 to 1“



More than one class

called logit

$$p_1 = P(Y_1 = 1 \mid X = x) \propto \exp(\sum_i x_i W_{i1} + b_1) \qquad p_1 = \frac{\exp(\sum_i x_i W_{i1} + b_1)}{\sum_j \exp(\sum_i x_i W_{ij} + b_j)}$$

$$p_2 = P(Y_2 = 1 \mid X = x) \propto \exp(\sum_i x_i W_{i2} + b_2)$$

Normalisation

$$\sum_{i=1} p_i = 1$$

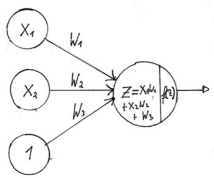
Multinomial case: just another **non-linearity *softmax***

$$p_1 = P(Y_1 = 1 \mid X = x) = \frac{\exp(\sum_i x_i W_{i1} + b_1)}{\sum_j \exp(\sum_i x_i W_{ij} + b_j)} = \text{softmax}(\sum_i x_i W_{i1} + b_1)$$

Loss for multinomial regression

This is the prob. the model evaluates for the true class $y^{(i)}$ of training example $x^{(i)}$

Training Examples Y=1 or Y=0



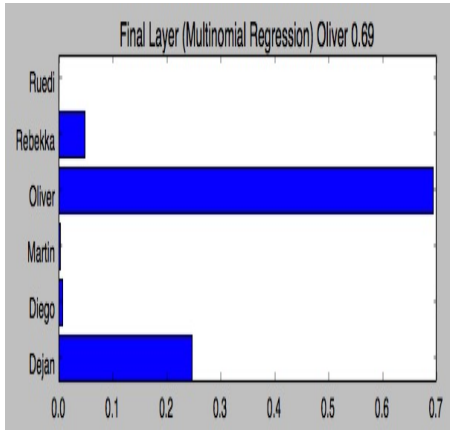
$$\text{loss} = -\frac{1}{N}\sum_{n=1}^{N}\log(p_{\text{model}}(y^{(i)} \mid x^{(i)};\theta))$$

$$\text{loss} = -\frac{1}{N}\sum_{n=1}^{N}\log(p_{\text{model}}(y^{(i)} \mid x^{(i)};\theta)) = -\frac{1}{N}\left(\sum_{i\in All\ ones}\log(p_1(x^{(i)})) + \sum_{i\in All\ zeros}\log(p_0(x^{(i)}))\right)$$

N Training Examples classes (1,2,3,…,K)

$$\text{loss} = -\frac{1}{N}\sum_{n=1}^{N}\log(p_{\text{model}}(y^{(i)} \mid x^{(i)};\theta)) = -\frac{1}{N}\left(\sum_{i\in y_j=1}\log(p_1(x^{(i)})) + \sum_{i\in y_j=2}\log(p_2(x^{(i)})) + \ldots + \sum_{i\in y_j=K}\log(p_K(x^{(i)}))\right)$$

p_i



Output of last layer

Example: Look at class of single training example. Say it's Dejan, if classified correctly p_dejan = 1 ➔ Loss = 0. Real bad classifier put's p_dejan=0 ➔ Loss = Inf.

Evaluation of classifiers

Movitation: Unbalanced data sets

	True Default: yes	True Default: non	Total
Pred Default: yes	81 (TP)	23(FP)	104
Pred Default: no	252 (FN)	9644 (TN)	9896
Total	333	9667	10000

- 333 defaulters, but only 81 are “found” by the classifier
  - True Positive Rate = Sensitivity $= \frac{\mathrm{TP}}{\mathrm{TP+FN}} = \frac{81}{333} \approx 24\%$ are found by the classifier
- On the other hand
  - True Negative Rate = Specify $= \frac{TN}{TN+FP} = \frac{9644}{9667} \approx 99.76\%$ of the one called non-default did not default

Log Score

- Accuracy has problems in unbalanced data
- For two classes there are special measures (AUC)
- In general
  - Just use the log-score
  - Mean of $-\log\left(p_{true}(x)\right)$ with $p_{true}(x)$ the probability of the true class

Practical Considerations

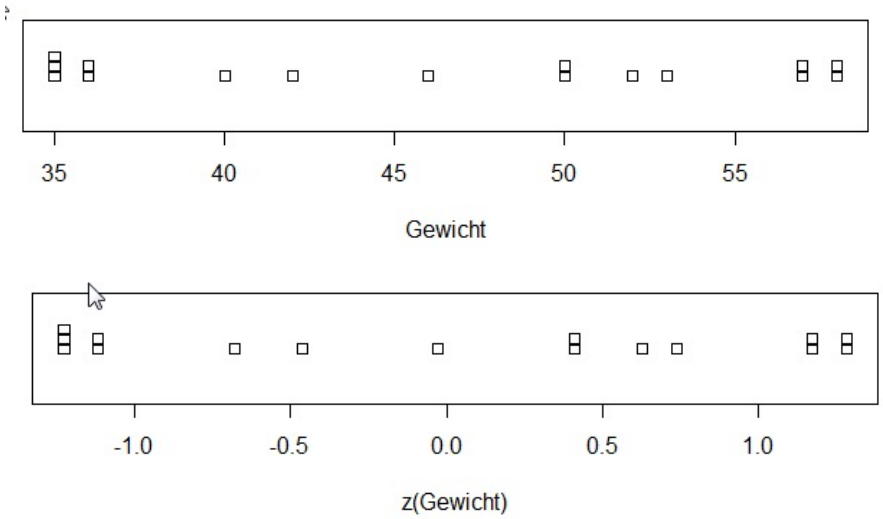
Scaling

- Logistic regression (and neural networks) depend on scale of X
- It's not good if different columns of X have different ranges
- Trick (z-transformation a.k.a. standard scaling)
  - Subtract mean and divide by standard deviation

$$x \mapsto z(x) = \frac{x - \bar{x}}{sd_x} = \frac{1}{sd_x} x - \frac{\bar{x}}{sd_x}$$

- Example (Toilet Paper)

Gewicht	Z(Gewicht)
35	-1.23
35	-1.23
35	-1.23
36	-1.12
...	...
42	-0.46
57	1.17
57	1.17
58	1.28
58	1.28

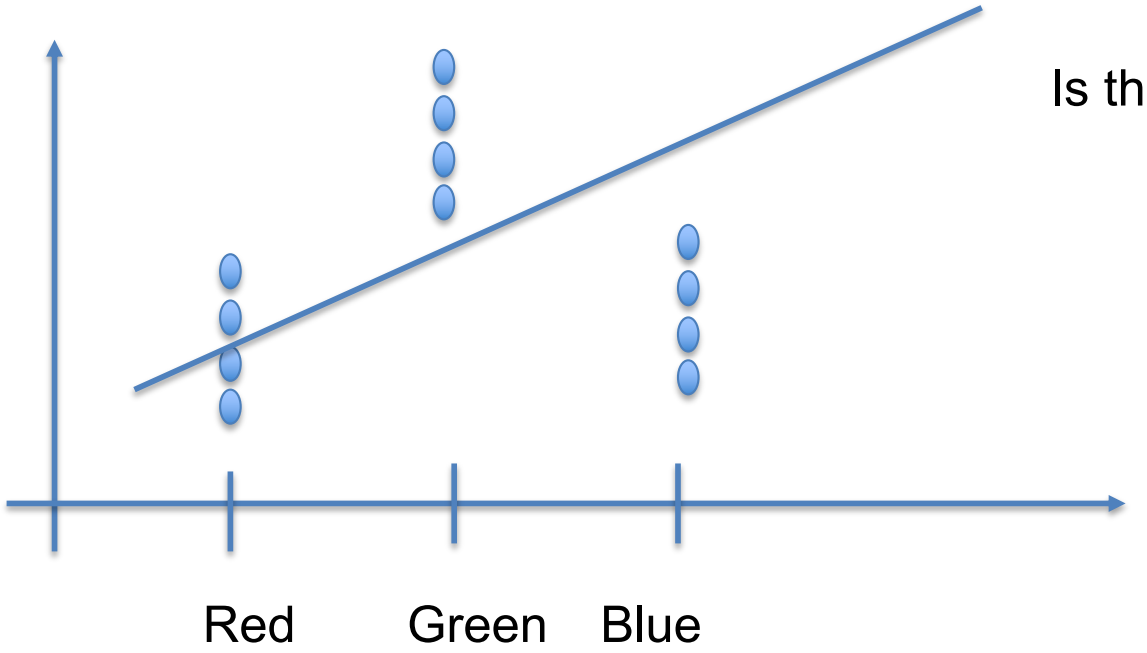


Scaling in scikit-learn

```
In [91]: import numpy as np
         import sklearn
         from sklearn.datasets import load_iris
         from sklearn.linear_model import LogisticRegression
         X, y = load_iris(return_X_y=True)
         print(X[0:4,])

         [[5.1 3.5 1.4 0.2]
          [4.9 3.  1.4 0.2]
          [4.7 3.2 1.3 0.2]
          [4.6 3.1 1.5 0.2]]

In [92]: # Scaling
         # Learning the scaler
         scaler = sklearn.preprocessing.StandardScaler().fit(X)
         X = scaler.transform(X)
         print(X[0:4,])
         np.mean(X, axis=0) #Close to zero
         np.std(X, axis=0)  #Close to one

         [[-0.90068117  1.01900435 -1.34022653 -1.3154443 ]
          [-1.14301691 -0.13197948 -1.34022653 -1.3154443 ]
          [-1.38535265  0.32841405 -1.39706395 -1.3154443 ]
          [-1.50652052  0.09821729 -1.2833891  -1.3154443 ]]

Out[92]: array([1., 1., 1., 1.])
```

For training and testset, just use the training set to fit the scaler.

Handling Categorical Data



Is this a good idea?

Remedy: coding

Each factor makes a new dimension:
is.red, is.green, is.blue