Linear Regression and Support Vector Machines

Marcus Rüb

Hahn-Schickard Villingen-Schwenningen
Marcus.rueb@hahn-schickard.de

This session:

- Linear regression
- Least square
- Gradient decent
- SVM

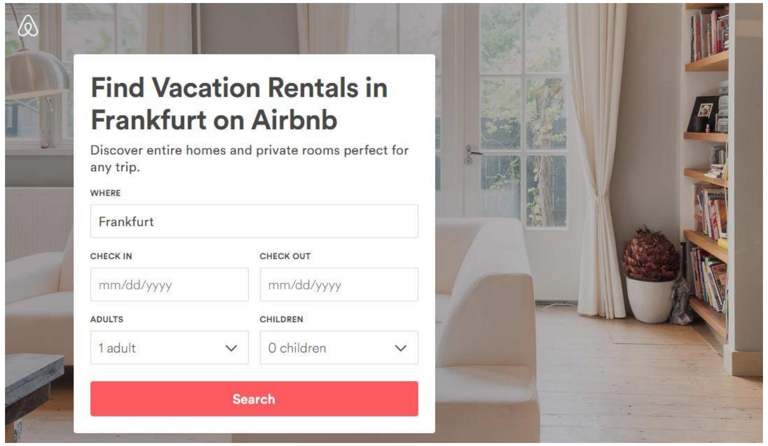
Outline for Supervised Learning (1)

Supervised Learning (1)

- Linear, polynomial regression
- Lasso, Ridge, ElasticNet regression
- Logistic Regression
- Support Vector Machines (SVM)

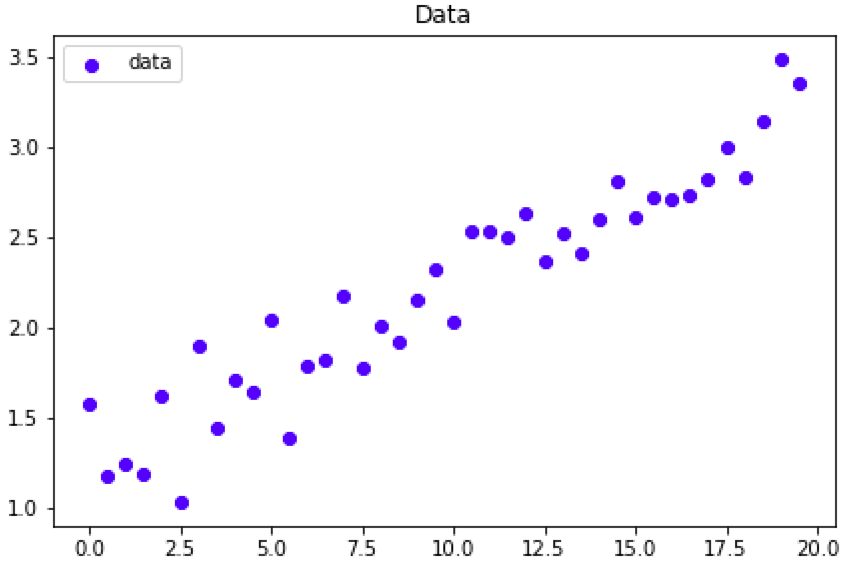
- Hands-on Supervised Learning

Predict price of vacation rentals in Frankfurt on Airbnb



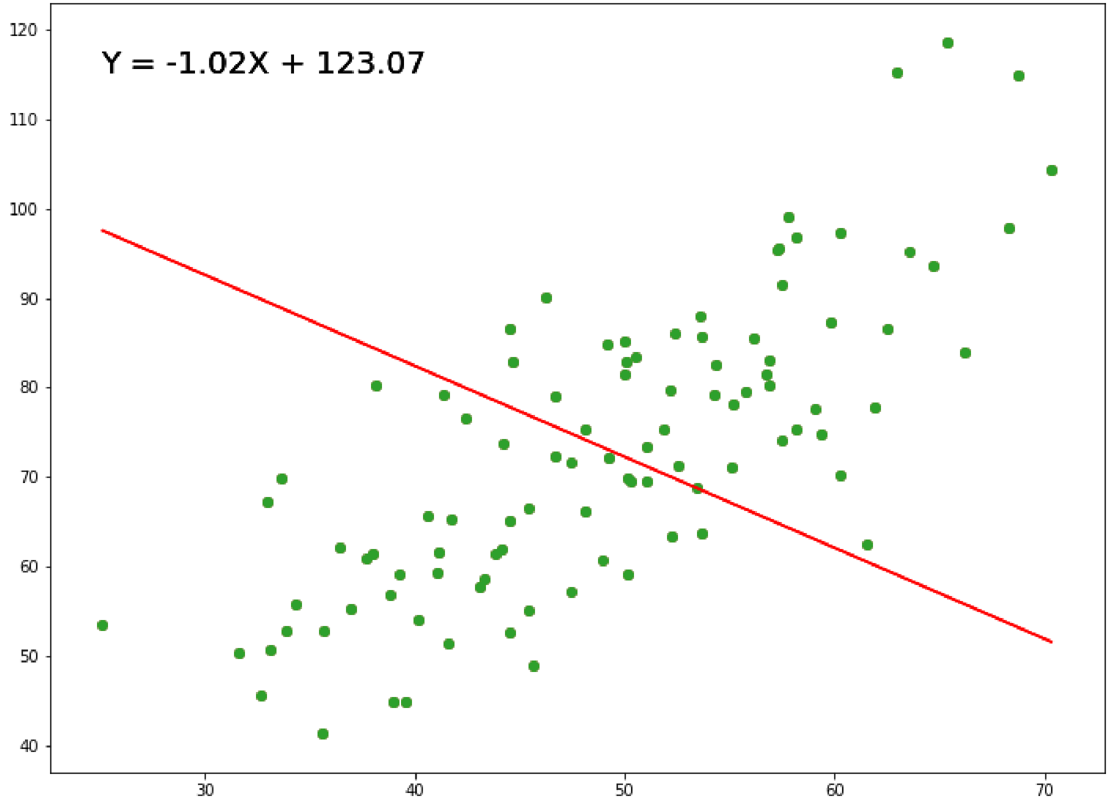
Linear Regression

Let’s create some data



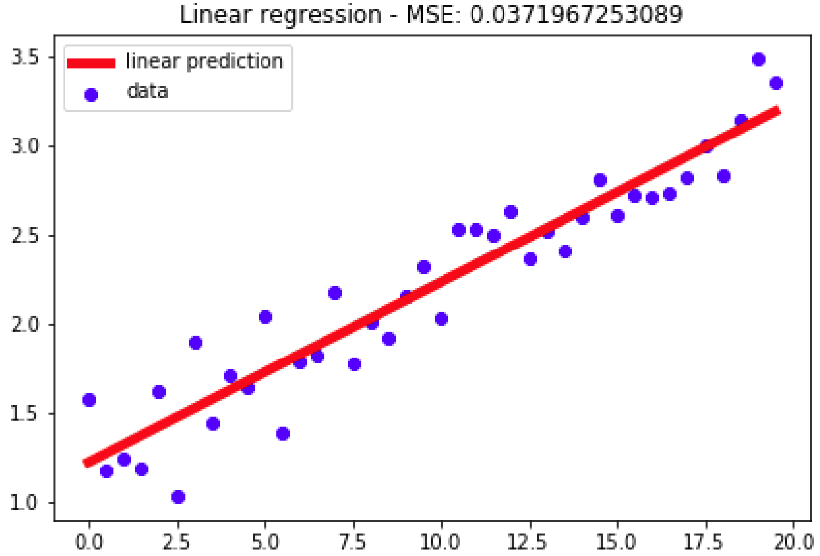
$$y = 0.1x + 1.25 + 0.2 GaussianNoise$$

Fit in a line



Linear Regression

Let’s perform linear regression…



$y = 0.1x + 1.25 + 0.2\ GaussianNoise$

$y = \mathrm{m}x + b$

w = 0.1014
b = 1.2258

$y = 0.1014x + 1.2258$

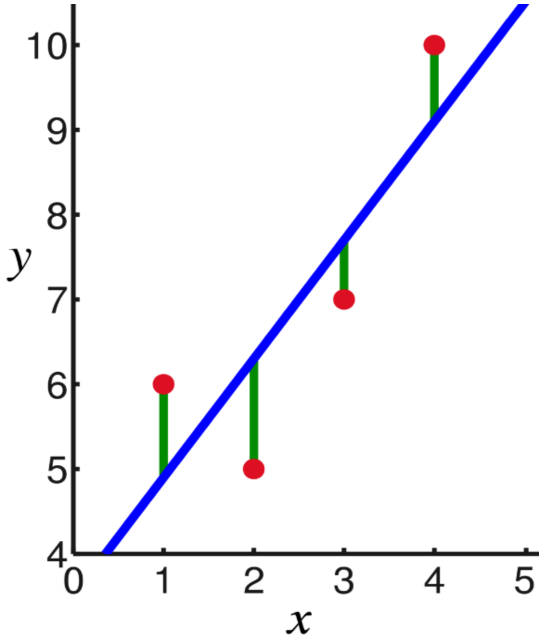
How we fitted the line?

We just found the values of ‘m’ and ‘b’ that minimize the Mean Squared Error

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

where N is the number of data points, f_i the value returned by the model and y_i the actual value for data point i.

Mean Squared Error



C.F. Gauss

Carl Friedrich Gauss

Born: 30. April 1777
† 23. Februar 1855
German mathematician

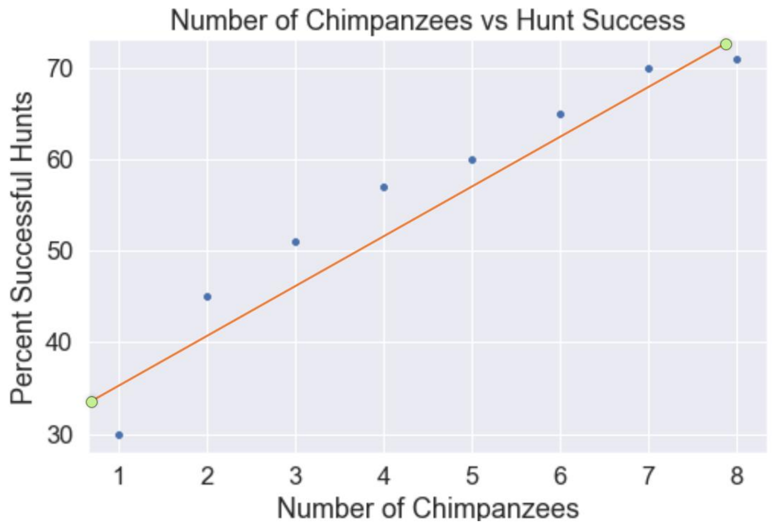
With 18 he invent the Least square method

Least Square

$$y = mx + b$$

$$m = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2} \qquad b = \frac{\Sigma y - m(\Sigma x)}{n}$$

	Number of Chimpanzees	Percent Successful Hunts
0	1	30
1	2	45
2	3	51
3	4	57
4	5	60
5	6	65
6	7	70
7	8	71
n	x	y



Number of Chimpanzees (x)	Percent Successful Hunts (y)	xy	x^2	y^2
1	30	30	1	900
2	45	90	4	2025

$$m = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2} \qquad b = \frac{\Sigma y - m(\Sigma x)}{n}$$

Number of Chimpanzees (x)	Percent Successful Hunts (y)	xy	x^2	y^2
1	30	30	1	900
2	45	90	4	2025
3	51	153	9	2601
4	57	228	16	3249
5	60	300	25	3600
6	65	390	36	4225
7	70	490	49	4900
8	71	568	64	5041
$\sum x$	$\sum y$	$\sum xy$	$\sum x^2$	$\sum y^2$
36	449	2249	204	26541

Number of Chimpanzees (x)	Percent Successful Hunts (y)	xy	x^2	y^2
1	30	30	1	900
2	45	90	4	2025
3	51	153	9	2601
4	57	228	16	3249
5	60	300	25	3600
6	65	390	36	4225
7	70	490	49	4900
8	71	568	64	5041
$\sum x$	$\sum y$	$\sum xy$	$\sum x^2$	$\sum y^2$
36	449	2249	204	26541

$$m = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2} \qquad b = \frac{\Sigma y - m(\Sigma x)}{n}$$

$$m = \frac{8(2249) - (36)(449)}{8(204) - (36)^2} \qquad b = \frac{449 - 5.4405(36)}{8}$$

$$m = 5.4405 \qquad b = 31.6429$$

$$y = mx + b$$

$$y = 5.4405x + 31.6429$$

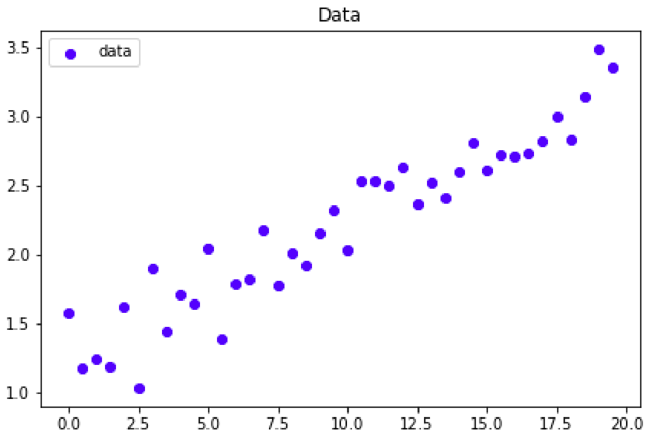
Least Squares - notation

$y = wx + b = b + wx$

$X = 1, x$

$W = b, w$

$y = WX = b + wx$



Also called features

$\boxed{X} = \boxed{x_0, x_1, x_2, x_3, \dots, x_n}$ $\quad x_0 = 1$

$W = w_0, w_1, w_2, w_3, \dots, w_n$ $\quad w_0 = b$

$y = WX = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$

x_0	x_1	y
1	0.0	1.57
1	0.5	1.18
1	1.0	1.24
1	1.5	1.19
1	2.0	1.62

X = matrix (m,n)
Y = vector (m)

m = number of samples
n = number of features

How we fitted the line?

Which values of line parameters minimize the Mean Squared Error?

Least squares Method

Carl Friedrich Gauss

Least Squares Method

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

$X = x_0, x_1, x_2, x_3, \dots, x_n$ $\quad x_0 = 1$

$W = w_0, w_1, w_2, w_3, \dots, w_n$ $\quad w_0 = b$

$y = WX = w_0 x_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$

$\hat{w}$ is the best approximation to W

How we fitted the line?

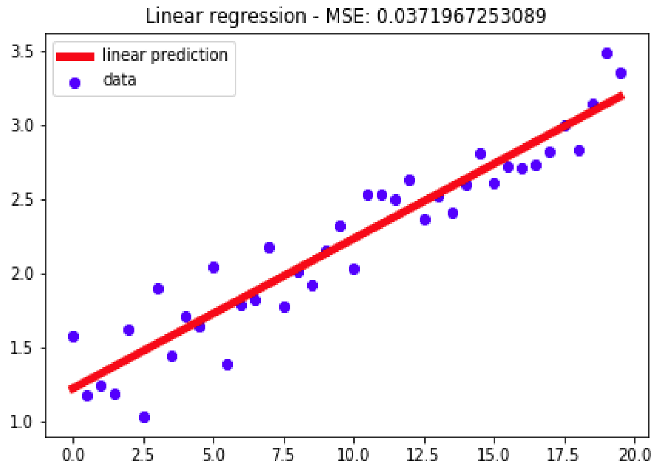
Which values of line parameters minimize the Mean Squared Error?

Least squares Method

Carl Friedrich Gauss

Least Squares Method

$$\hat{w} = (X^T X)^{-1} X^T y$$



Gradient Descent

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2$$

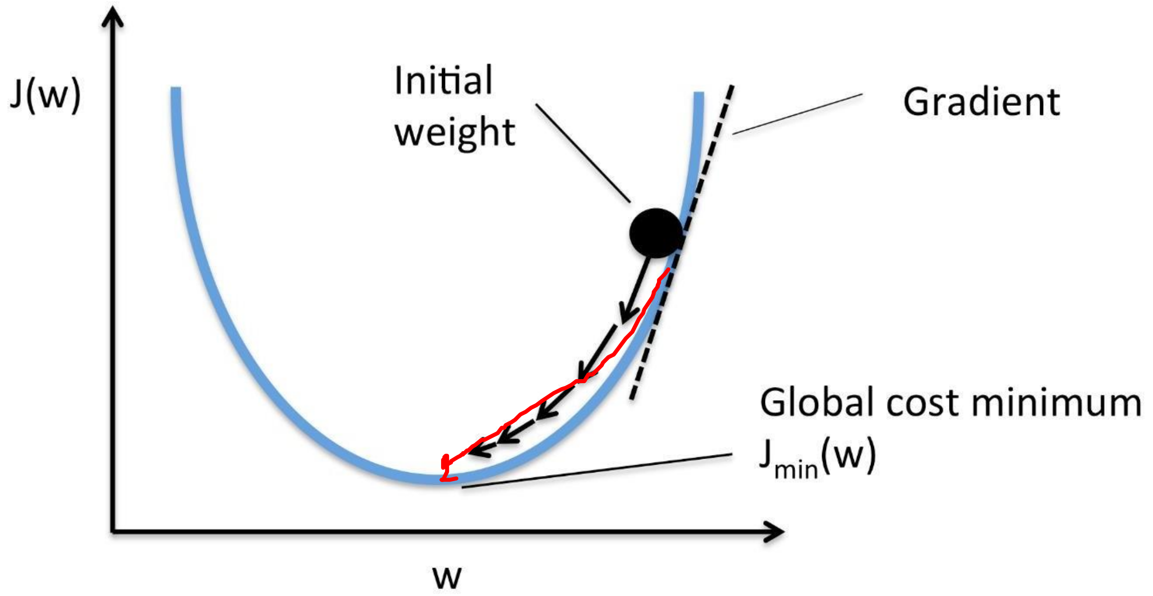
where N is the number of data points, f_i the value returned by the model and y_i the actual value for data point i.

Mean Squared Error

Learning

$$W = W - \alpha \frac{\partial J}{\partial W}$$

There is another way: Gradient Descent



Gradient Descent Visualization. Credit: rasbt.github.io

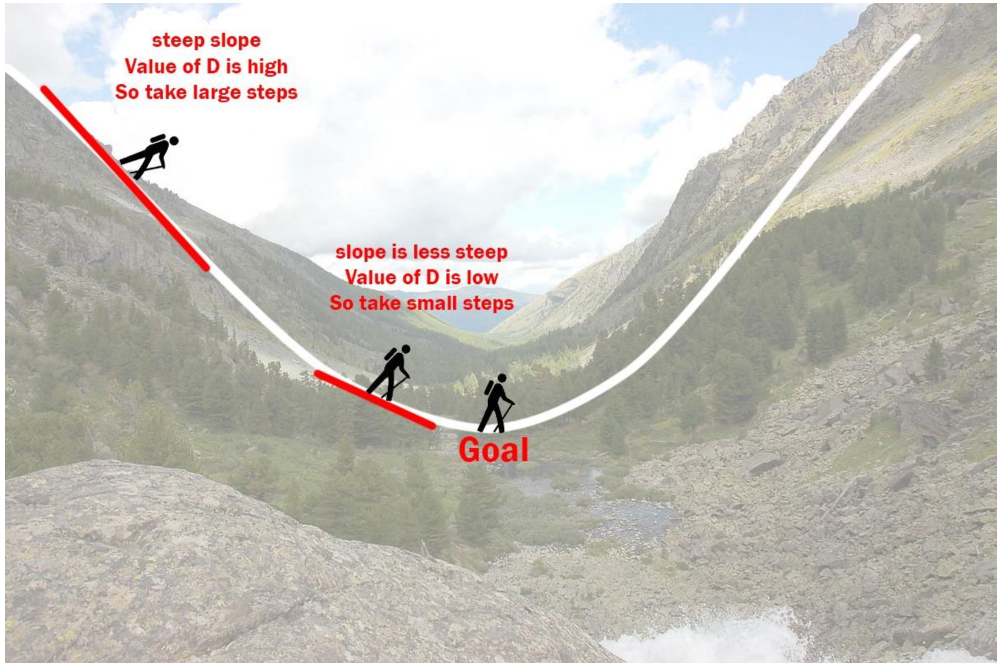
Marcus.Rueb@Hahn-Schickard.de - Tel.: 07721/943-180

Gradient Descent

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)^2$$

$$y = mx + b$$

$$E = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$



Derivation:

$$D_{\underline{m}} = \frac{1}{n}\sum_{i=0}^{n}2(y_i - (mx_i + c))(-x_i)$$

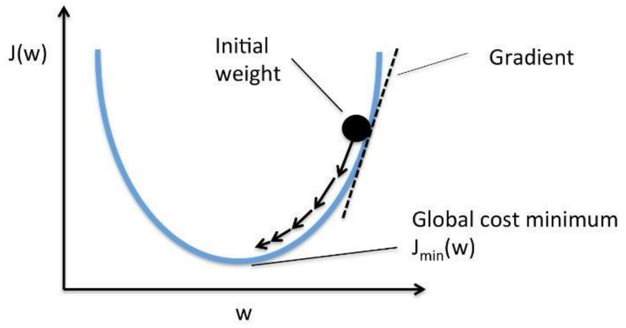
$$D_m = \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - \bar{y}_i)$$

$$D_{\underline{c}} = \frac{-2}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)$$

$$m = m - \underline{L} \times D_m$$

$$c = c - L \times D_c$$

$$\hat{w} = (X^T X)^{-1} X^T y$$



Least Squares

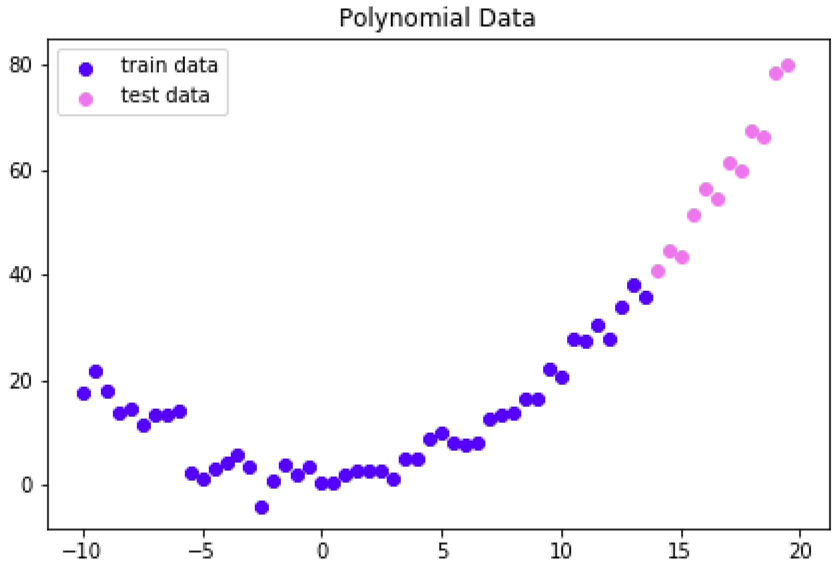
- when there is a relatively small number of features (< 1,000)

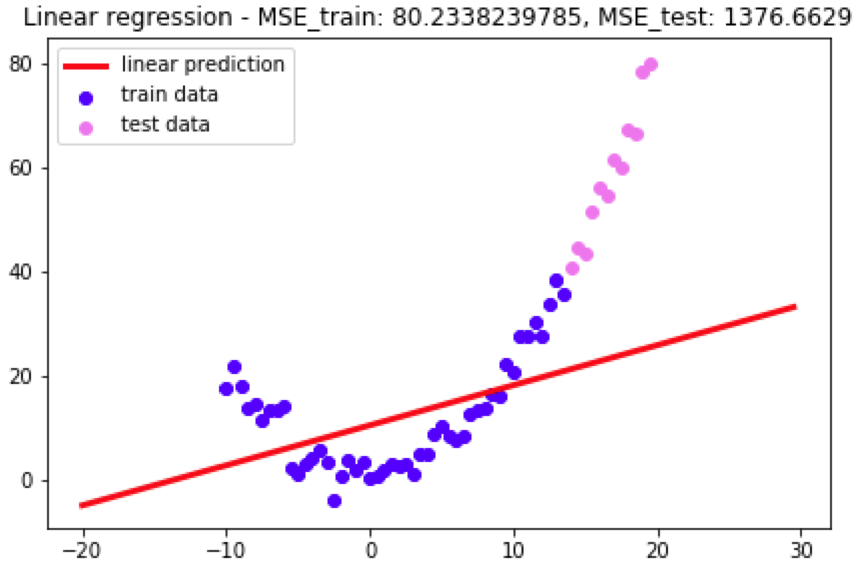
Gradient Descent

- when there are many features (> 1,000)
- when we need to stop training at any time
    - e.g. if we only have 1 minute
- If data does not fit in memory
- If you have new data (e.g. stream) and don't want to start all over (with all previous data)

Marcus.Rueb@Hahn-Schickard.de - Tel.: 07721/943-180

Polynomial Regression





$y = 0.2x^2 + 0.1x + 1 + 3GaussianNoise$

Linear Regression: $y = w_0 + w_1x$

Polynomial Regression

- You already know how to do it
- It is not a new technique, it's a **feature**

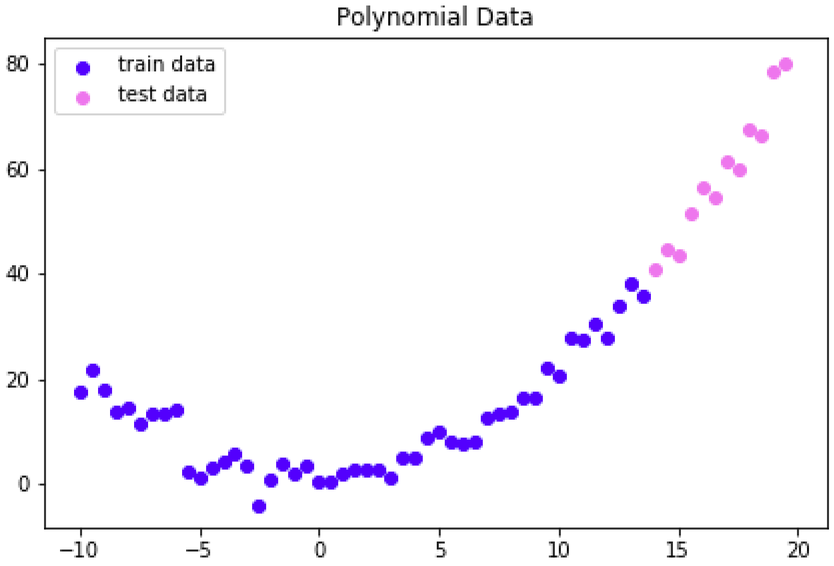
x

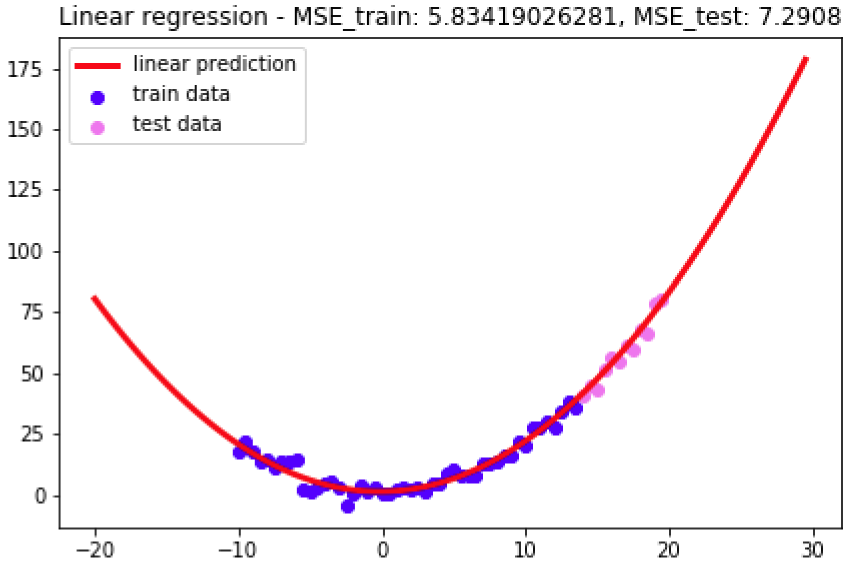
x_0	x_1	y
1	-10.0	17.74
1	-9.5	21.86
1	-9	17.84
1	-8.5	13.71
1	-8	14.47

x x^2

x_0	x_1	x_2	y
1	-10.0	100.0	17.74
1	-9.5	90.25	21.86
1	-9	81.00	17.84
1	-8.5	72.25	13.71
1	-8	64.00	14.47

Polynomial Regression





$y = 0.2x^2 + 0.1x + 1 + 3GaussianNoise$

Linear Regression: $y = w_0 + w_1x + w_2x^2$

Polynomial Regression

- Polynomial Regression = Linear Regression with polynomial features
- You can get creative:
  - x^2, x^3, x^4…
  - zx^2, zx^3, z^2x^2, …

Features		x	x^2	
	x_0	x_1	x_2	y
	1	-10.0	100.0	17.74
	1	-9.5	90.25	21.86
	1	-9	81.00	17.84
	1	-8.5	72.25	13.71
	1	-8	64.00	14.47

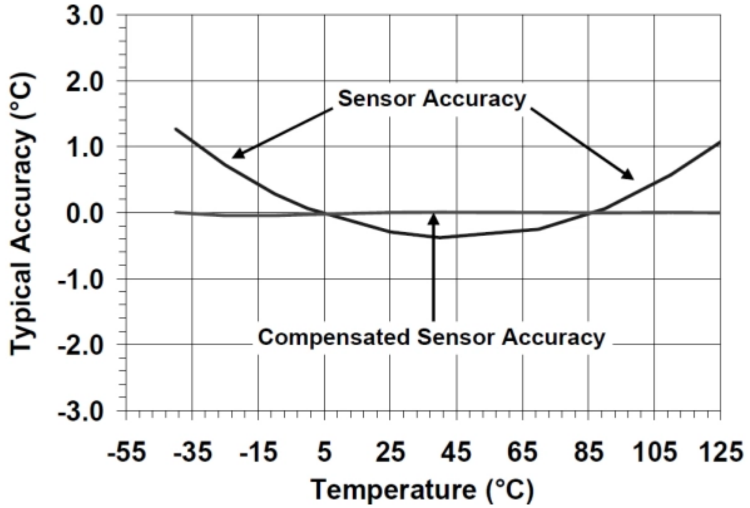
What if some of the inputs are irrelevant?

- Ridge Regression
- Lasso Regression
- ElasticNet Regression

Where is this used?

C02 Sensor

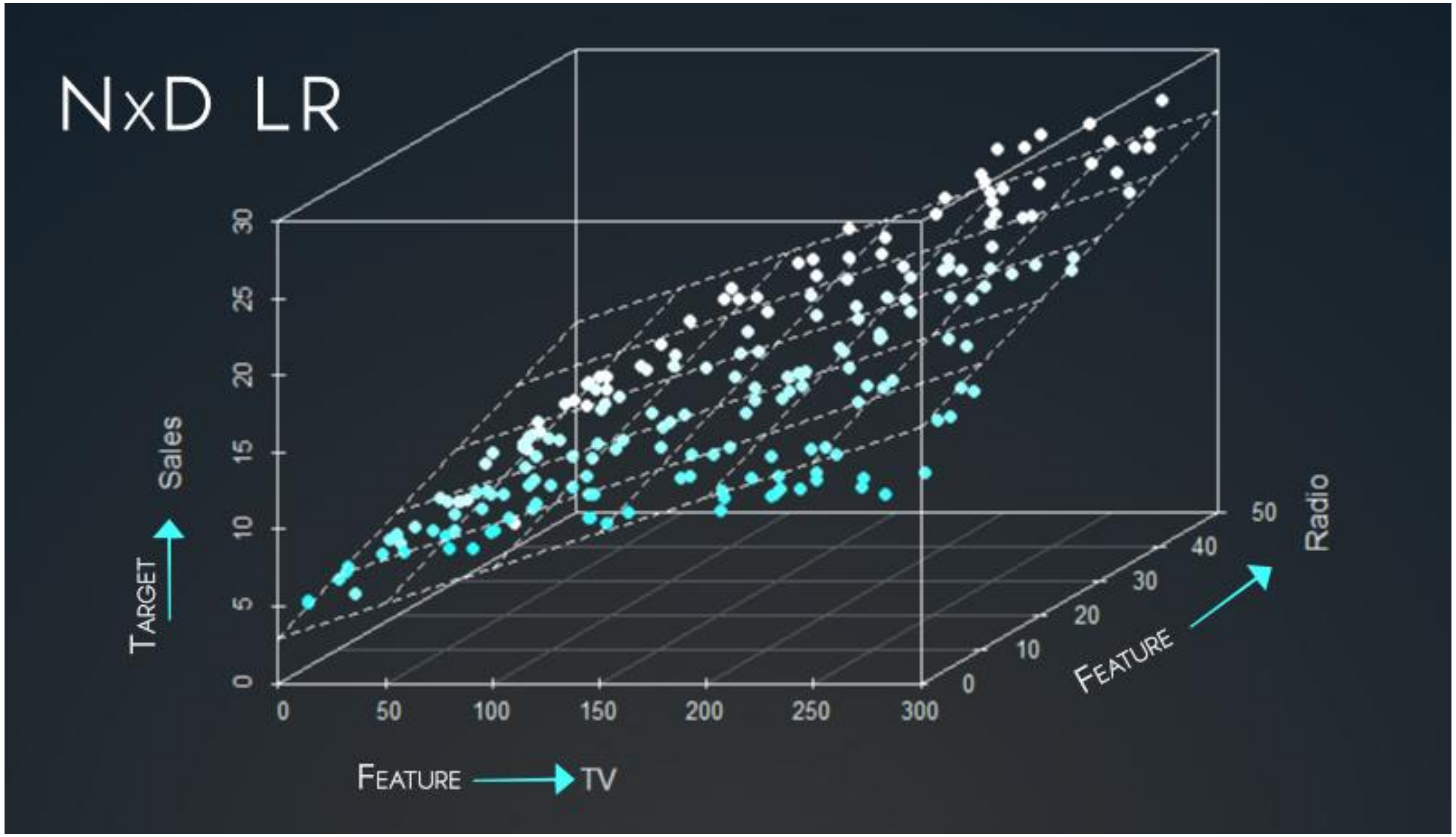
Error with different temperatures, humidity and pressure



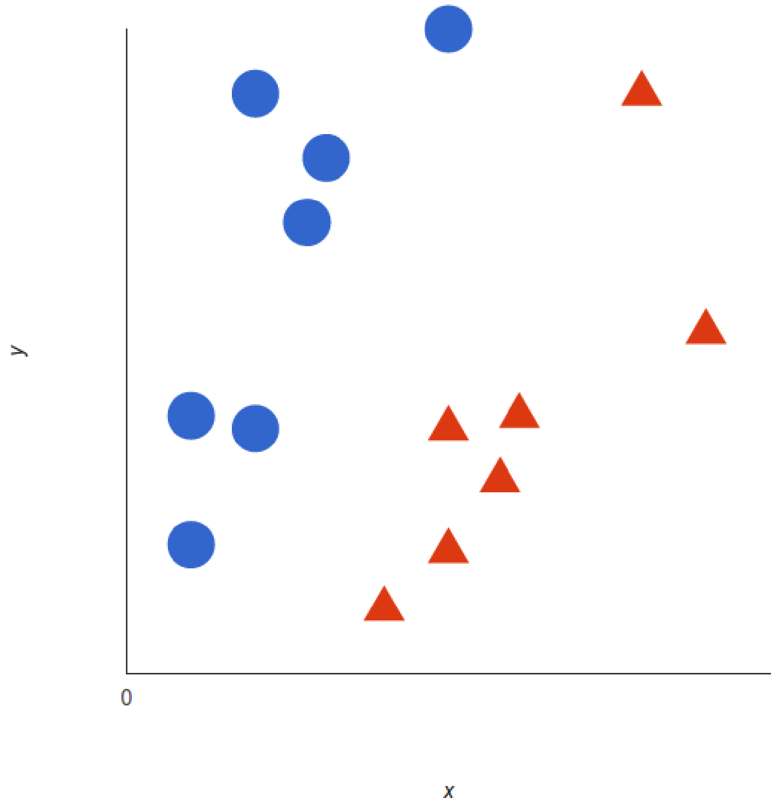
C02 = Sensor*x0+temp*x1+hum*x2+pres*x3 + b

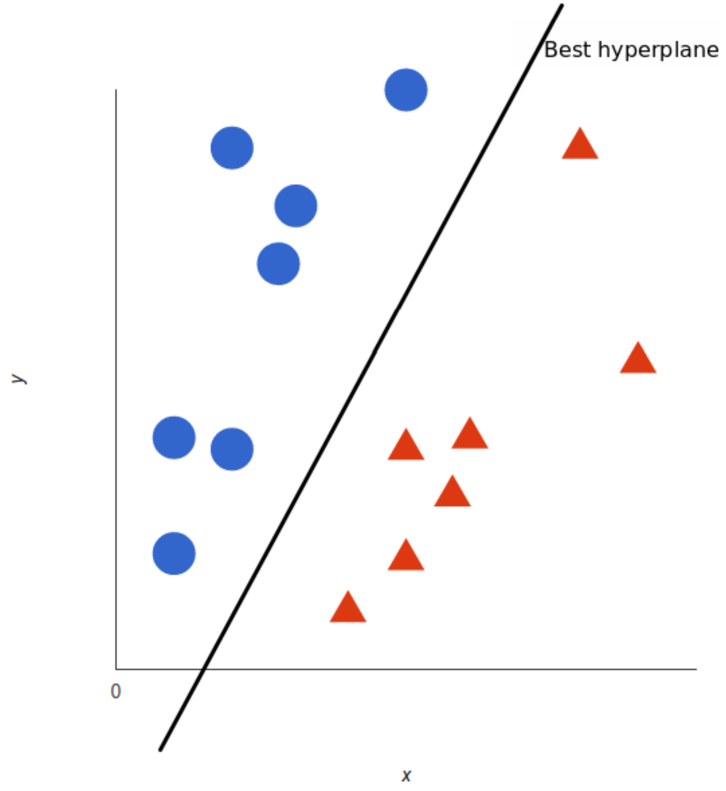
Car price Calculator

What for features do we have?

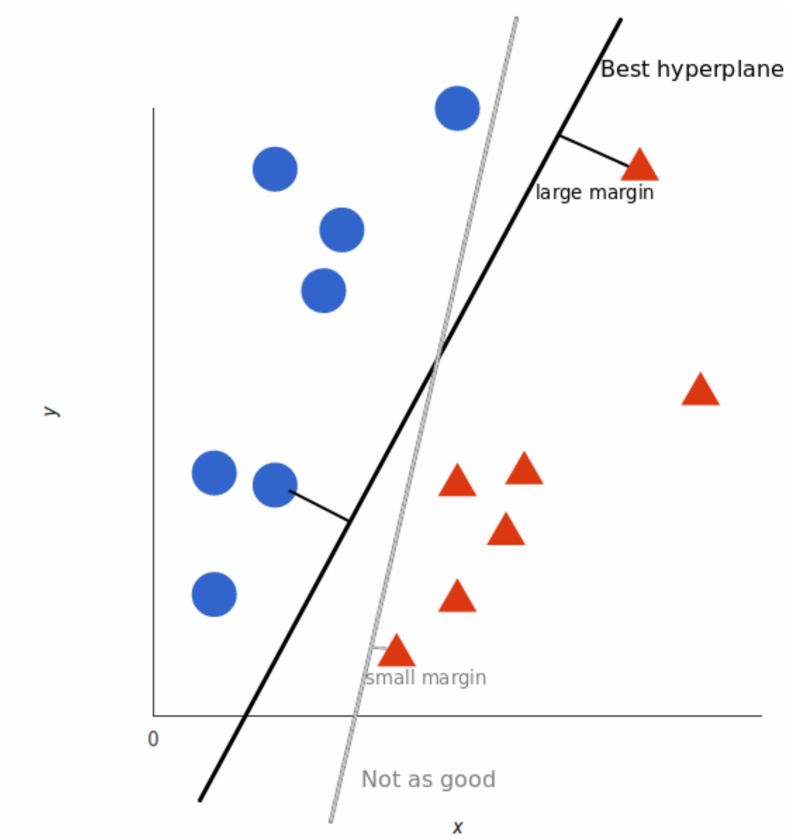


Support Vector Machines

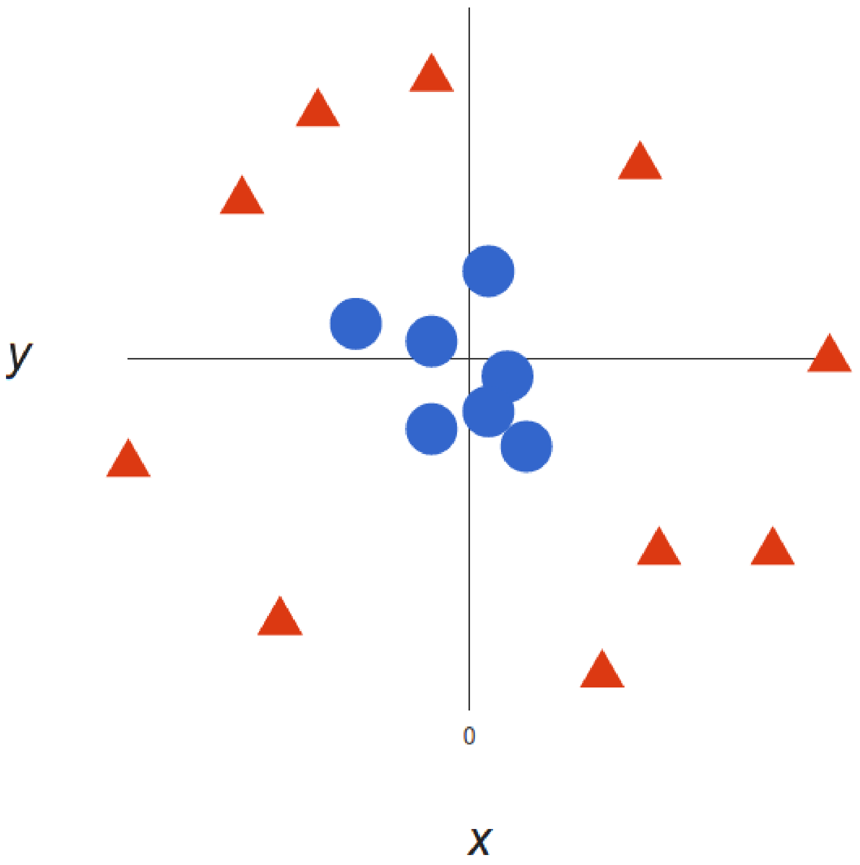




But, what exactly is the best hyperplane?



Nonlinear data



Questions

What is linear regression?

What is it used for?

How do we get the linear model? Name 2 Methods. And when do you use them.

Explain the steps of the gradient decent.

What is SVM used for?

Which features do we have for example in a House Price calculator with linear regression?

What is the kernel trick?

Calculate a linear regression model for this Datapoints

N	X	Y
0	1	2
1	2	3
2	3	7
3	4	8

Marcus.Rueb@Hahn-Schickard.de - Tel.: 07721/943-180