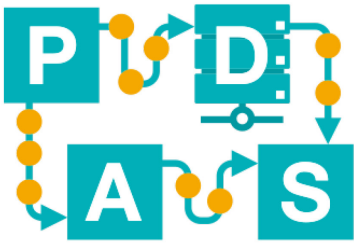
Regression

Lecture 5

IDS-L5



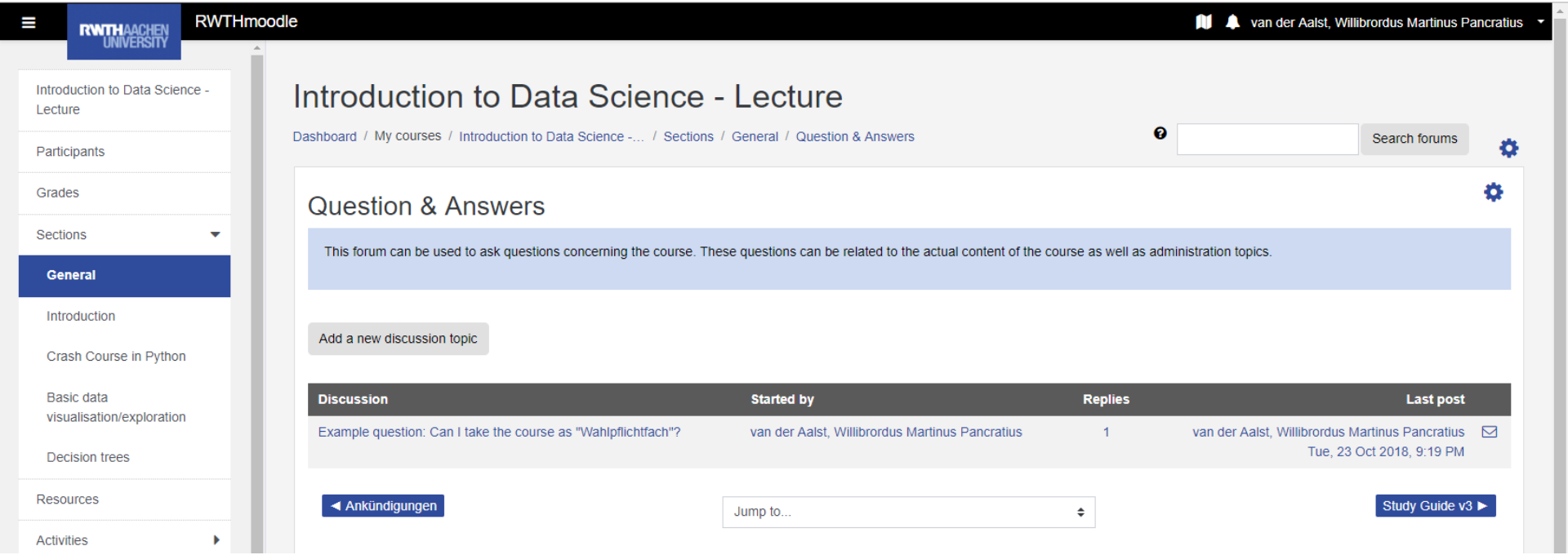
Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

Questions about the course

- For every question related to the course, please use the Q&A forum on Moodle (rather than emails)
  - For both technical and organizational matters
  - Sections > General > Question & Answers
  - Before posting, check if someone else has already asked the same question.

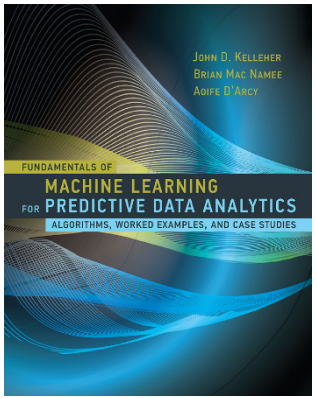
Questions about the course



Chair of Process and Data Science

Outline of Today's Lecture

- Regression: Basic idea
- Simple linear regression
- Multiple descriptive features
- Interpretation of results
- Handling categorical features
- Logistic regression
- Extensions: non-linear and multinomial



Based on Chapter 7 of Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Regression: Basic idea

Error-based learning

- A parameterized prediction model is initialized with a set of random parameters and an error function is used to judge how well this initial model performs when making predictions for instances in a training dataset.
- Based on the value of the error function the parameters are iteratively adjusted to create a more and more accurate model.

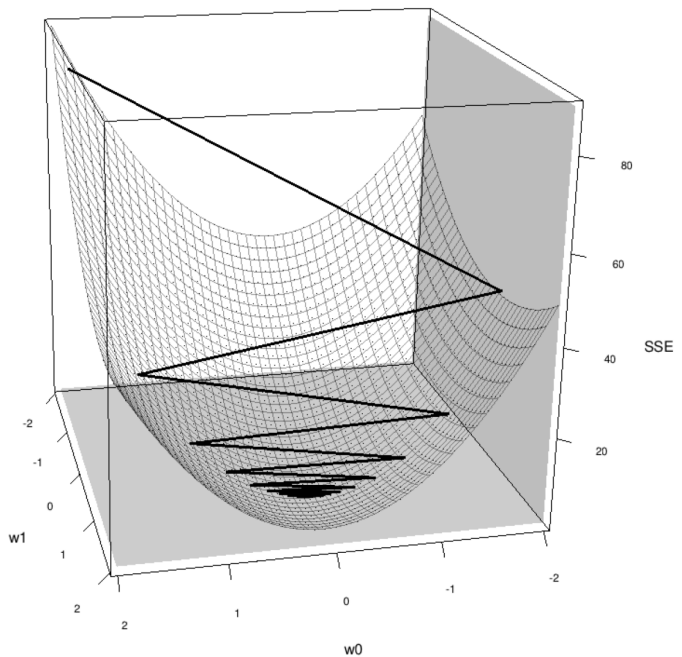
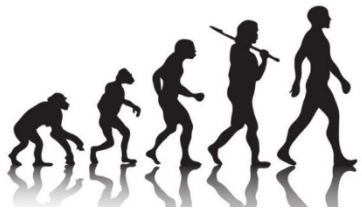
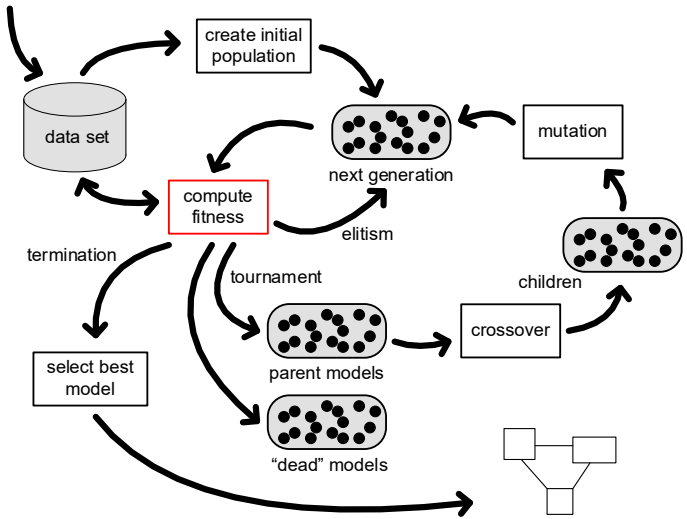


Figure taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Error-based learning

- **Addressed in this course:**
  - **Regression (this course)**
  - **SVMs (next lecture)**
  - **Neural networks**
- **Genetic algorithms**
- **Various other evolutionary approaches.**

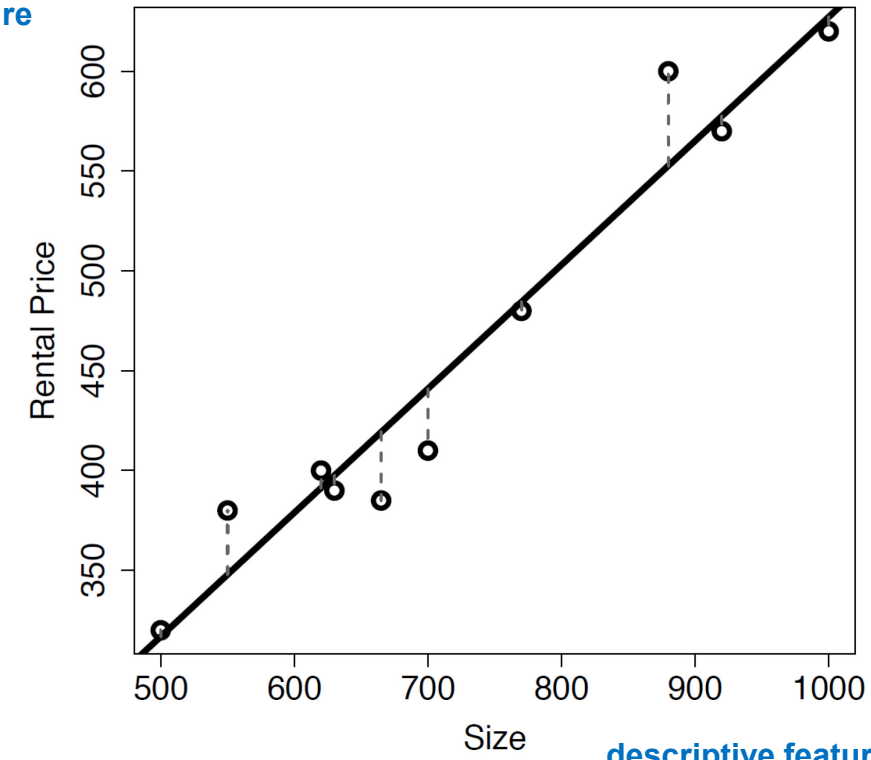




Chair of Process and Data Science

Regression: Basic idea

target feature



descriptive features (only one here)

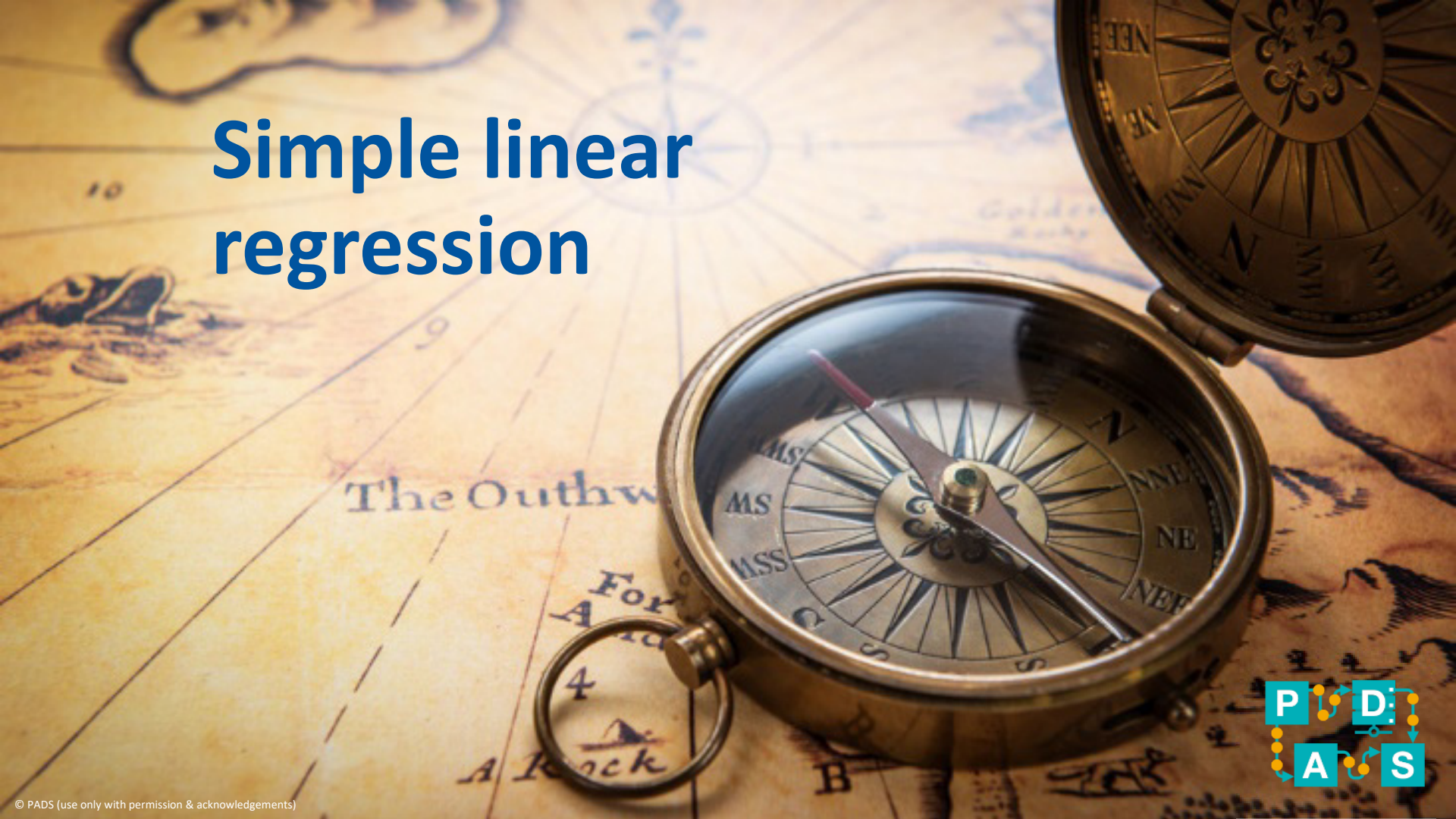
Find the line such that the errors between model and observed data are minimized.

Chair of Process and Data Science

Decision trees versus regression

- **Decision trees** were initially developed for categorical features and then extended to continuous features.
- **Regression** followed the reverse path (most suitable for continuous data).
- Both are **supervised** learning techniques.

Simple linear regression



Example data set

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	[illegible]	12	50	A	[illegible]00
9	920	14	8	C	570
10	1,000	9	24	B	620

Rental prices of offices in Dublin.

Let's take Rental Price as the target feature and pick one descriptive feature: Size.

Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Simplified data set

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8		
9		
10	1,000	620

- **How does the rental price depend on the size of the office?**
  - x = Size
  - y = Rental Price

$$y = ax + b$$

$$\text{Rental Price} = 6.47 + 0.62 \times \text{Size}$$?

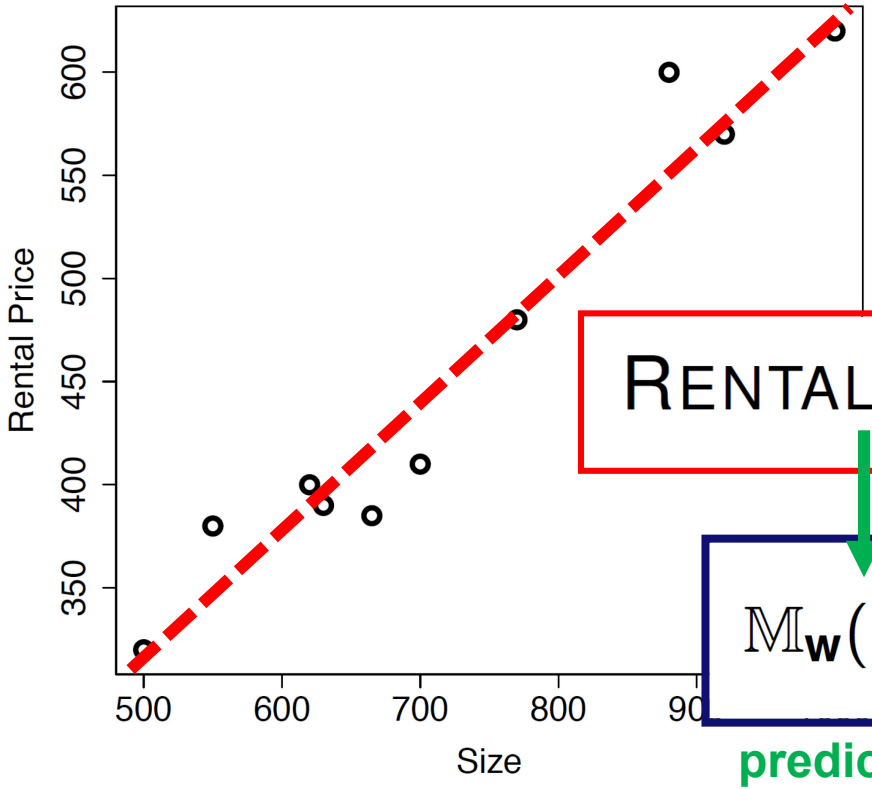
General problem

Find:

$$y = f(x_1, x_2, x_3, \ldots, x_n)$$

To keep it simple we start with just one descriptive feature and a linear function.

Simple regression



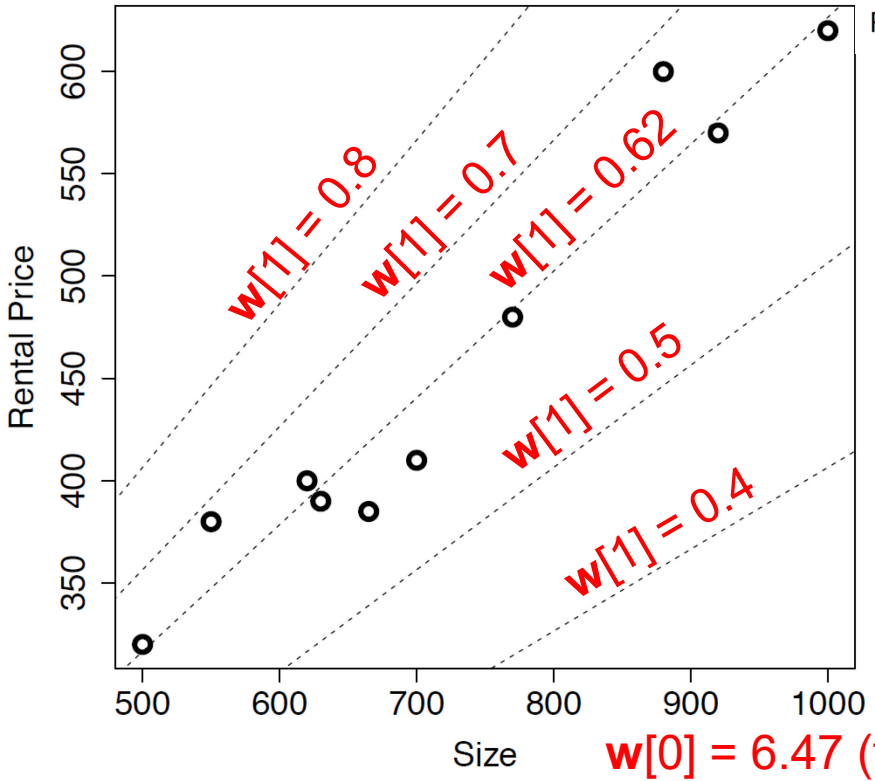
$$\text{RENTAL PRICE} = 6.47 + 0.62 \times \text{SIZE}$$

$$\mathbb{M}_{\mathbf{w}}(d) = \mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}[1]$$

prediction weights feature

Chair of Process and Data Science

Different values of w[1]



RENTAL PRICE $= 6.47 + 0.62 \times$ SIZE

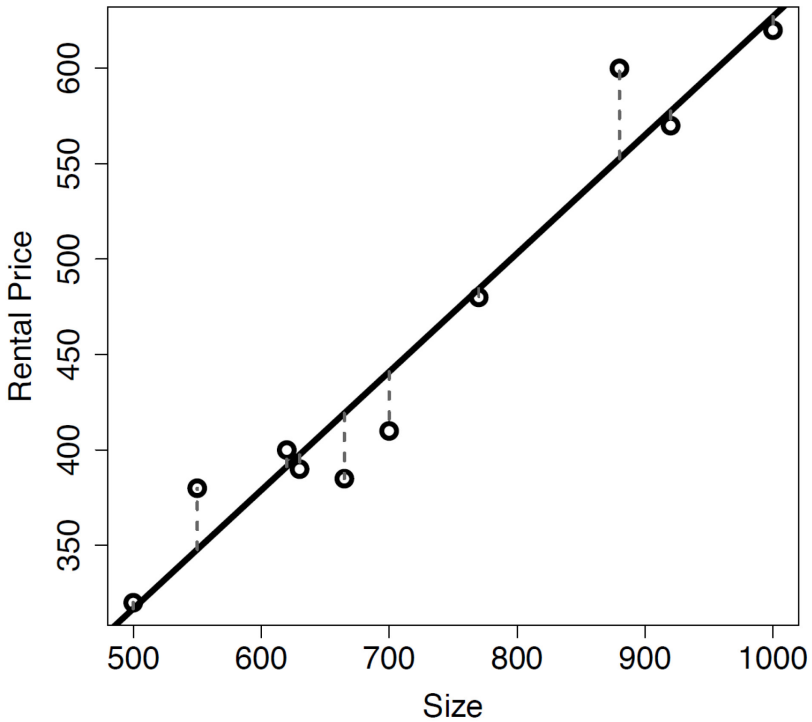
$$\mathbb{M}_{\mathbf{w}}(d) = \mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}[1]$$

Goal: pick the one with the "smallest error"

w[0] = 6.47 (fixed)

Chair of Process and Data Science

Measuring the error: Sum of squared errors

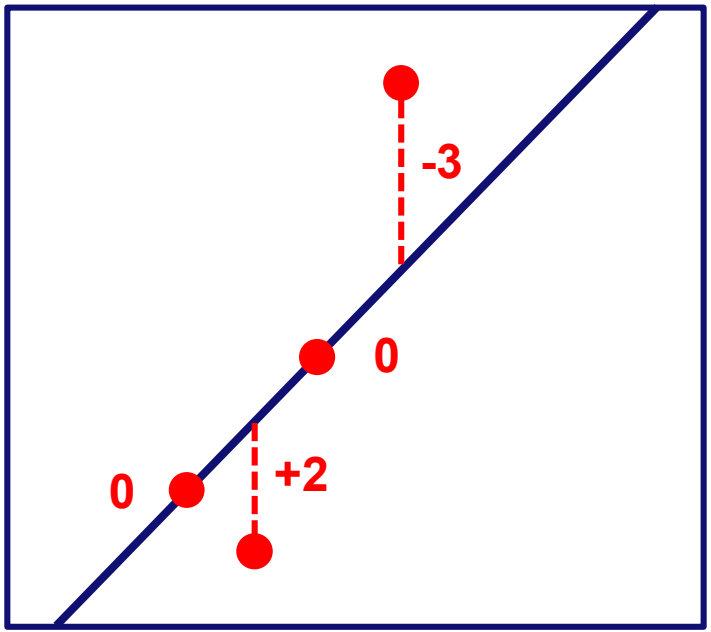


$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i[1]))^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}_i[1]))^2$$

ID	Rental Price	Model Prediction	Error Error	Squared Error
1	320	316.79	3.21	10.32
2	380	347.82	32.18	1,035.62
3	400	391.26	8.74	76.32
4	390	397.47	-7.47	55.80
5	385	419.19	-34.19	1,169.13
6	410	440.91	-30.91	955.73
7	480	484.36	-4.36	19.01
8	600	552.63	47.37	2,243.90
9	570	577.46	-7.46	55.59
10	620	627.11	-7.11	50.51
			Sum	**5,671.64**
		Sum of squared errors (Sum/2)		**2,835.82**

Measuring the error: Different notions



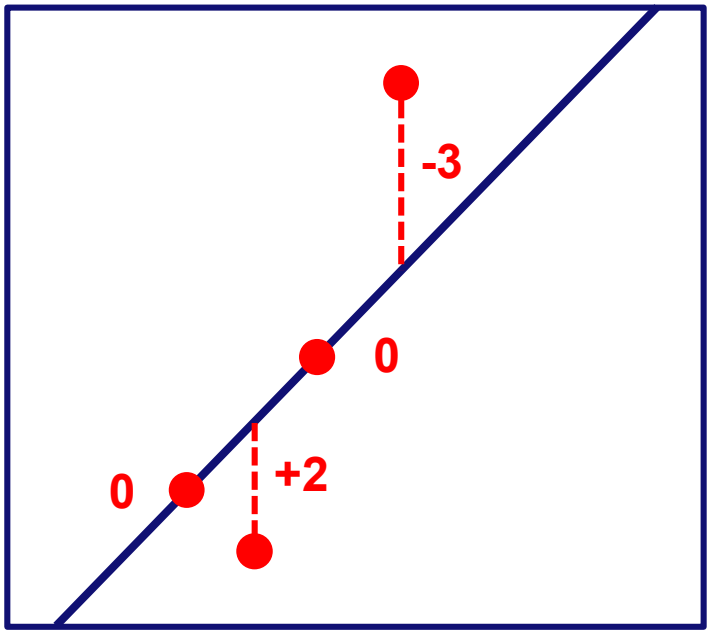
$$\text{sum of squared errors} = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2$$

$$\text{mean squared error} = \frac{\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}$$

$$\text{root mean squared error} = \sqrt{\frac{\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}}$$

$$\text{mean absolute error} = \frac{\sum_{i=1}^{n} abs(t_i - \mathbb{M}(\mathbf{d}_i))}{n}$$

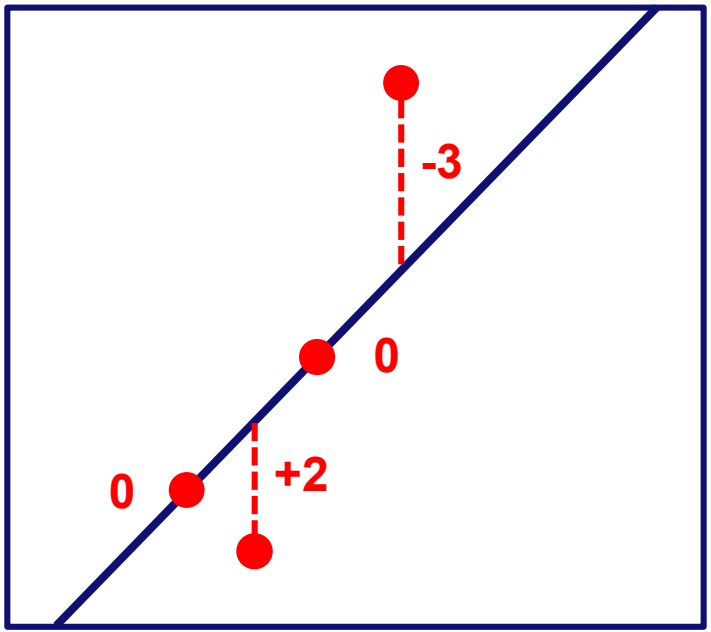
Sum of squared errors



$$\text{sum of squared errors} = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2$$

$$(0^2 + 2^2 + 0^2 + (-3)^2)/2 = 6.5$$

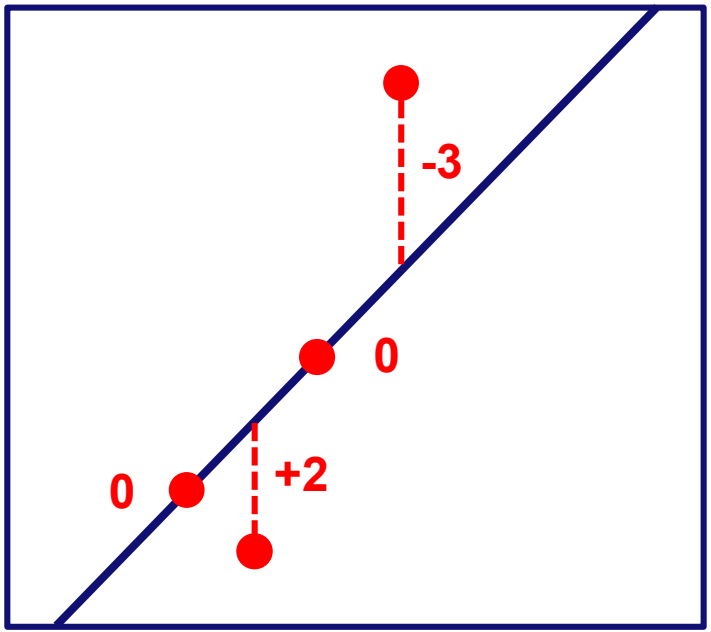
Mean squared error



$$\text{mean squared error} = \frac{\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}$$

$$(0^2 + 2^2 + 0^2 + (-3)^2)/4 = 3.25$$

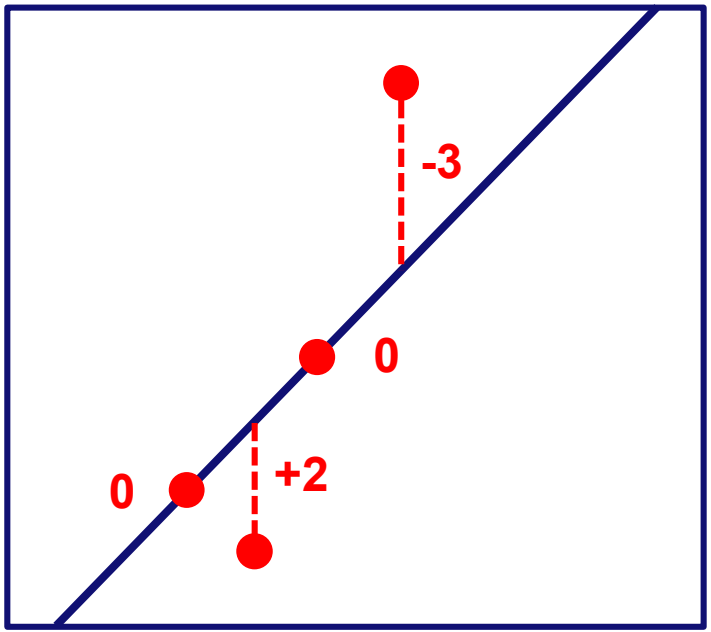
Root mean squared error



$$\text{root mean squared error} = \sqrt{\frac{\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}}$$

$$\sqrt{(0^2 + 2^2 + 0^2 + (-3)^2)/4} = 1.803$$

Measuring the error

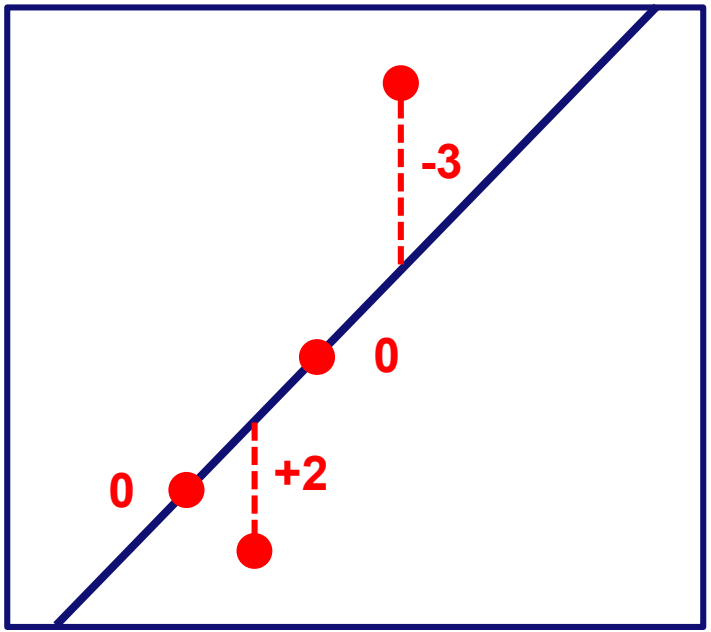


$$\text{mean absolute error} = \frac{\sum_{i=1}^{n} abs(t_i - \mathbb{M}(\mathbf{d}_i))}{n}$$

(|0| + |2| + |0| + |-3|)/4 = 1.25

Chair of Process and Data Science

Measuring the error



$$\text{sum of squared errors} = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2$$

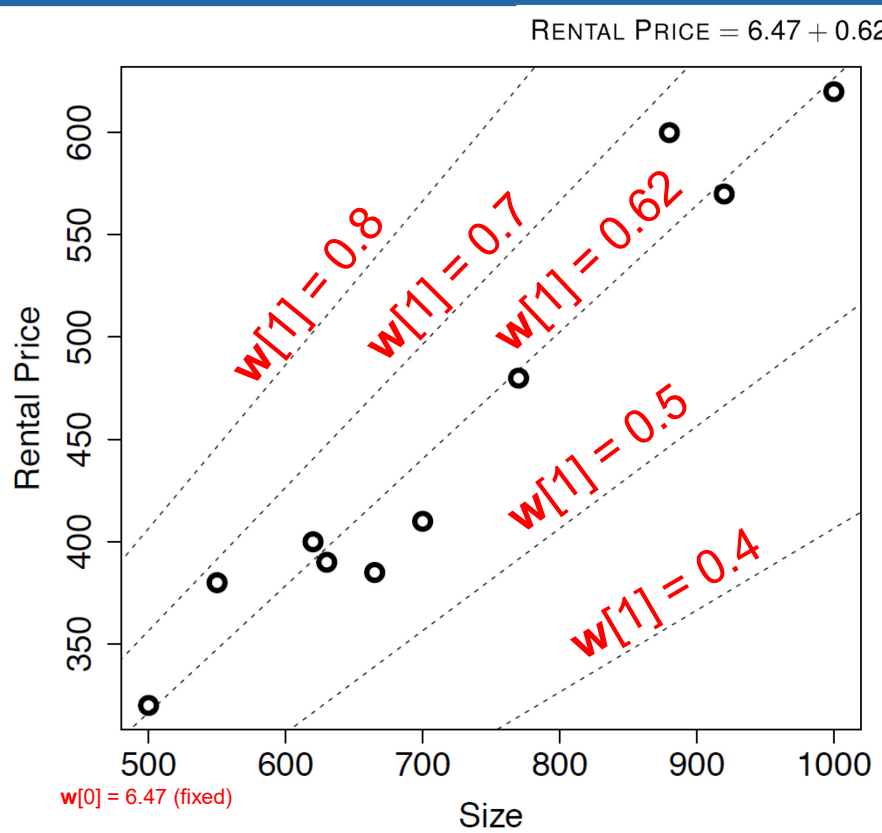
$$\text{mean squared error} = \frac{\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}$$

$$\text{root mean squared error} = \sqrt{\frac{\sum_{i=1}^{n}(t_i - \mathbb{M}(\mathbf{d}_i))^2}{n}}$$

$$\text{mean absolute error} = \frac{\sum_{i=1}^{n} abs(t_i - \mathbb{M}(\mathbf{d}_i))}{n}$$

They all express the same idea, but the first one is chosen because it has the simplest derivative.

Sum of squared errors for example

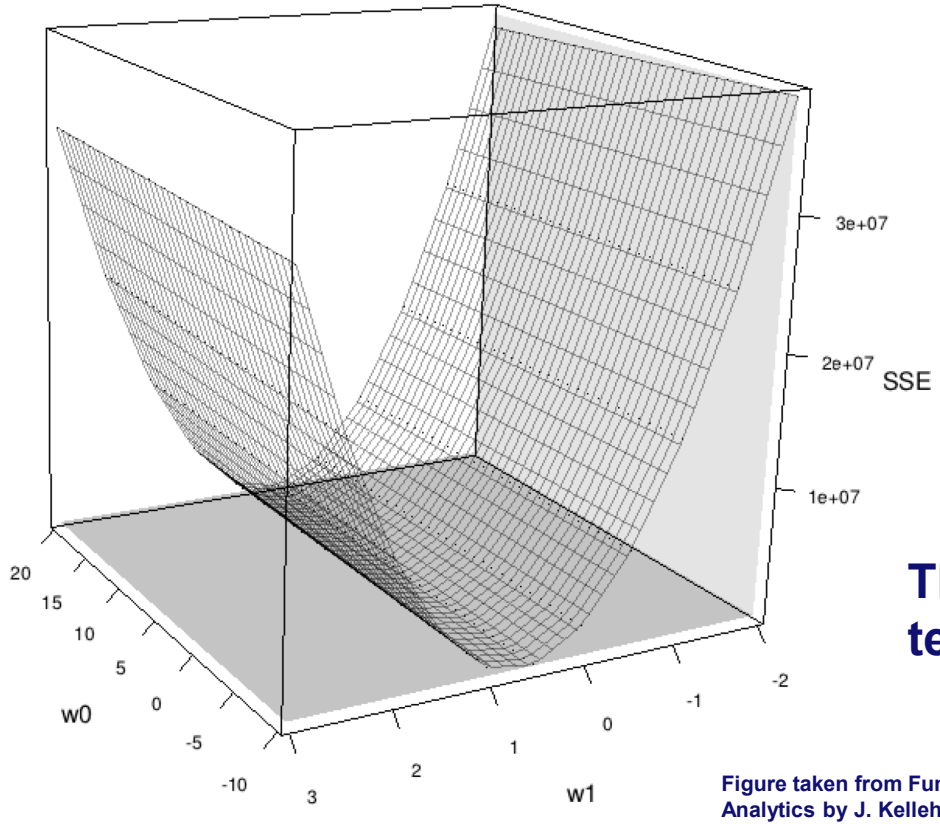


$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i[1]))^2$$

w[1]	error
w[1] = 0.80	90978
w[1] = 0.70	20092
w[1] = 0.62	**2837**
w[1] = 0.50	42712
w[1] = 0.40	136218

Matches intuition!

Let's plot the error in terms of w[0] and w[1]



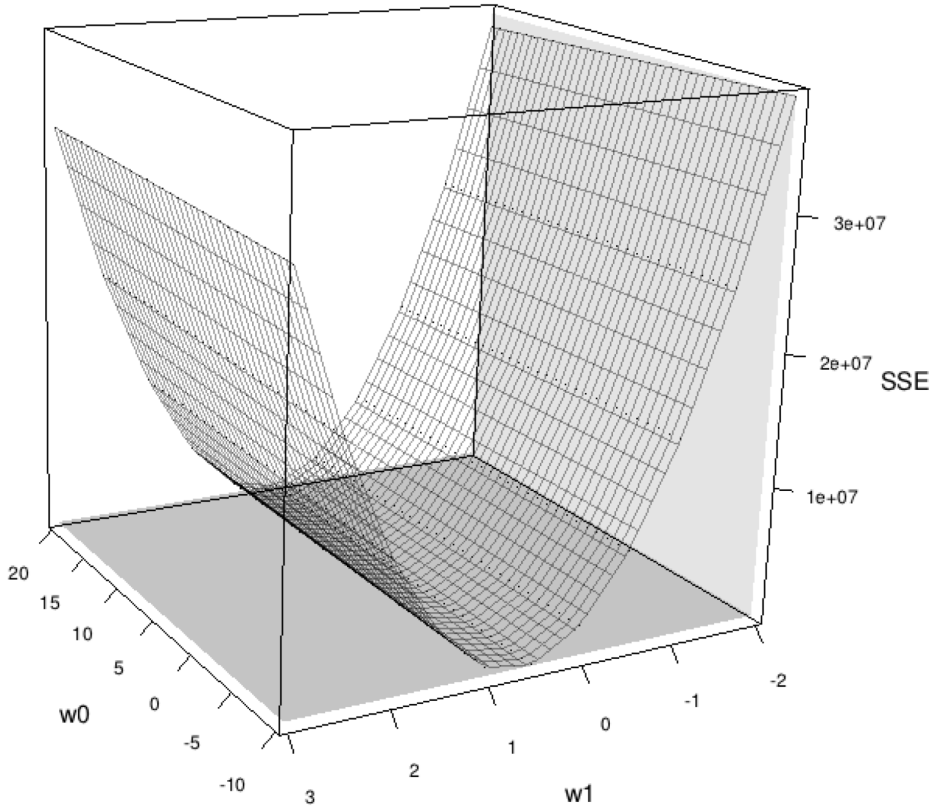
$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i[1]))^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}_i[1]))^2$$

The error surface shows the error in terms of weights of regression function.

Figure taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

How to find the lowest point?

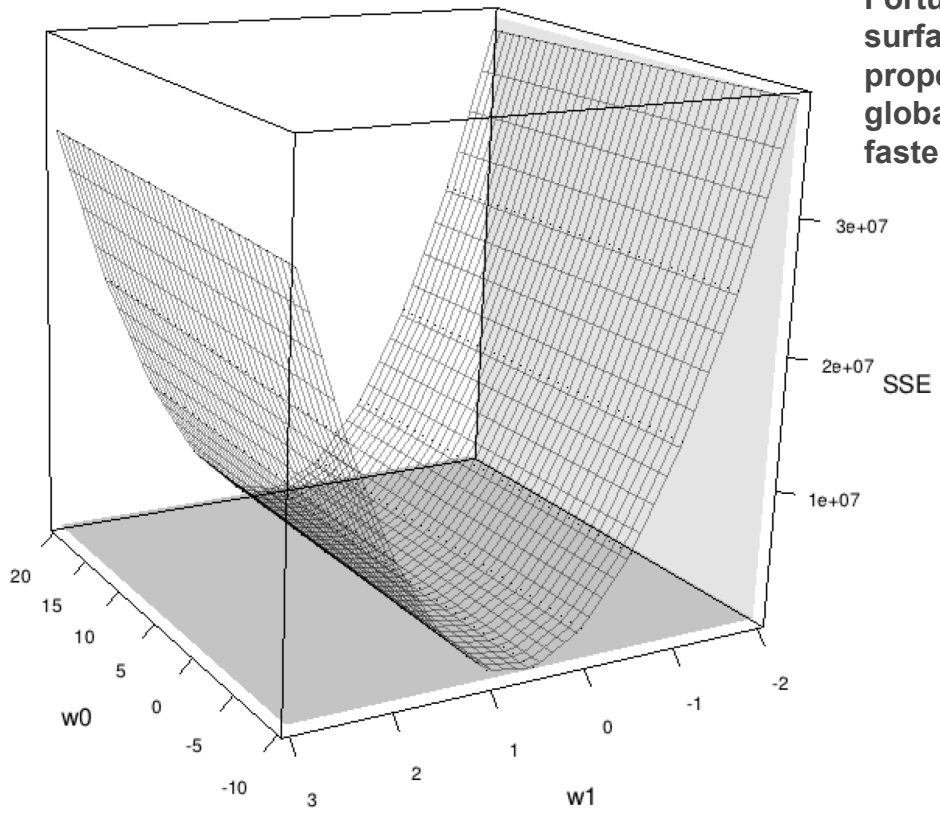


$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i[1]))^2$$
$$= \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}_i[1]))^2$$

Brute force (try many values for w[0] and w[1]) is not feasible in practice.

Fortunately, the error surface has particular properties (convex and global minimum) enabling faster methods.

How to find the lowest point?



Fortunately, the error surface has particular properties (convex and global minimum) enabling faster methods.

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i[1]))^2$$
$$= \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}_i[1]))^2$$

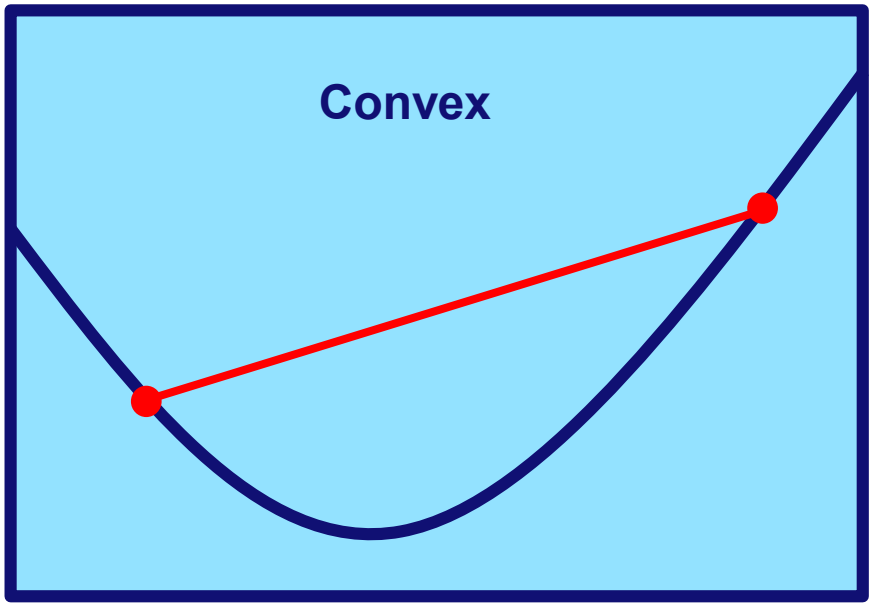
Partial derivatives in global minimum need to be equal to 0:

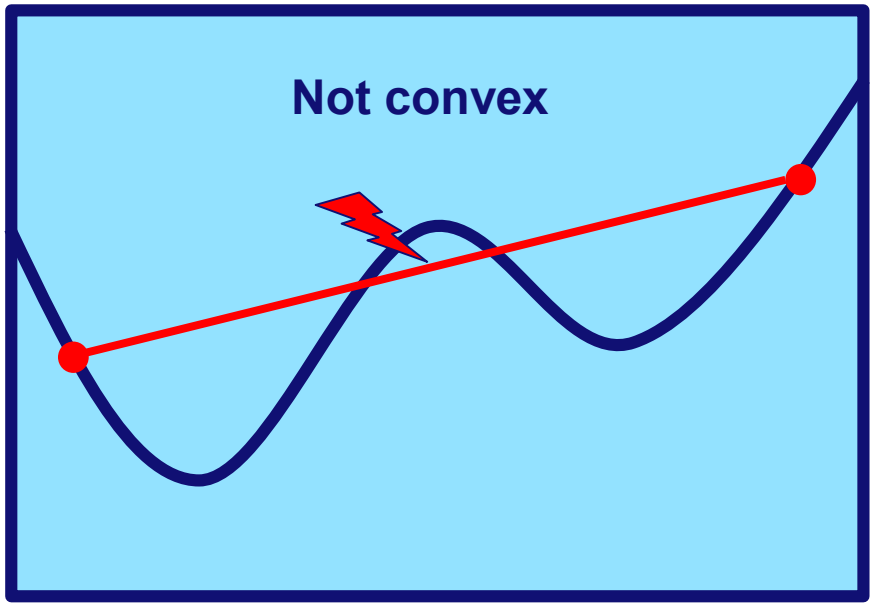
$$\frac{\partial}{\partial \mathbf{w}[0]}\frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}_i[1]))^2 = 0$$

$$\frac{\partial}{\partial \mathbf{w}[1]}\frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}_i[1]))^2 = 0$$

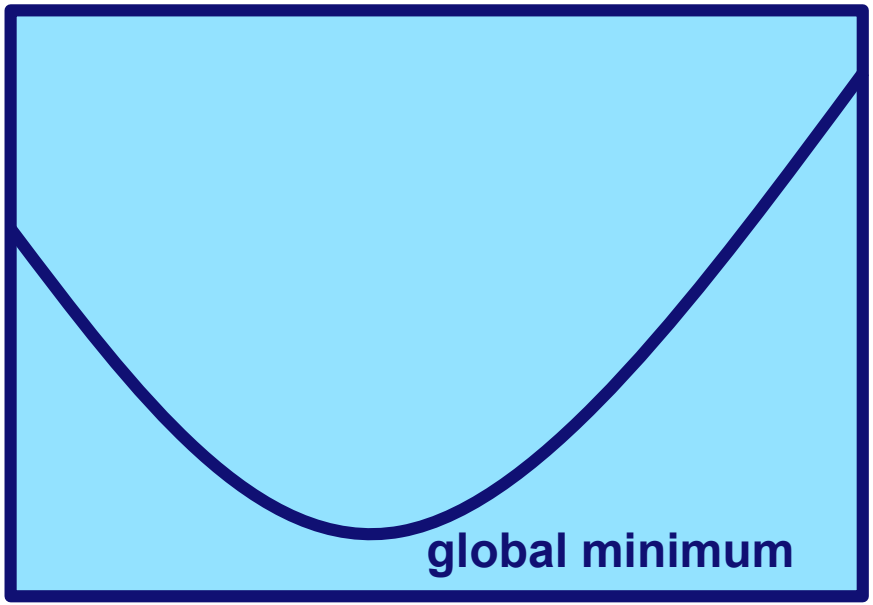

Finding the lowest point is easy when
the terrain is convex.

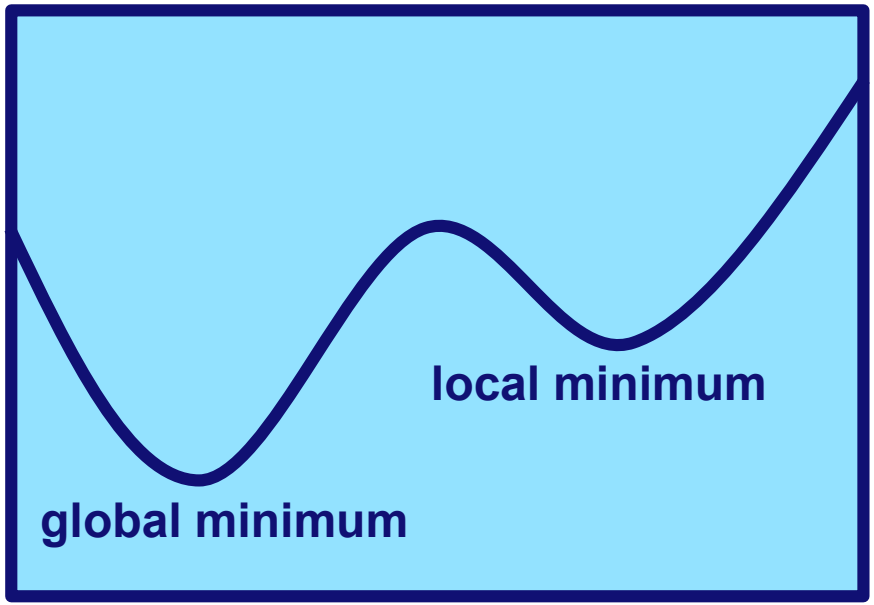
Convex versus non-convex





Convex: local minimum = global min.

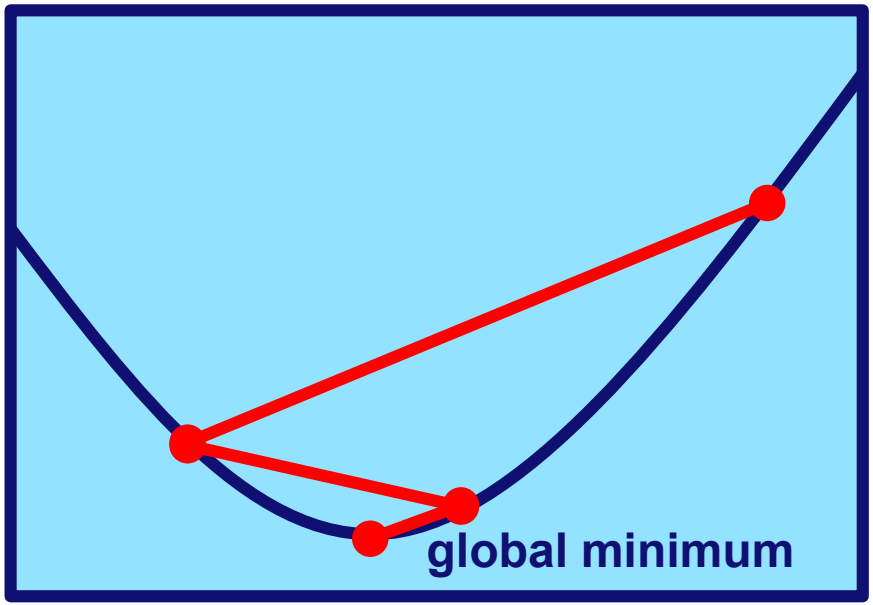


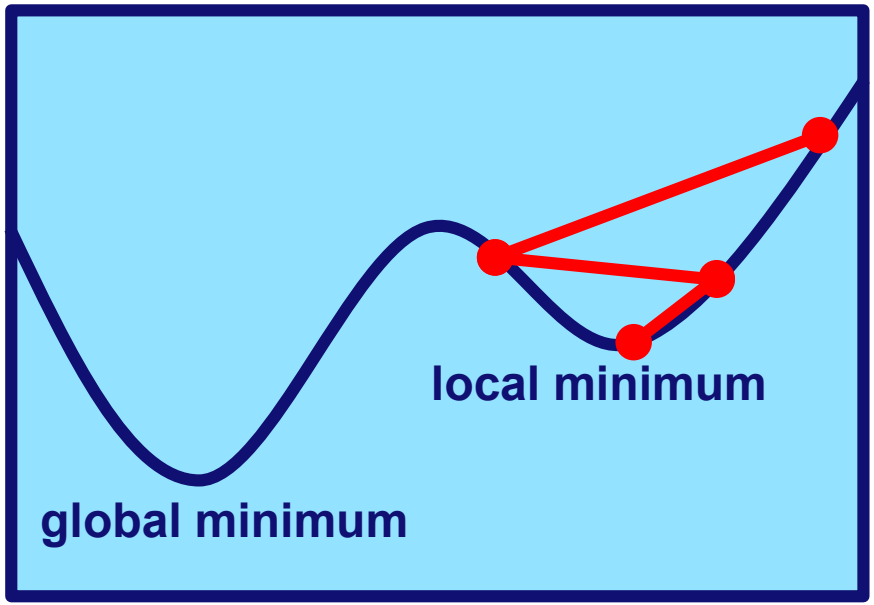


Easy to see: draw line between global and local optimum

Chair of Process and Data Science

Gradient descent





In which direction to walk?

The steepest way down.

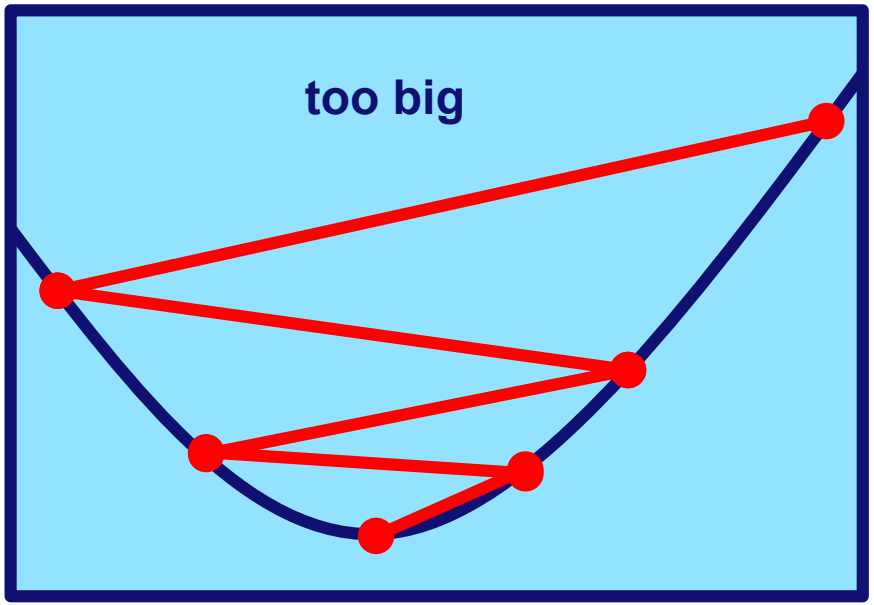
This is known because we have the derivative of the function.

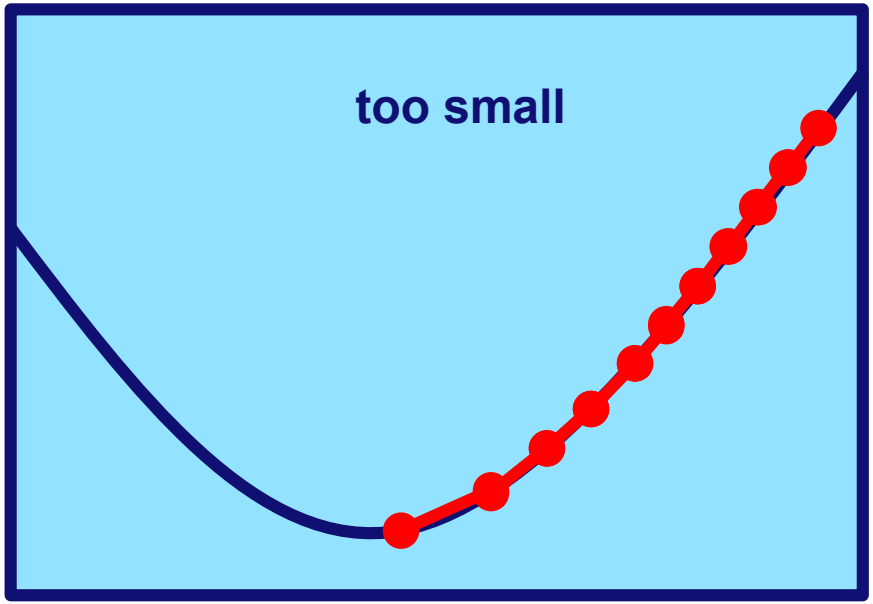
This will lead to a lower point and therefore converge.

Unknown: what step size?

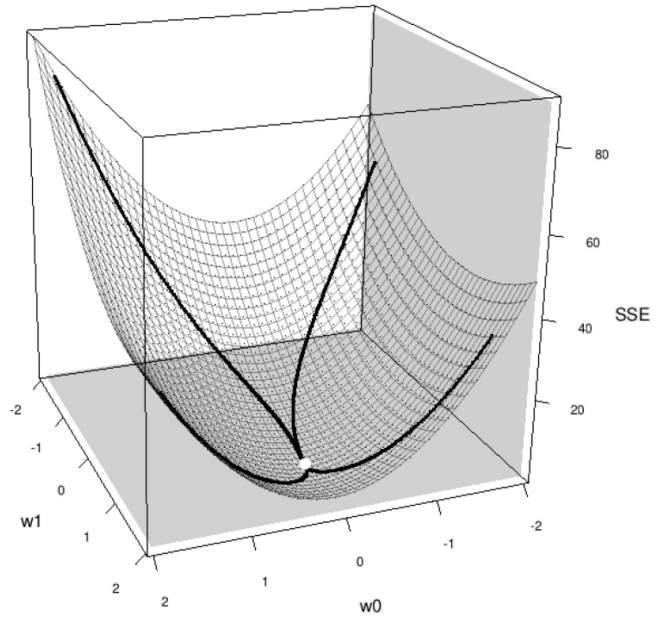
Chair of Process and Data Science

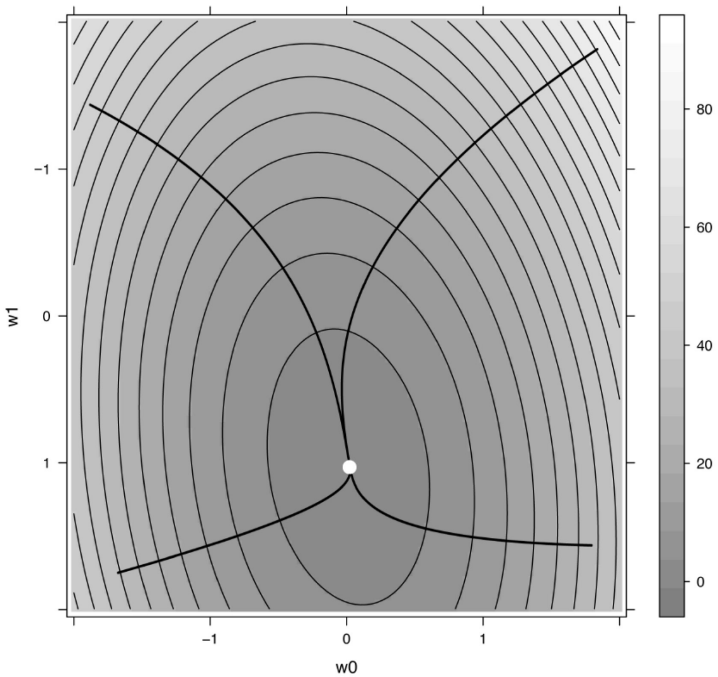
Gradient descent





Four different starting positions



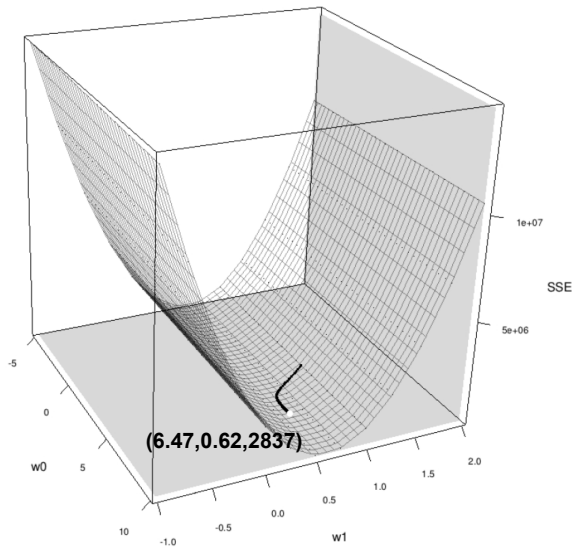


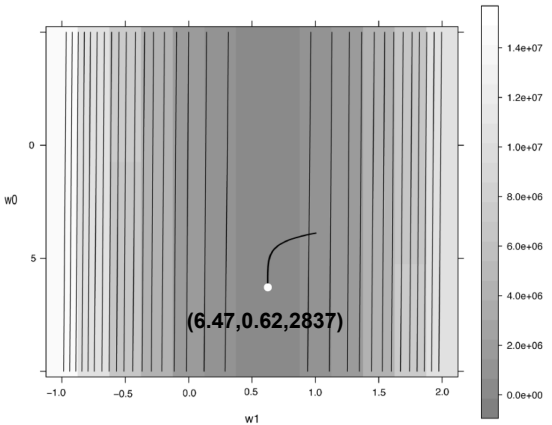
All end up (by definition) in the same global minimum.

Chair of Process and Data Science

Example path for the office rental data set

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620

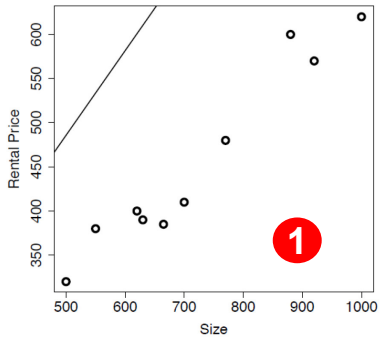


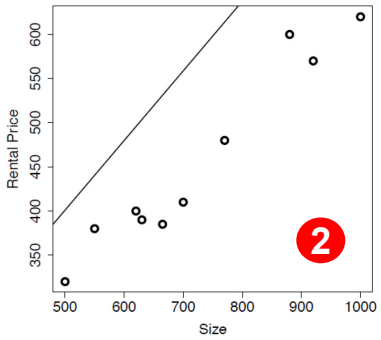


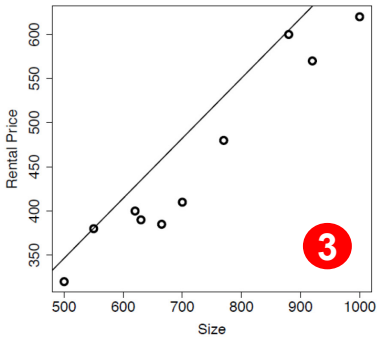
$$\text{Rental Price} = 6.47 + 0.62 \times \text{Size}$$

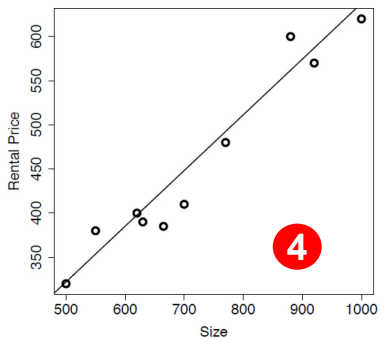
Chair of Process and Data Science

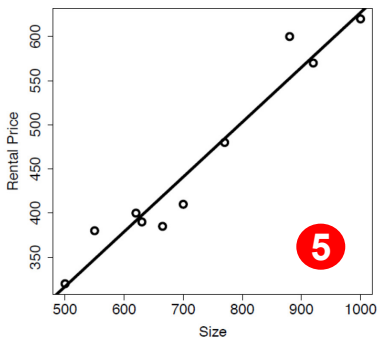
Example path for the office rental data set

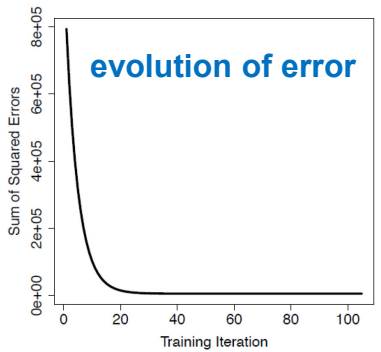












Not all steps are shown.

Chair of Process and Data Science

Multiple descriptive features

Nothing really new

$$
\begin{aligned}
\mathbb{M}_{\mathbf{w}}(\mathbf{d}) &= \mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}[1] + \cdots + \mathbf{w}[m] \times \mathbf{d}[m] \\
&= \mathbf{w}[0] + \sum_{j=1}^{m} \mathbf{w}[j] \times \mathbf{d}[j] \qquad \text{(w[i] is the weight of the i-th feature d[i])} \\
&= \sum_{j=0}^{m} \mathbf{w}[j] \times \mathbf{d}[j] \qquad \text{(d[0] = 1, just for notational convenience)} \\
&= \mathbf{w} \cdot \mathbf{d} \qquad \text{(dot product using d and w vectors)}
\end{aligned}
$$

Chair of Process and Data Science

Error function (having all the nice properties)

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2$$

$$= \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w} \cdot \mathbf{d}_i))^2$$

Revisiting the original data set

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

$$\text{Rental Price} = \mathbf{w}[0] + \mathbf{w}[1] \times \text{Size} + \mathbf{w}[2] \times \text{Floor} + \mathbf{w}[3] \times \text{Broadband Rate}$$

Revisiting the original data set

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

Optimal weights:

$$\mathbf{w}[0] = -0.1513,$$
$$\mathbf{w}[1] = 0.6270,$$
$$\mathbf{w}[2] = -0.1781,$$
$$\mathbf{w}[3] = 0.0714.$$

$$\text{Rental Price} = -0.1513 + 0.6270 \times \text{Size} - 0.1781 \times \text{Floor} + 0.0714 \times \text{Broadband Rate}$$

Sketch of overall algorithm

Require: set of training instances $\mathcal{D}$
Require: a learning rate α that controls how quickly the algorithm converges
Require: a function, **errorDelta**, that determines the direction in which to adjust a given weight, **w**[j], so as to move down the slope of an error surface determined by the dataset, $\mathcal{D}$
Require: a convergence criterion that indicates that the algorithm has completed

1: **w** $\leftarrow$ random starting point in the weight space
2: **repeat**
3: **for** each **w**[j] in **w** **do**
4: **w**[j] $\leftarrow$ **w**[j] $+ \alpha \times$ **errorDelta**($\mathcal{D}$, **w**[j])
5: **end for**
6: **until** convergence occurs

randomly pick initial point

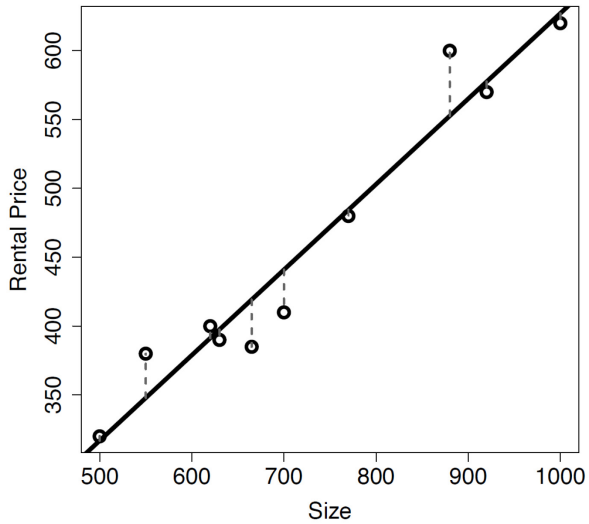
run downhill in the steepest direction with speed α

stop when improvements become too small

Interpretation of results

Interpretation is easy (one feature)

ID	Size	Rental Price
1	500	320
2	550	380
3	620	400
4	630	390
5	665	385
6	700	410
7	770	480
8	880	600
9	920	570
10	1,000	620



$$\text{Rental Price} = 6.47 + 0.62 \times \text{Size}$$

Interpretation not so easy (multiple features)

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

What if there are dozens of features with completely different ranges?

$$\text{Rental Price} = -0.1513 + 0.6270 \times \text{Size} - 0.1781 \times \text{Floor} + 0.0714 \times \text{Broadband Rate}$$

Interpretation not so easy (multiple features)

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

- **The weights change dramatically when the units change.**
- **This shows that the weight itself is irrelevant.**

(only the sign has a meaning)

cm^2 m^2

$$\text{Rental Price} = -0.1513 + 0.6270 \times \text{Size} - 0.1781 \times \text{Floor} + 0.0714 \times \text{Broadband Rate}$$

€ $ £ ₿ k€

Kbps, Mbps, Gbps

Alternative approach: significance test

- (No need to know the details.)
- Null hypothesis is that the feature does not have a significant effect on the model.
- Null hypothesis is rejected when p value is too small (e.g., 5% or 1%).

Descriptive Feature	Weight	Standard Error	*t*-statistic	*p*-value
SIZE	0.6270	0.0545	11.504	<0.0001
FLOOR	-0.1781	2.7042	-0.066	0.949
BROADBAND RATE	0.071396	0.2969	0.240	0.816

only Size is not rejected

Handling categorical features

Dealing with categorical variables

categorical descriptive features

categorical target feature

f1	f2	f3	f4	…	fn	class
high	true	gold	88		59.99	A
high	false	gold	76		50.00	B
low	false	silver	32		39.50	B
low	true	silver	89		49.99	C
high	true	gold	21		59.99	C
low	true	gold	45		29.99	A

Thus far we assumed features were continuous.

Categorical descriptive features

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

values A, B, and C

General approach: introduce a 0/1 feature for every possible value.

- **A = (1,0,0)**
- **B = (0,1,0)**
- **C = (0,0,1)**

Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Chair of Process and Data Science

Categorical descriptive features

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

$$\begin{aligned}\text{Rental Price} = \mathbf{w}[0] &+ \mathbf{w}[1] \times \text{Size} + \mathbf{w}[2] \times \text{Floor} \\ &+ \mathbf{w}[3] \times \text{Broadband Rate} \\ &+ \mathbf{w}[4] \times \text{Energy Rating A} \\ &+ \mathbf{w}[5] \times \text{Energy Rating B} \\ &+ \mathbf{w}[6] \times \text{Energy Rating C}\end{aligned}$$

One Hot Encoding

ID	Size	Floor	Broadband Rate	Energy Rating A	Energy Rating B	Energy Rating C	Rental Price
1	500	4	8	0	0	1	320
2	550	7	50	1	0	0	380
3	620	9	7	1	0	0	400
4	630	5	24	0	1	0	390
5	665	8	100	0	0	1	385
6	700	4	8	0	1	0	410
7	770	10	7	0	1	0	480
8	880	12	50	1	0	0	600
9	920	14	8	0	0	1	570
10	1 000	9	24	0	1	0	620

Categorical descriptive features

Next to One Hot Encoding:

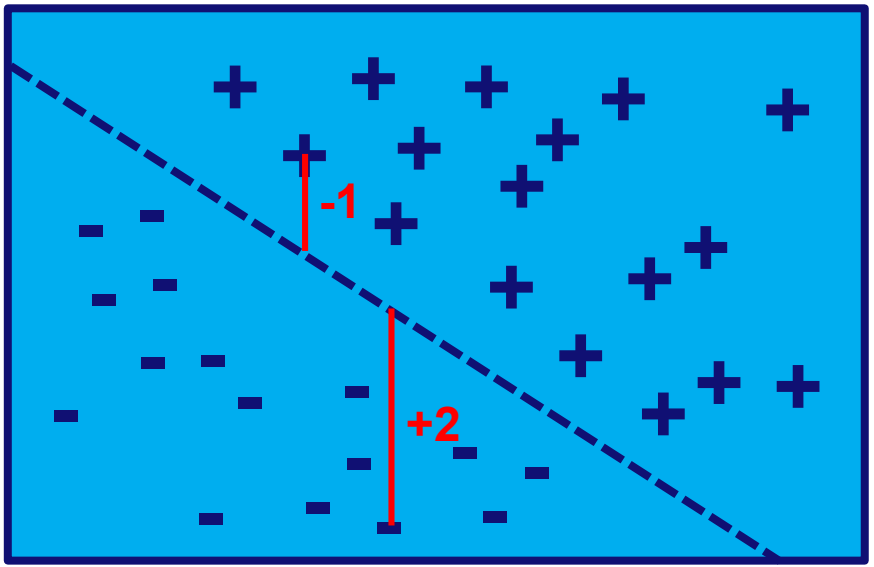
- **Binary values (true, false) can be translated to a single numeric value (0 or 1).**
- **Categorical variables with a clear order (i.e., ordinal) can be translated to a single numeric value.**
  (e.g., excellent = 1.0, good = 0.7, average = 0.5, poor = 0.3, horrible = 0.0)

Possible issues

- Adding order to unordered categorical variables (e.g., simple encoding) is dangerous (e.g., mapping country names onto numbers).
- Of course all encodings are approximations and also intermediate values will be considered possible by the “regression machine”.
- In One Hot Encoding dependencies are missing. If A=1, then logically B=0, but also B=0.66 is possible.
- The approach may also introduce many additional features (computationally expensive).

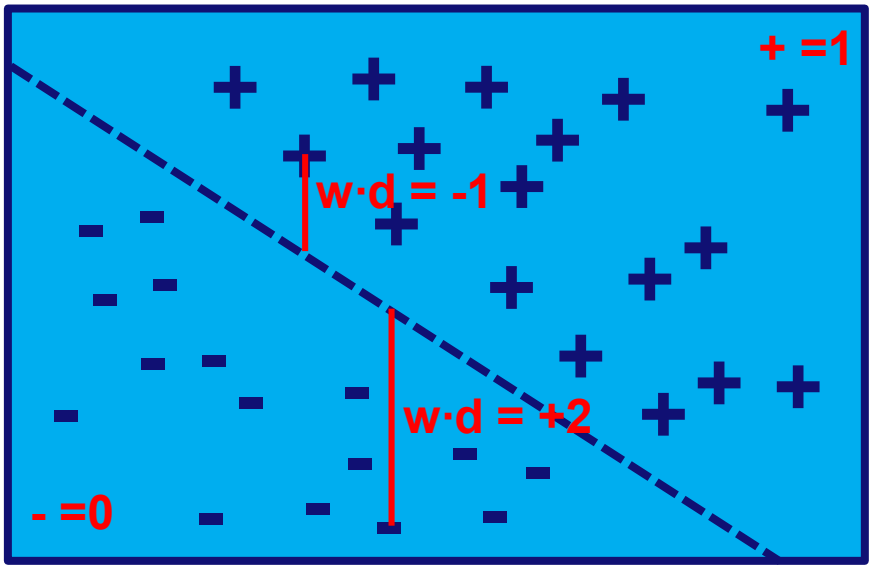
Categorical target features

x	y	class
2.1	2.0	+
3.1	1.4	+
1.2	0.8	-
2.9	3.3	+
0.8	1.1	-
…	…	..



Line is used to separate rather than predict.

Categorical target features: Naïve approach



- If w·d ≥ 0 and -, then squared error is 0.
- If w·d < 0 and +, then squared error is 0.
- If w·d ≥ 0 and +, then squared error is 1.
- If w·d < 0 and -, then squared error is 1.

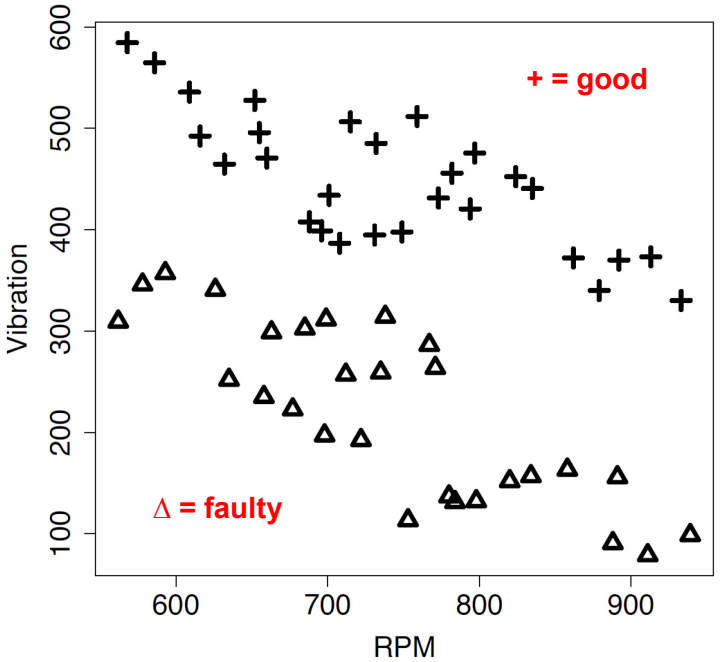
- Idea: use 0 and 1 for + and -.
- Use $$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$
- w·d indicates distance (e.g. -1 and +2) from the line based on weights w.
- Optimize w such that the sum of squared errors is minimized.

Note that w·d is negative if above line and w·d is positive if below line.

Example from book

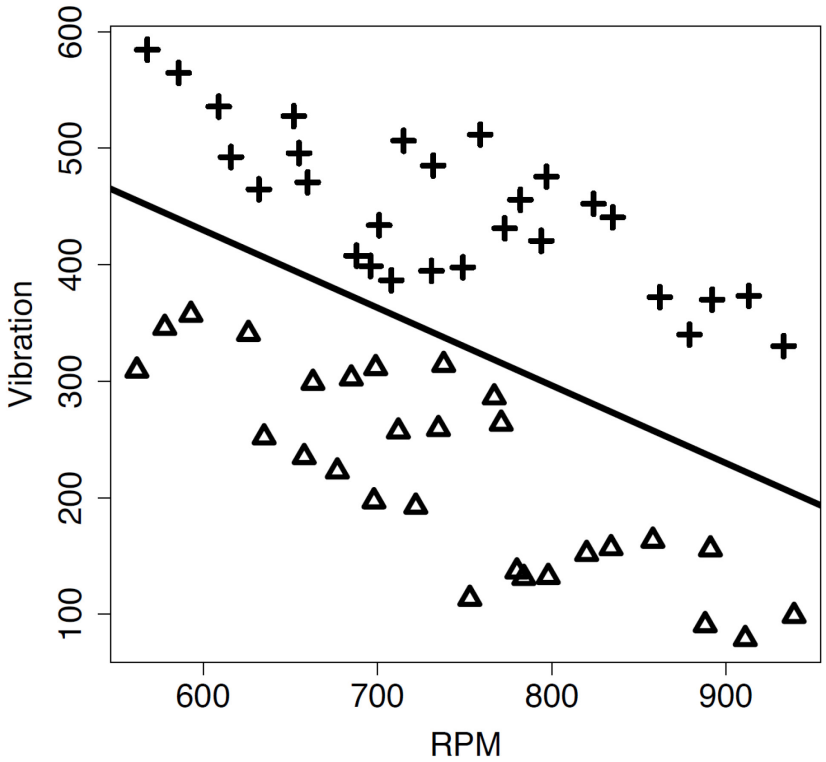
ID	RPM	VIBRATION	STATUS
1	568	585	good
2	586	565	good
3	609	536	good
4	616	492	good
5	632	465	good
6	652	528	good
7	655	496	good
8	660	471	good
9	688	408	good
10	696	399	good
11	708	387	good
12	701	434	good
13	715	506	good
14	732	485	good
15	731	395	good
16	749	398	good
17	759	512	good
18	773	431	good
19	782	456	good
20	797	476	good
21	794	421	good
22	824	452	good
23	835	441	good
24	862	372	good
25	879	340	good
26	892	370	good
27	913	373	good
28	933	330	good

ID	RPM	VIBRATION	STATUS
29	562	309	faulty
30	578	346	faulty
31	593	357	faulty
32	626	341	faulty
33	635	252	faulty
34	658	235	faulty
35	663	299	faulty
36	677	223	faulty
37	685	303	faulty
38	698	197	faulty
39	699	311	faulty
40	712	257	faulty
41	722	193	faulty
42	735	259	faulty
43	738	314	faulty
44	753	113	faulty
45	767	286	faulty
46	771	264	faulty
47	780	137	faulty
48	784	131	faulty
49	798	132	faulty
50	820	152	faulty
51	834	157	faulty
52	858	163	faulty
53	888	91	faulty
54	891	156	faulty
55	911	79	faulty
56	939	99	faulty



Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Example from book



Let's just assume that we have line separating instances:

$$\text{VIBRATION} = 830 - 0.667 \times \text{RPM}$$

Rewrite this to

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} = 0$$

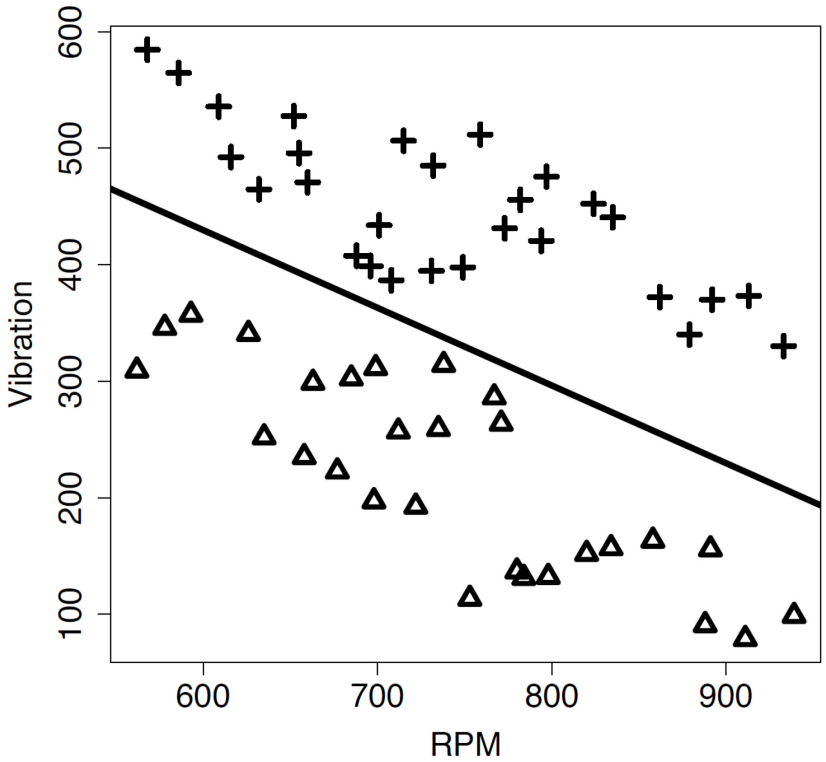
Below line

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} > 0$$

Above line

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} < 0$$

Example from book



Now take:

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} \geq 0$$

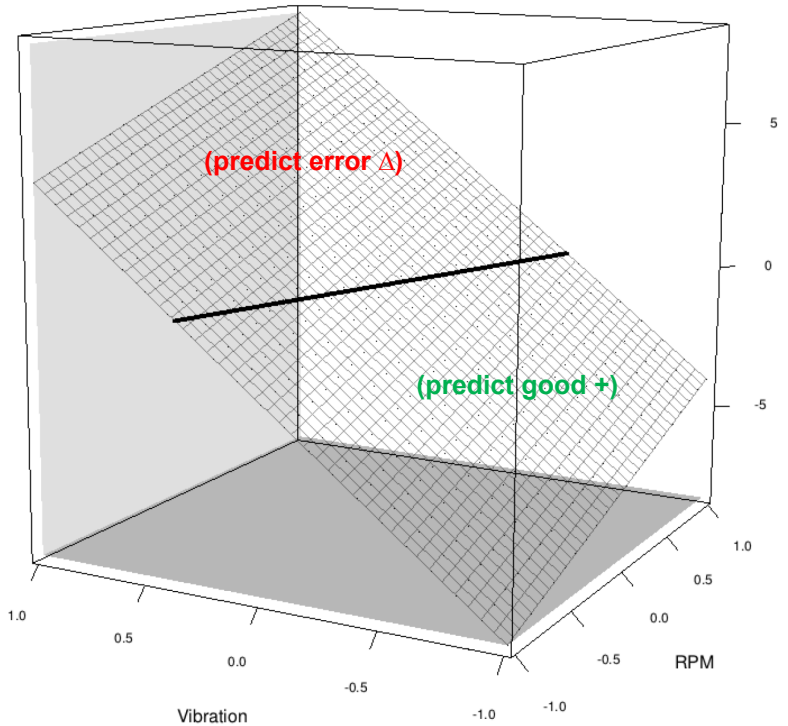
$\Rightarrow$ **1** **(predict error Δ)**

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} < 0$$

$\Rightarrow$ **0** **(predict good +)**

Example from book

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION}$$



Now take:

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} \geq 0$$

$\Rightarrow$ **1** **(predict error Δ)**

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} < 0$$

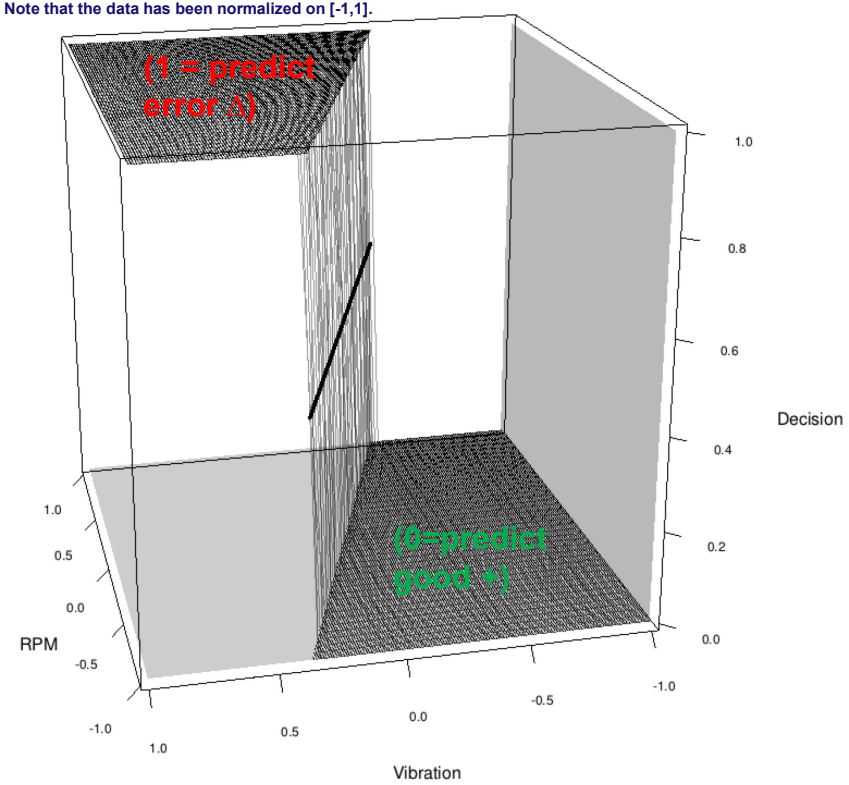
$\Rightarrow$ **0** **(predict good +)**

Note that the data has been normalized on [-1,1].

Chair of Process and Data Science

Example from book

Note that the data has been normalized on [-1,1].



Now take:

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

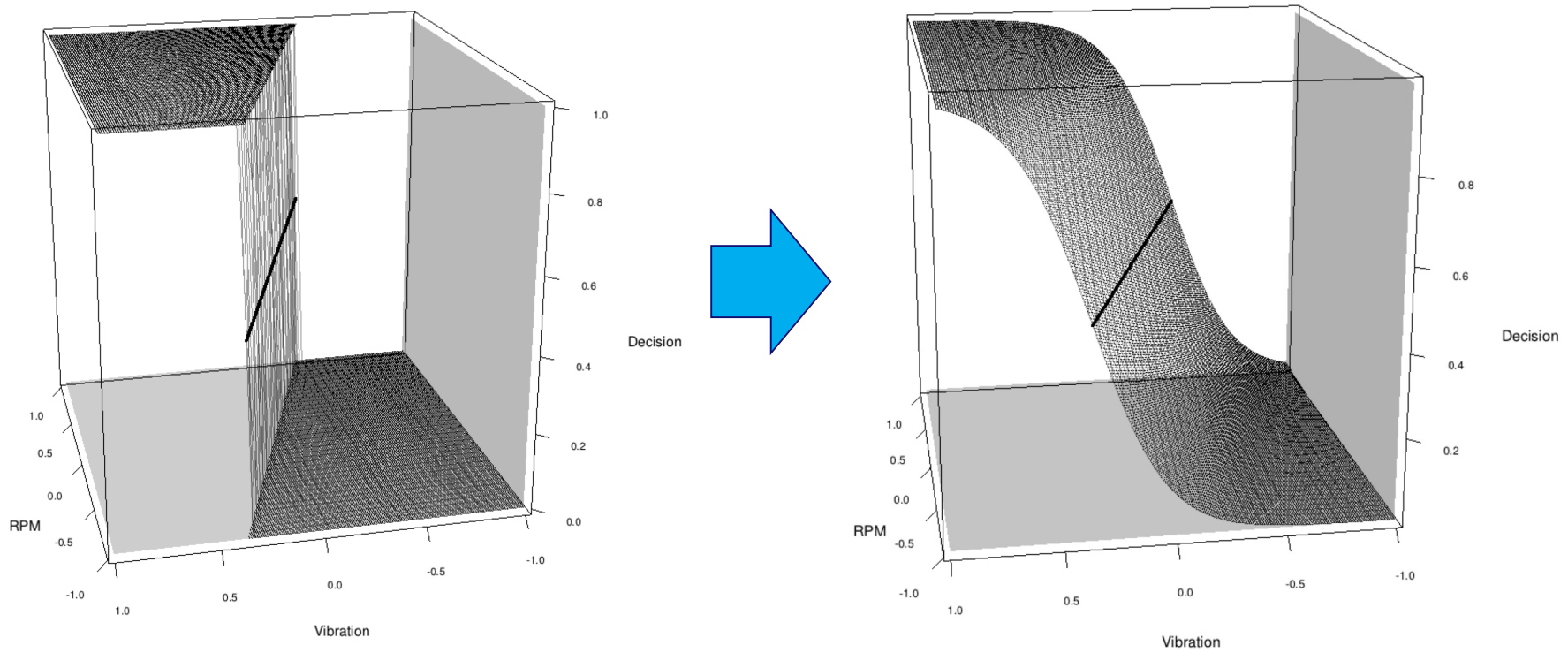
$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} \geq 0$$

$\Rightarrow$ **1** **(predict error Δ)**

$$830 - 0.667 \times \text{RPM} - \text{VIBRATION} < 0$$

$\Rightarrow$ **0** **(predict good +)**

Problem: Decision surface is discontinuous

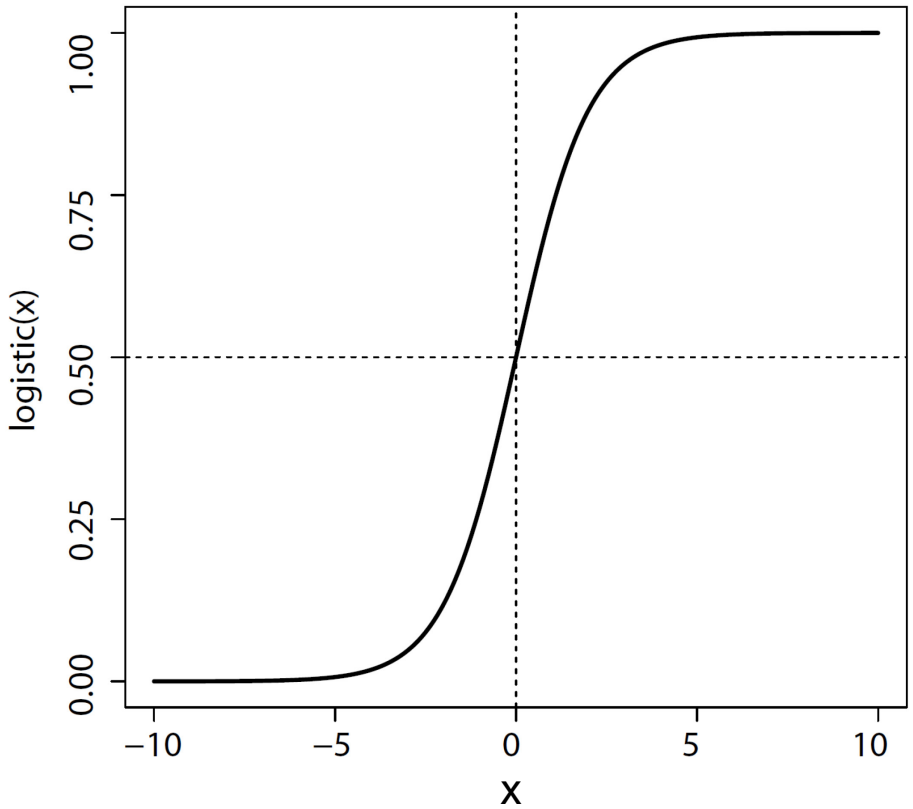


How to make it smooth: Logistic regression!

Chair of Process and Data Science

Logistic regression

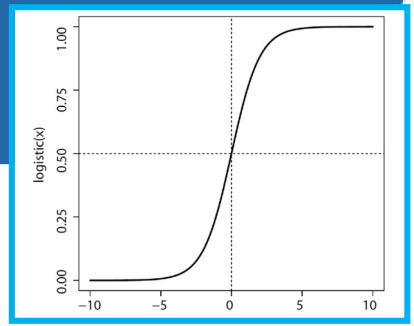
Logistic function



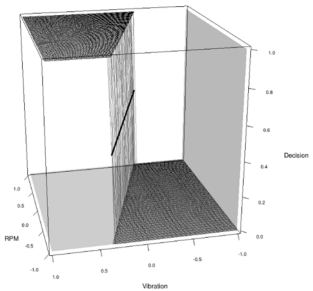
$$Logistic(x) = \frac{1}{1 + e^{-x}}$$

- e=2.7182818… Eulers number
- Any value is mapped on a value between 0 and 1:
  - logistic(0) = 0.5
  - logistic(-inf) = 0
  - logistic(+inf) = 1
- Quickly approaches 0 and 1 and can therefore serve as a “smooth binary value”.

Using the logistic function

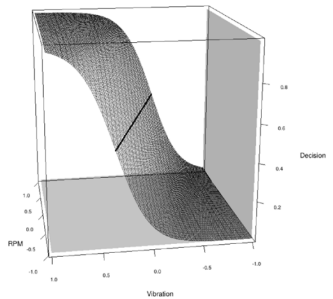


$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$



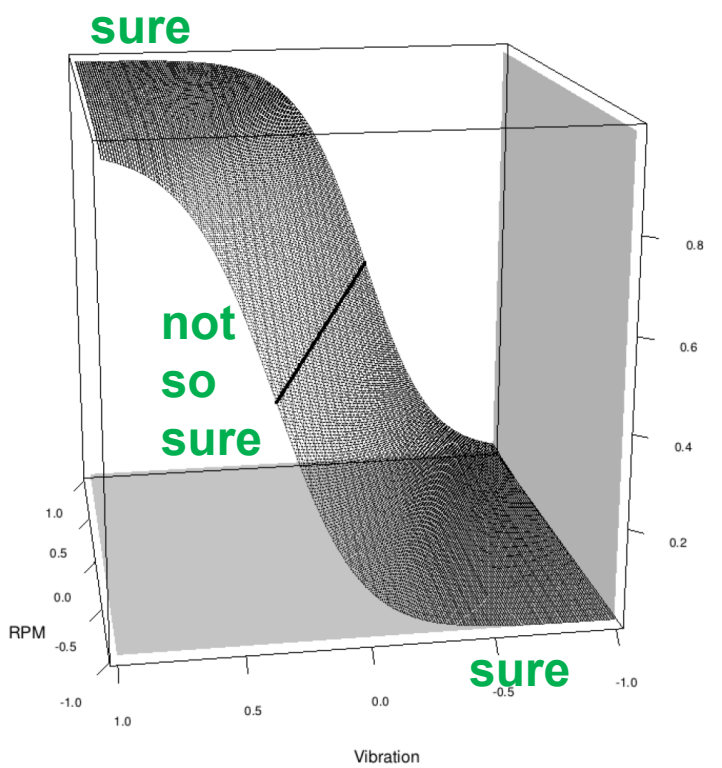
$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \textit{Logistic}(\mathbf{w} \cdot \mathbf{d})$$

$$= \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{d}}}$$



Chair of Process and Data Science

Using the logistic function



$$\mathbb{M}_{\mathbf{w}}(\langle \text{RPM}, \text{VIBRATION} \rangle) = \frac{1}{1 + e^{-(-0.4077 + 4.1697 \times \text{RPM} + 6.0460 \times \text{VIBRATION})}}$$

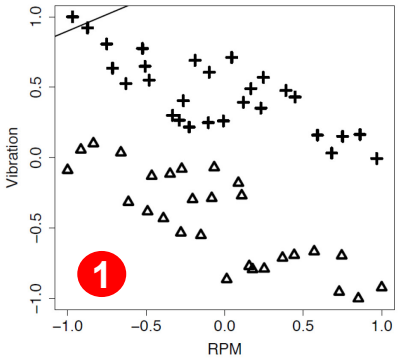
Probabilistic interpretation:

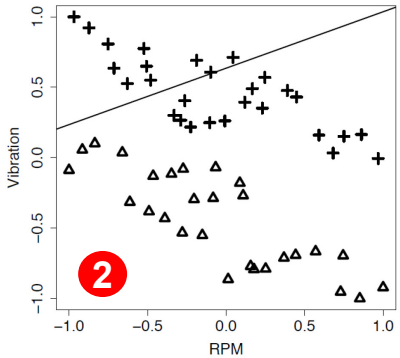
$$P(t = \textit{'faulty'} | \mathbf{d}) = \mathbb{M}_{\mathbf{w}}(\mathbf{d})$$

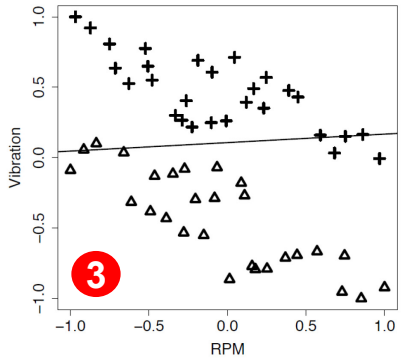
$$P(t = \textit{'good'} | \mathbf{d}) = 1 - \mathbb{M}_{\mathbf{w}}(\mathbf{d})$$

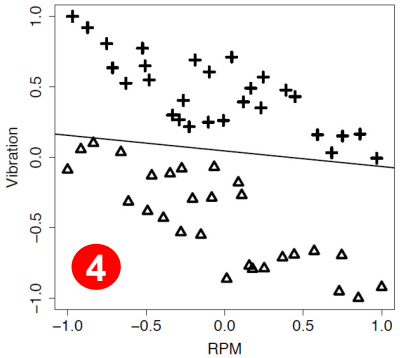
Note that the data has been normalized on [-1,1].

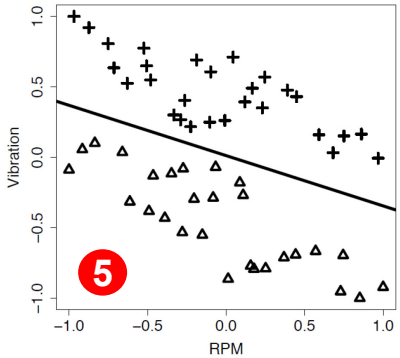
Gradient descend process

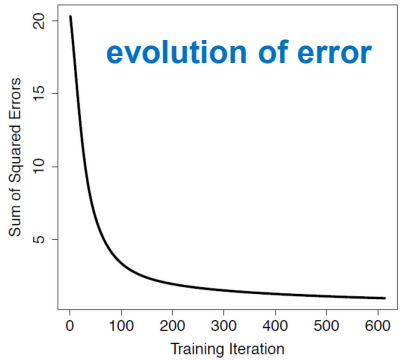












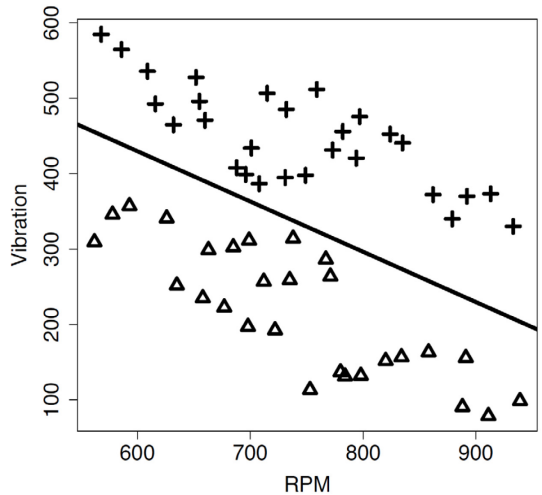
Selection of intermediate models.

Chair of Process and Data Science

Summary: Logistic regression

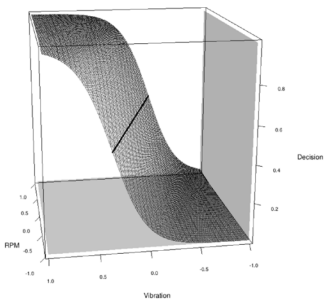
- **Input: continuous descriptive features and a binary categorical target feature.**
- **Output:**

A binary classifier w·d:



A indication of certainty [0,1]:

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = Logistic(\mathbf{w} \cdot \mathbf{d}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{d}}}$$



Chair of Process and Data Science

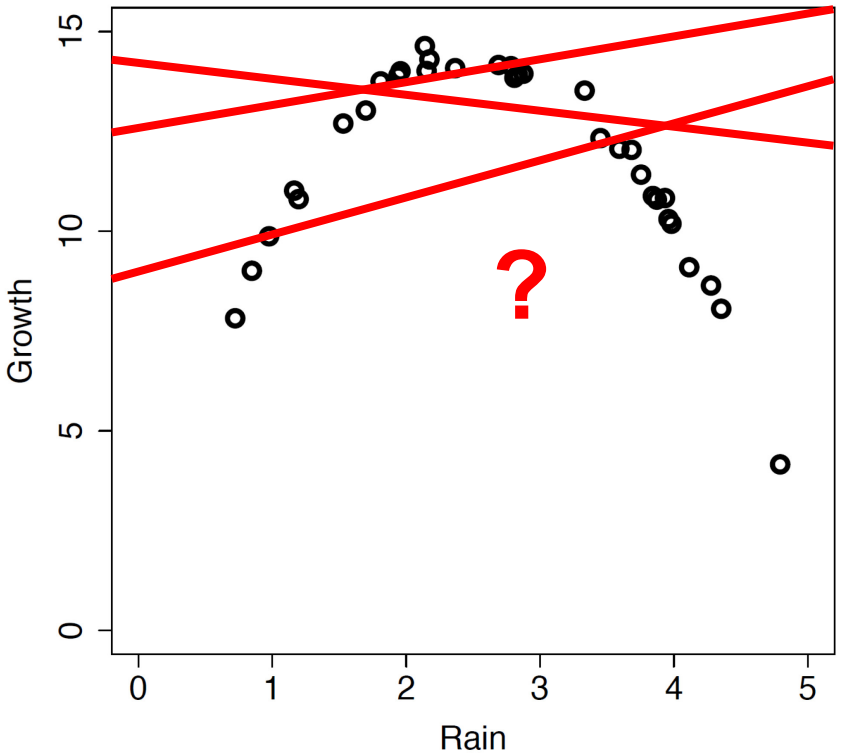
Extensions: non-linear and multinomial

Non-linear relationships

ID	Rain	Growth	ID	Rain	Growth	ID	Rain	Growth
1	2.153	14.016	12	3.754	11.420	23	3.960	10.307
2	3.933	10.834	13	2.809	13.847	24	3.592	12.069
3	1.699	13.026	14	1.809	13.757	25	3.451	12.335
4	1.164	11.019	15	4.114	9.101	26	1.197	10.806
5	4.793	4.162	16	2.834	13.923	27	0.723	7.822
6	2.690	14.167	17	3.872	10.795	28	1.958	14.010
7	3.982	10.190	18	2.174	14.307	29	2.366	14.088
8	3.333	13.525	19	4.353	8.059	30	1.530	12.701
9	1.942	13.899	20	3.684	12.041	31	0.847	9.012
10	2.876	13.949	21	2.140	14.641	32	3.843	10.885
11	4.277	8.643	22	2.783	14.138	33	0.976	9.876

Example taken from Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.

Non-linear relationships



ID	Rain	Growth	ID	Rain	Growth	ID	Rain	Growth
1	2.153	14.016	12	3.754	11.420	23	3.960	10.307
2	3.933	10.834	13	2.809	13.847	24	3.592	12.069
3	1.699	13.026	14	1.809	13.757	25	3.451	12.335
4	1.164	11.019	15	4.114	9.101	26	1.197	10.806
5	4.793	4.162	16	2.834	13.923	27	0.723	7.822
6	2.690	14.167	17	3.872	10.795	28	1.958	14.010
7	3.982	10.190	18	2.174	14.307	29	2.366	14.088
8	3.333	13.525	19	4.353	8.059	30	1.530	12.701
9	1.942	13.899	20	3.684	12.041	31	0.847	9.012
10	2.876	13.949	21	2.140	14.641	32	3.843	10.885
11	4.277	8.643	22	2.783	14.138	33	0.976	9.876

How to still use the “linear machinery” and still handle non-linear functions?

Basic idea: transform data before

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \sum_{k=0}^{b} \mathbf{w}[k] \times \phi_k(\mathbf{d})$$

- **In order to handle non-linear relationships, the data is transformed *before* the "linear machinery" is used.**
- **Several "basis functions" are used to transform descriptive features.**

$$\begin{aligned} \phi_0(\text{RAIN}) &= 1 \\ \phi_1(\text{RAIN}) &= \text{RAIN} \\ \phi_2(\text{RAIN}) &= \text{RAIN}^2 \end{aligned}$$

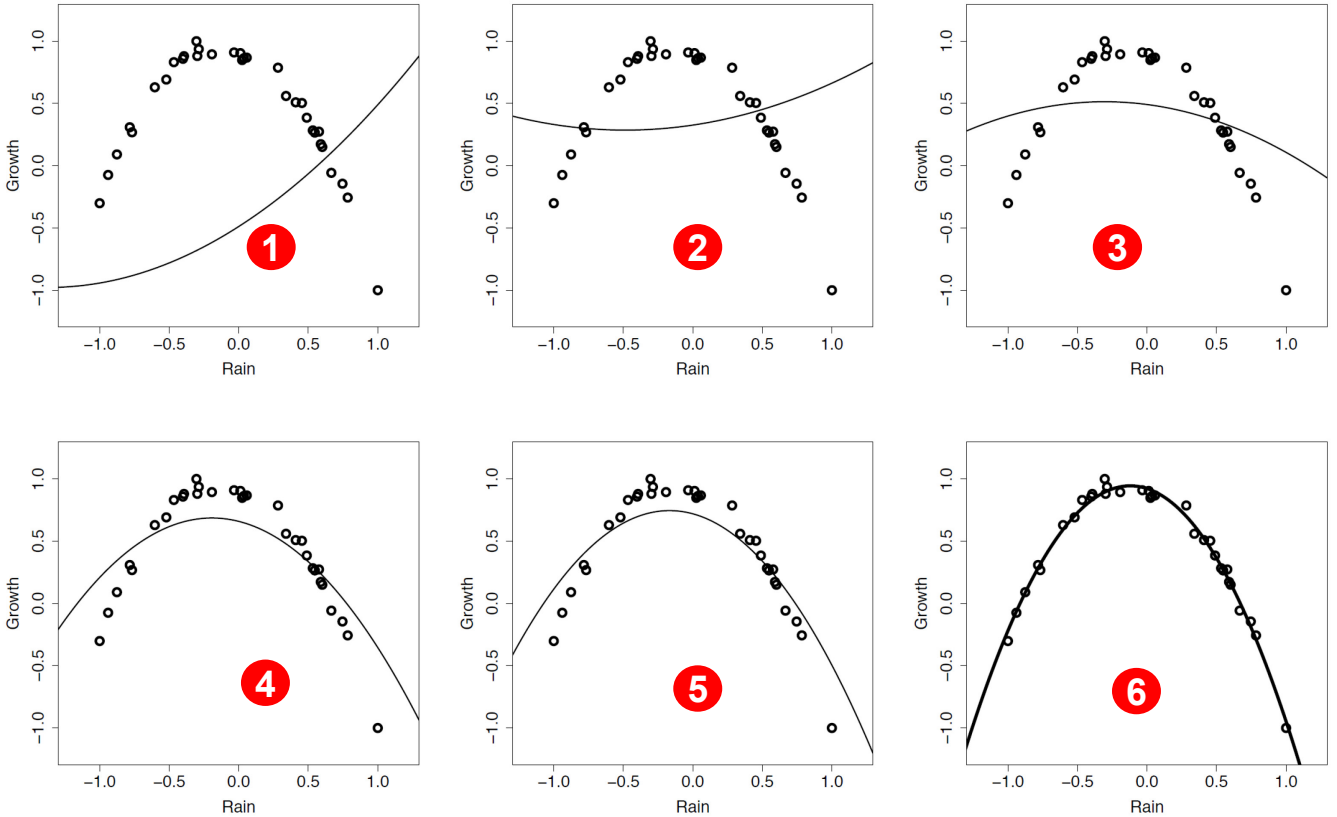
$$\text{GROWTH} = \mathbf{w}[0] \times \phi_0(\text{RAIN}) + \mathbf{w}[1] \times \phi_1(\text{RAIN}) + \mathbf{w}[2] \times \phi_2(\text{RAIN})$$

It works!

$\text{GROWTH} = \mathbf{w}[0] \times \phi_0(\text{RAIN}) + \mathbf{w}[1] \times \phi_1(\text{RAIN}) + \mathbf{w}[2] \times \phi_2(\text{RAIN})$

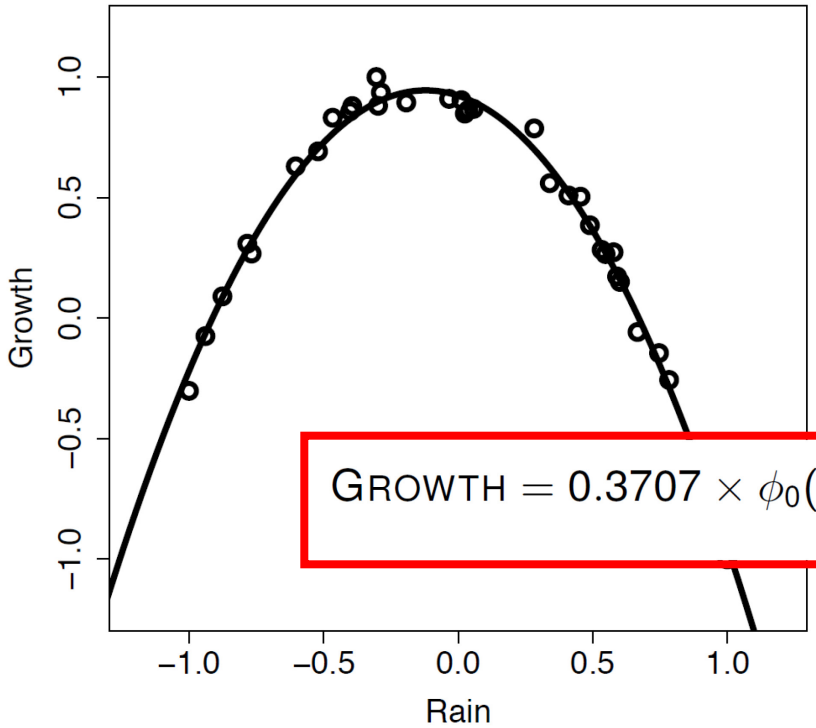
$$\phi_0(\text{RAIN}) = 1$$
$$\phi_1(\text{RAIN}) = \text{RAIN}$$
$$\phi_2(\text{RAIN}) = \text{RAIN}^2$$

Second-order polynomial.



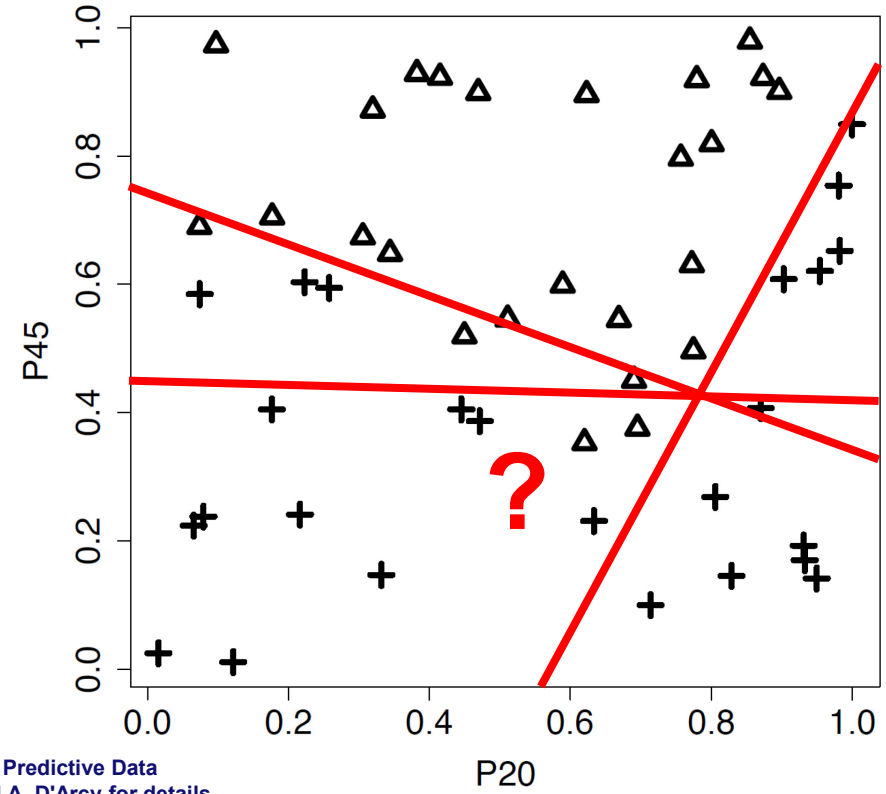
Note only selected steps are shown and normalized data.

Final model



$$\text{GROWTH} = 0.3707 \times \phi_0(\text{RAIN}) + 0.8475 \times \phi_1(\text{RAIN}) + -1.717 \times \phi_2(\text{RAIN})$$

Also for logistic regression

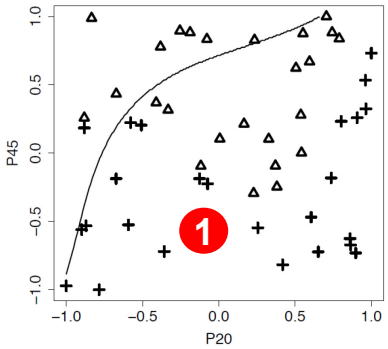


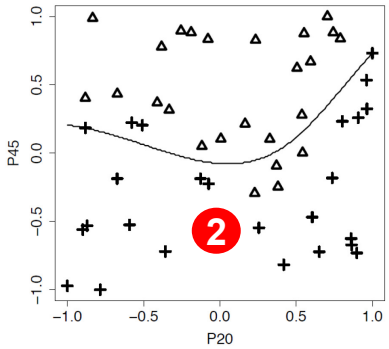
How to separate?

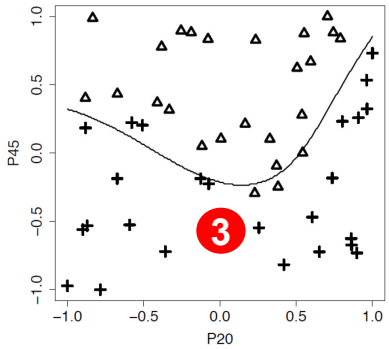
See Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy for details.

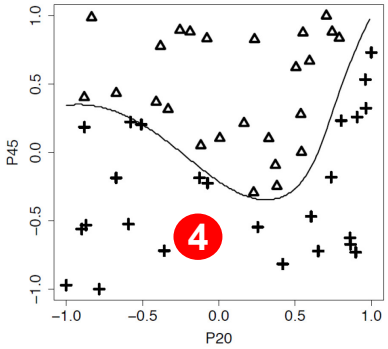
Chair of Process and Data Science

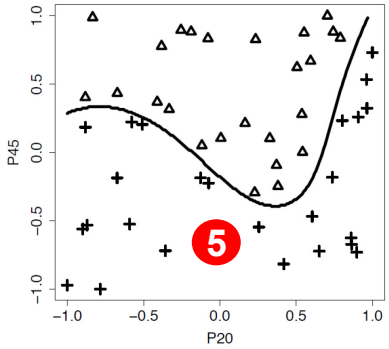
Logistic regression in action

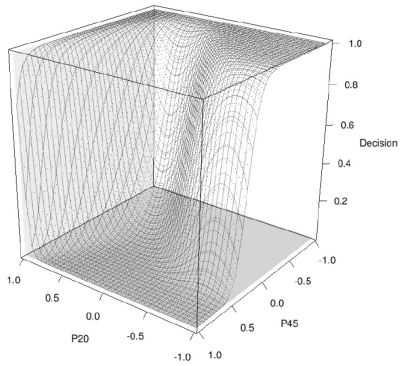












Note: only selected steps are shown and normalized data.

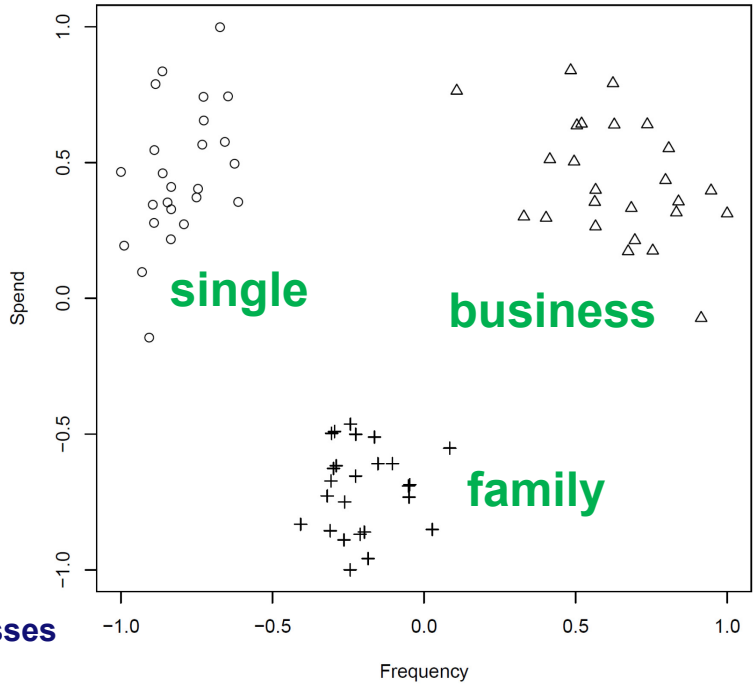
Chair of Process and Data Science

Multinomial regression

How to handle categorical data that is not binary?

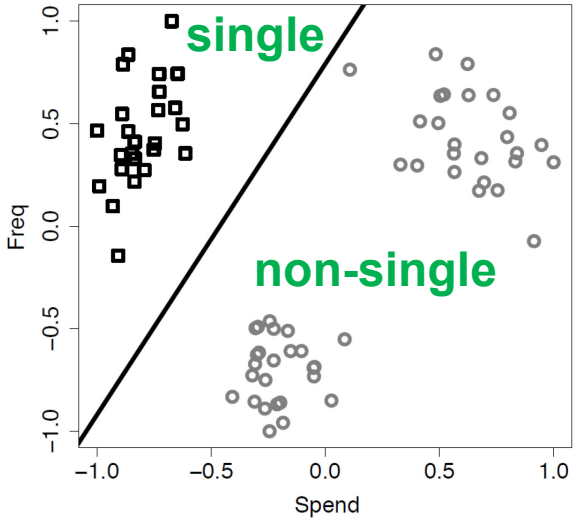
ID	Spend	Freq	Type
1	21.6	5.4	single
2	25.7	7.1	single
3	18.9	5.6	single
4	25.7	6.8	single
⋮			
26	107.9	5.8	business
27	92.9	5.5	business

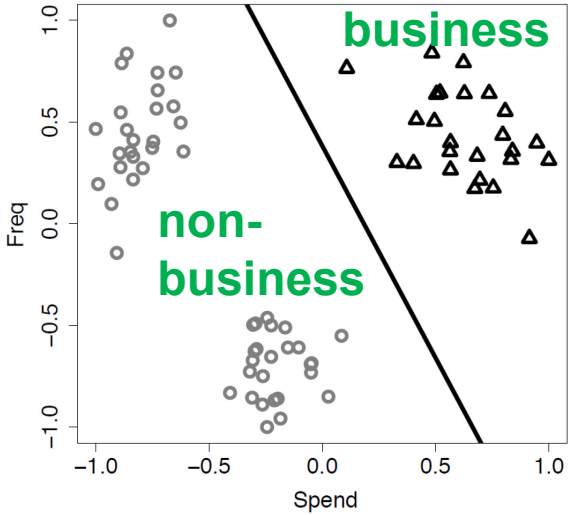
ID	Spend	Freq	Type
28	122.6	6.0	business
29	107.7	5.7	business
⋮			
47	53.2	2.6	family
48	52.4	2.0	family
49	46.1	1.4	family
50	65.3	2.2	family

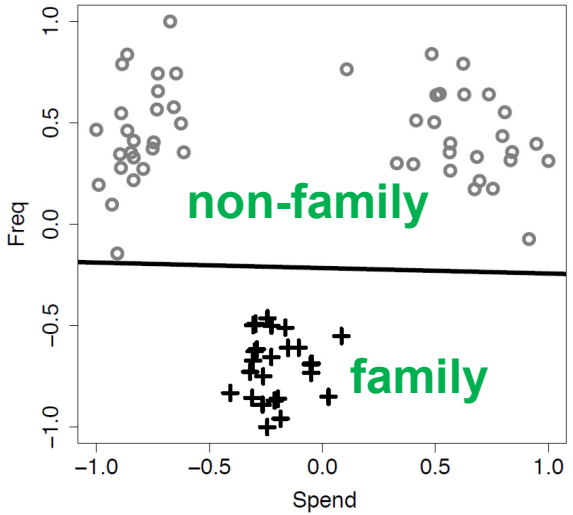


three classes

Make binary with one-versus-rest



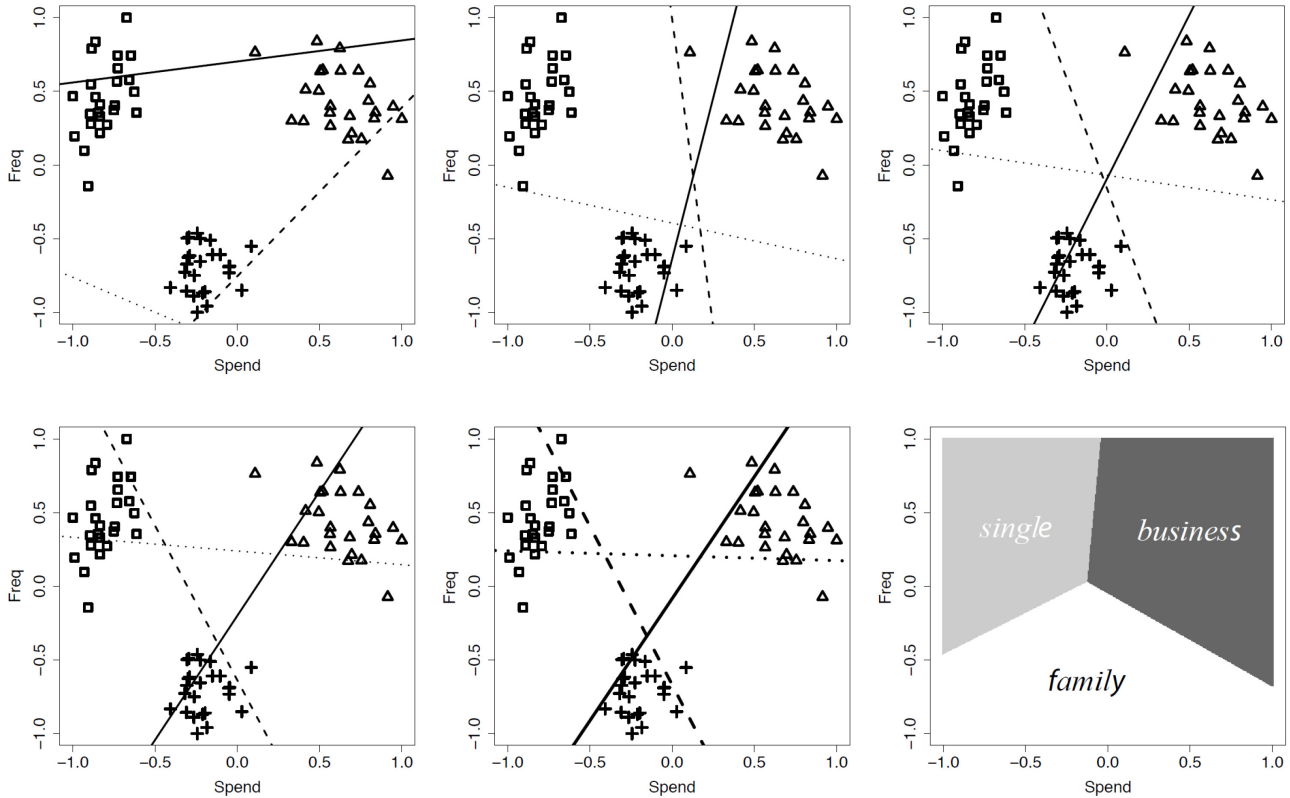




Normalize the outputs of the three models and for a specific instance pick the best scoring one. (See book for details.)

Chair of Process and Data Science

Resulting model



Note: only selected steps are shown and normalized data.

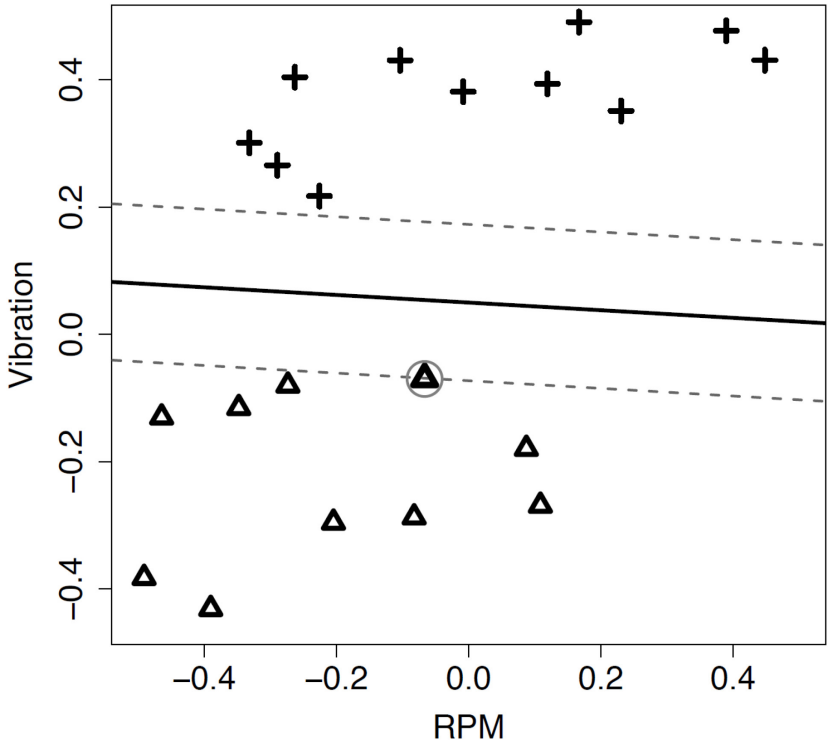
Chair of Process and Data Science

Conclusion

Short summary of lecture

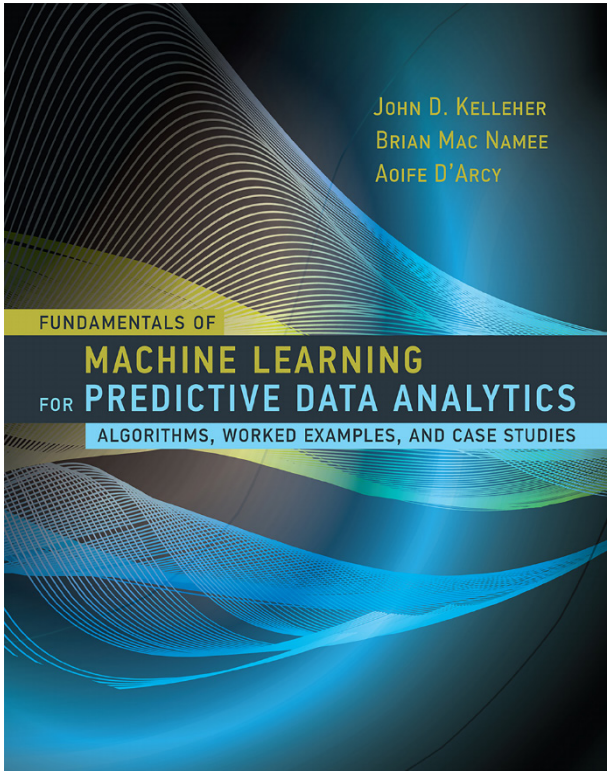
- Regression is based on error-based learning: Search for the model with the smallest error.
- For supervised learning only.
- Good support for continuous features.
- Extensions to categorical variables possible (e.g. logistic regression).

Next: Support Vector Machines



Chair of Process and Data Science

Relevant Literature



- **Chapter 7 of Fundamentals of Machine Learning for Predictive Data Analytics by J. Kelleher, B. Mac Namee and A. D'Arcy.**

#	Lecture	date	day
Lecture 1	Introduction	10/10/2018	Wednesday
Lecture 2	Crash Course in Python	11/10/2018	Thursday
Instruction 1	*Python*	*12/10/2018*	*Friday*
Lecture 3	Basic data visualisation/exploration	17/10/2018	Wednesday
Lecture 4	Decision trees	18/10/2018	Thursday
Instruction 2	*Decision trees and data visualization/explora*	*19/10/2018*	*Friday*
Lecture 5	Regression	24/10/2018	Wednesday
Lecture 6	Support vector machines	25/10/2018	Thursday
Instruction 3	*Regression and support vector machines*	*26/10/2018*	*Friday*
Lecture 7	Neural networks (1/2)	31/10/2018	Wednesday
Instruction 4	*Neural networks and supervised learning*	*02/11/2018*	*Friday*
Lecture 8	Neural networks (2/2)	07/11/2018	Wednesday
Lecture 9	Evaluation of supervised learning problems	08/11/2018	Thursday
Instruction 5	*Neural networks and supervised learning*	*09/11/2018*	*Friday*
Lecture 10	Clustering	14/11/2018	Wednesday
Lecture 11	Frequent items sets	15/11/2018	Thursday
Lecture 12	Association rules	21/11/2018	Wednesday
Lecture 13	Sequence mining	22/11/2018	Thursday
Instruction 6	*Clustering, frequent items sets, association ru*	*23/11/2018*	*Friday*
Lecture 14	Process mining (unsupervised)	28/11/2018	Wednesday
Lecture 15	Process mining (supervised)	29/11/2018	Thursday
Instruction 7	*Process mining and sequence mining*	*30/11/2018*	*Friday*
Lecture 16	Text mining (1/2)	05/12/2018	Wednesday
Instruction 8	*Text mining and process mining*	*06/12/2018*	*Thursday !!*
Lecture 17	Text mining (2/2)	12/12/2018	Wednesday
Lecture 18	Data preprocessing, data quality, binning, etc	13/12/2018	Thursday
Instruction 12	*Example exam questions*	*25/01/2018*	*Friday*
backup		30/01/2019	Wednesday
backup		31/01/2019	Thursday
extra	*Question hour*	*01/02/2019*	*Friday*

Lecture 5	Regression	24/10/2018	Wednesday
Lecture 6	Support vector machines	25/10/2018	Thursday
Instruction 3	*Regression and support vector machines*	*26/10/2018*	*Friday*
Lecture 7	Neural networks (1/2)	31/10/2018	Wednesday
Instruction 4	*Neural networks and supervised learning*	*02/11/2018*	*Friday*