

SUPERVISED LEARNING - REGRESSION

JENS BAETENS

LINEAIRE REGRESSION

	¹ feature	^{Label} output
0	15.923194	232.602081
1	4.294681	81.283221
2	18.450278	217.219276
3	1.454430	39.722608
4	15.529496	230.239091
5	0.994415	33.785656
6	17.832737	204.194535
7	9.533831	127.256491
8	11.308549	169.846785
9	1.165202	53.876601

Voorspel het resultaat in de output kolom op basis van de inputs (hier de feature kolom)

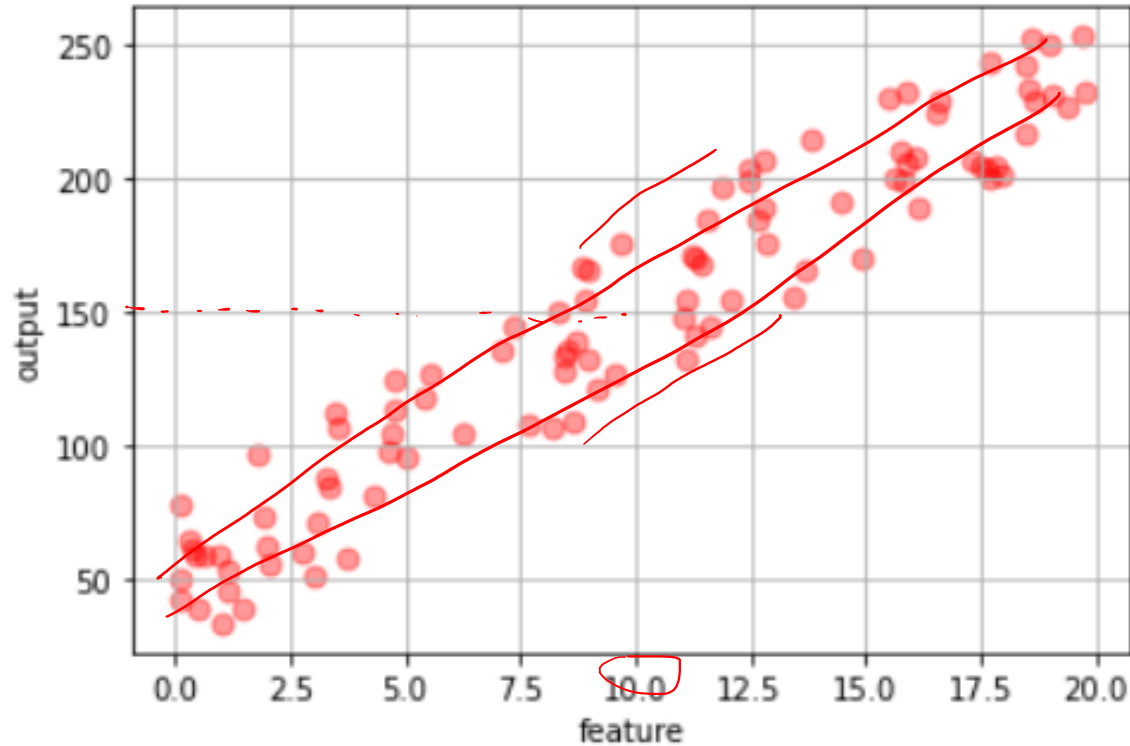
Output wordt ook vaak target genoemd
Trainingsset = 10 training examples

Output is een (continue) variabele

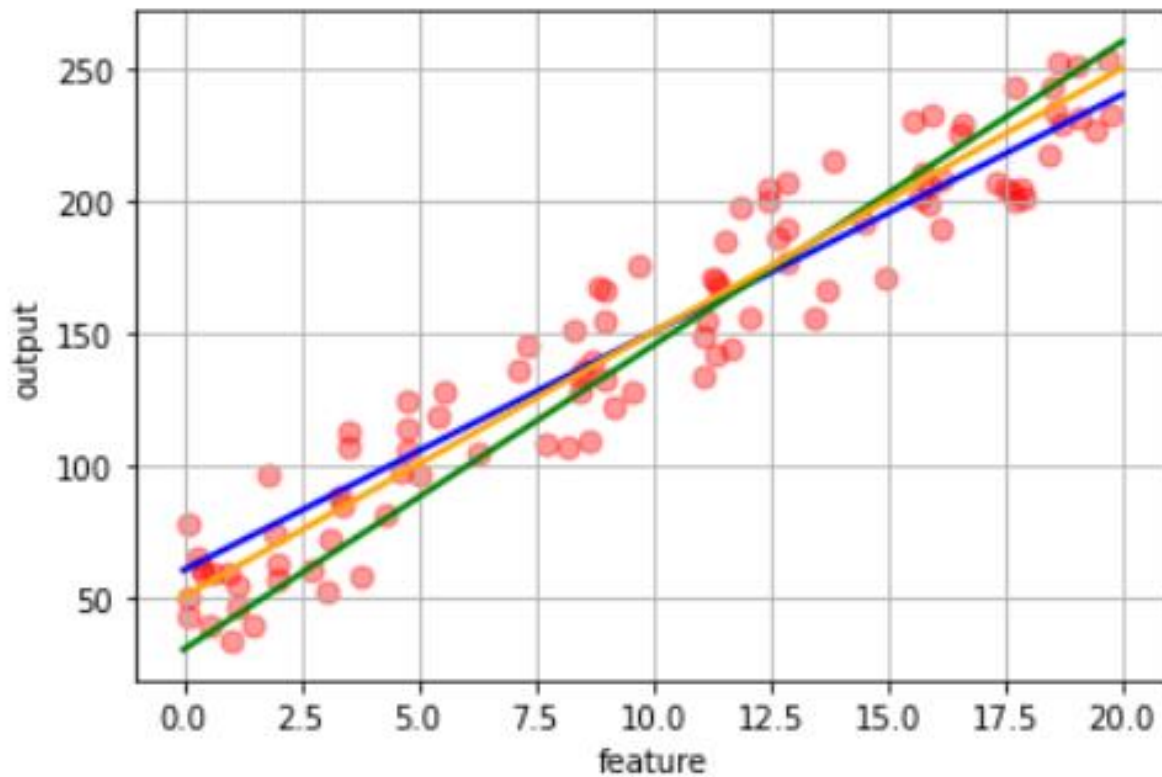
→ Zoek beste rechte

↳ Fout zo klein mogelijk

$$x=10 \rightarrow y=150$$



WAT IS HET BESTE MODEL?



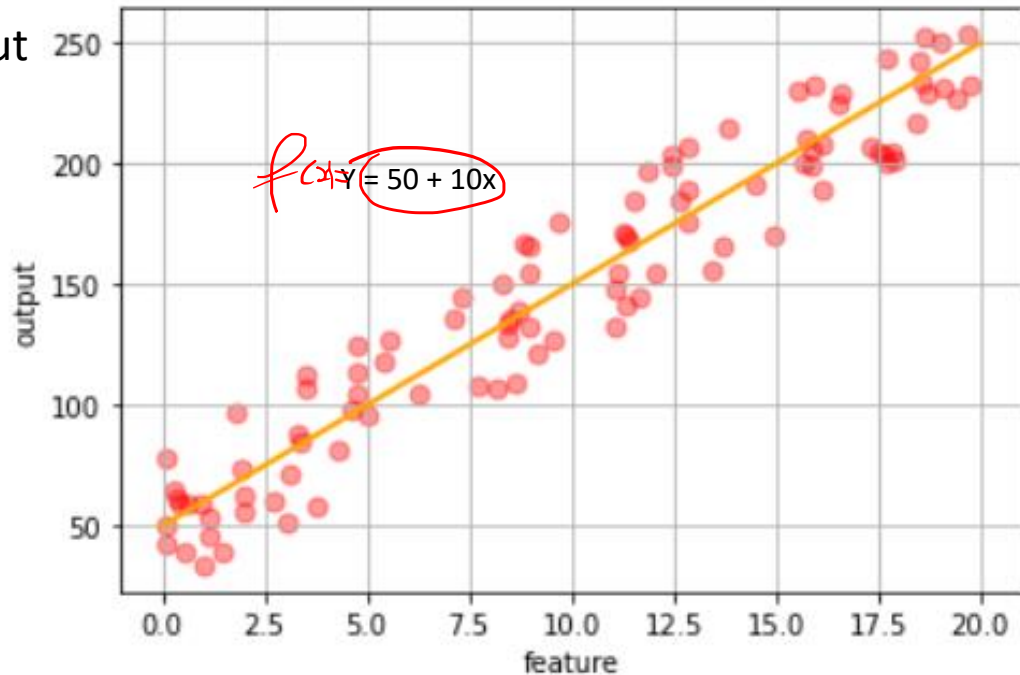
ENKELVOUDIGE LINEAIRE REGRESSIE

↳ 1 feature

Zoek verband feature en output

Lineaire trendlijn $f(x)$

Enkelvoudig of univariate



ENKELVOUDIGE LINEAIRE REGRESSIE

$$\begin{array}{c} w_1 \\ \downarrow \\ \text{---} \end{array}$$

De trendlijn = Het verband tussen twee waarden

$$f_w(x) = w_0 + w_1 x = \text{target}$$

$$y = ax + b$$

\downarrow \downarrow
 w_1 w_0
rico

$$w_0 = 0 \rightarrow y \rightarrow \text{fout}$$
$$= 1 \rightarrow y_2 \rightarrow \text{fout}$$

Regressie zoekt de optimale waarden voor w_0 en w_1

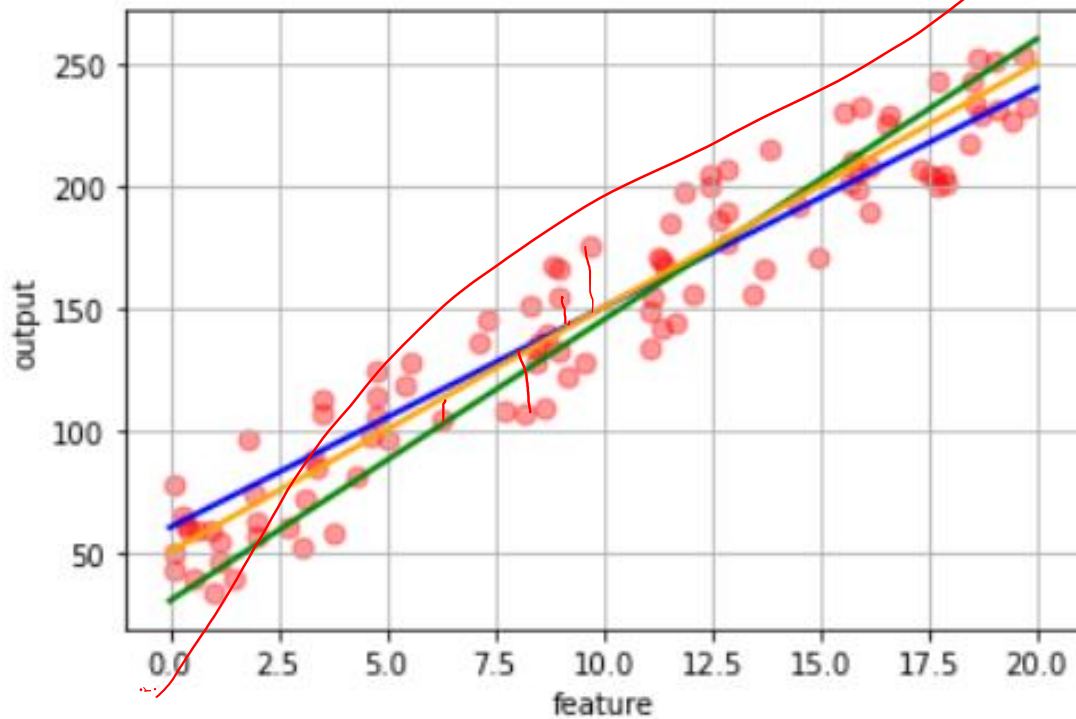
Deze waarden worden **gewichten** genoemd (weights) of de te trainen **parameters**

- Gecombineerd voorgesteld als vector $\mathbf{w} = [w_0, w_1]$

Het zoeken van het trendlijn / model / hypothese = training / learning

\hookrightarrow aanpassen gewichten

WAT IS HET BESTE MODEL?



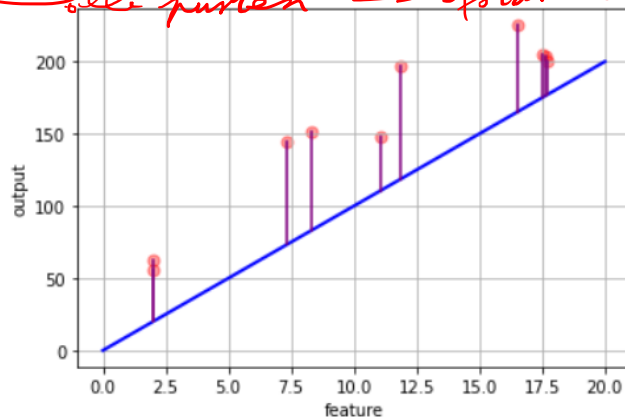
WAT IS HET BESTE MODEL?

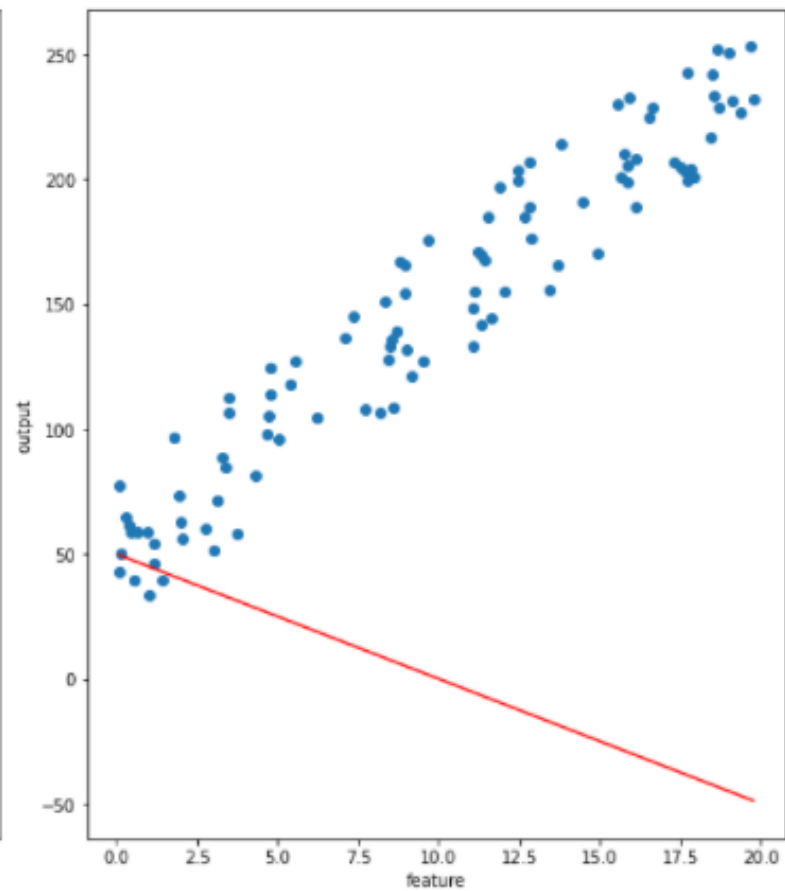
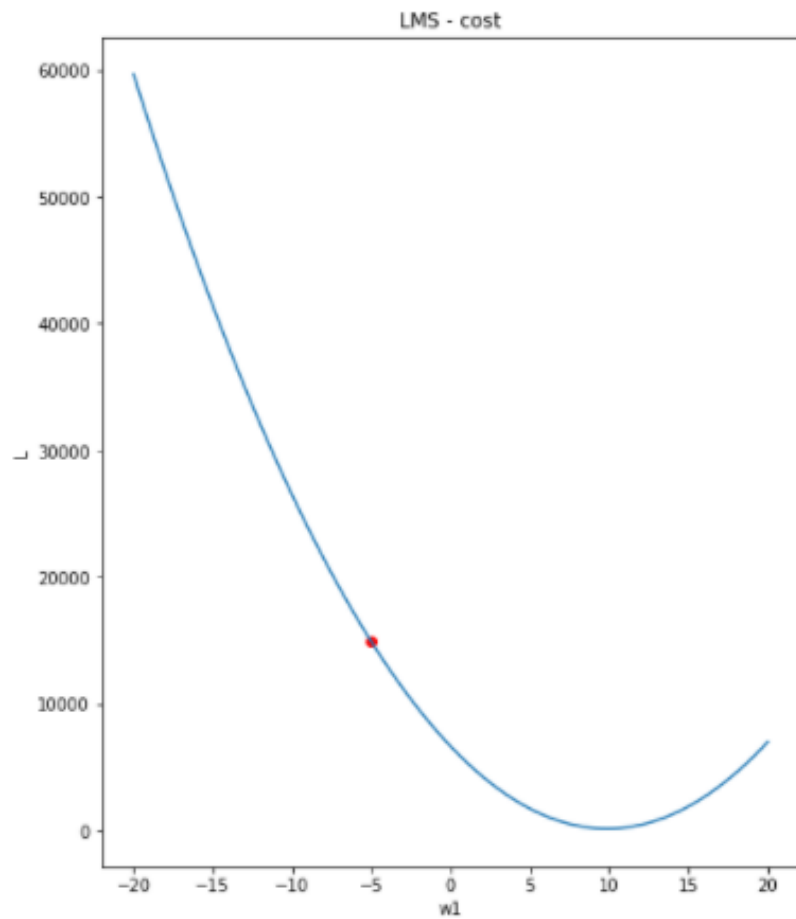
Beste model wordt gekozen door minimalisatie van een kostenfunctie.

Bvb: Least Mean Squares (LMS) voor N examples met input x^i en targets y^i

$$L(\mathbf{w}) = \frac{1}{2N} \sum_{i=1}^N (f_{\mathbf{w}}(x^i) - y^i)^2$$

gemiddelde nemen ↗ *200 punten* ↗ *afstand* → *paarse*





GRADIENT DESCENT

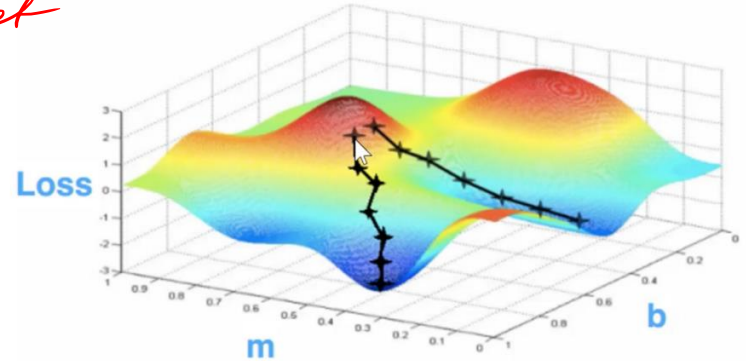


GRADIENT DESCENT – LOKAAL MINIMUM?

LMS-functie is convex

- Hierdoor altijd global minimum

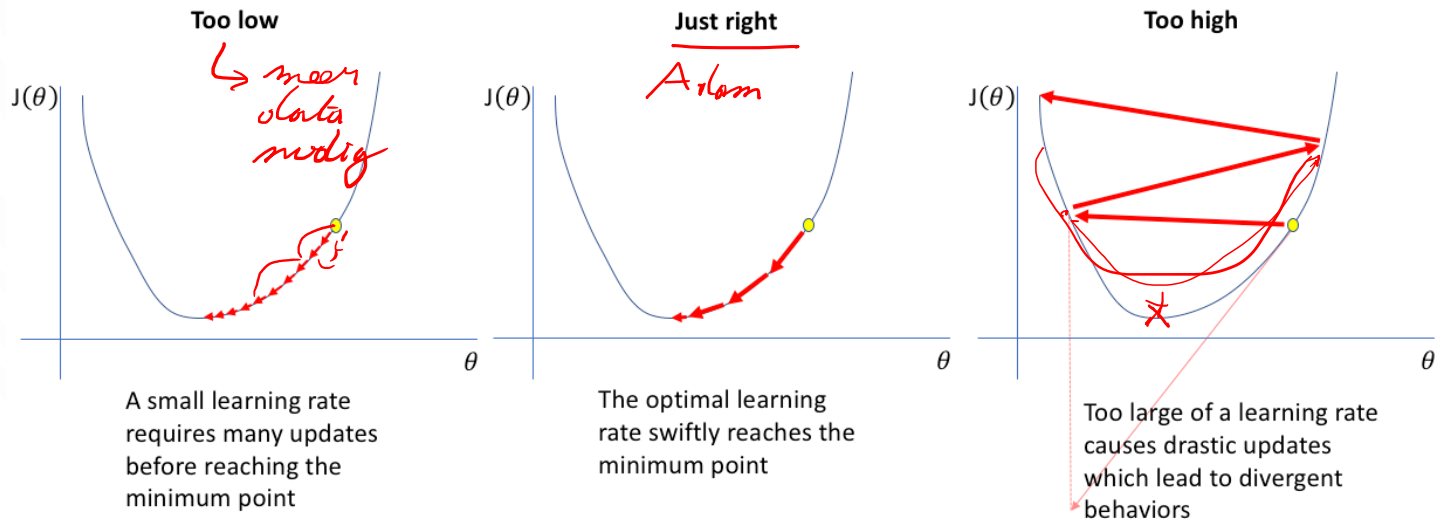
Bij neurale netwerken kan het wel



GRADIENT DESCENT – LEARNING RATE

Bepaalt hoe snel je het optimum benaderd.

“De grootte van de stappen”



TRAINEN VAN HET MODEL

Zelf implementeren of gebruik maken van bestaande frameworks (sklearn)
↳ raad ik niet aan

Construct model => Fit model => Make predictions

MEERDERE FEATURES

In de praktijk zijn er normal meer features beschikbaar.

- Meervoudige of multiple regression

Bovenstaande formules aan te passen met meer gewichten.

Hoeveel extra gewichten per feature nodig?

EVALUEREN VAN HET MODEL

Gemiddelde kwadratische fout $\underline{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \leftrightarrow \underline{CMS}$

Gemiddelde absolute fout $\underline{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$

Determinatiecoëfficiënt $\xrightarrow{\text{Excel}} R^2 = \frac{\sum_{i=1}^N (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^N (y_i - \bar{y})^2}$

FEATURE ENGINEERING - NORMALISATION

Herschaal elke kolom (behalve target) zodat

- Gemiddelde gelijk aan 0
- Standaardafwijking is 1

Andere vormen:

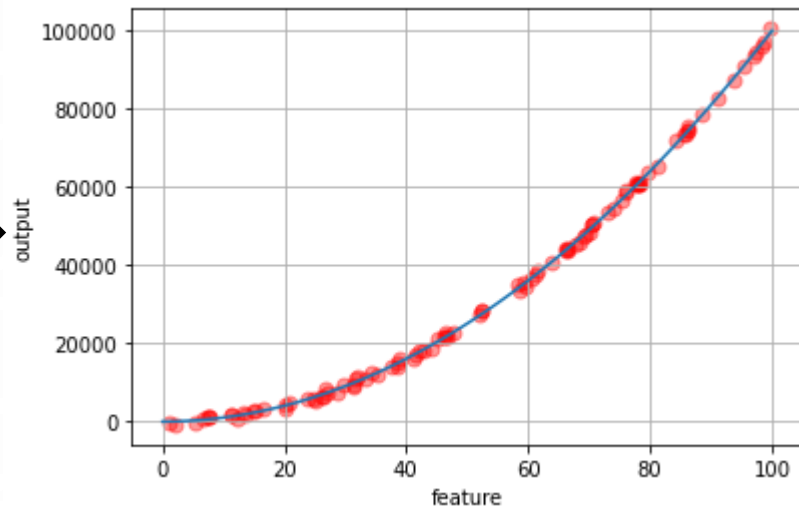
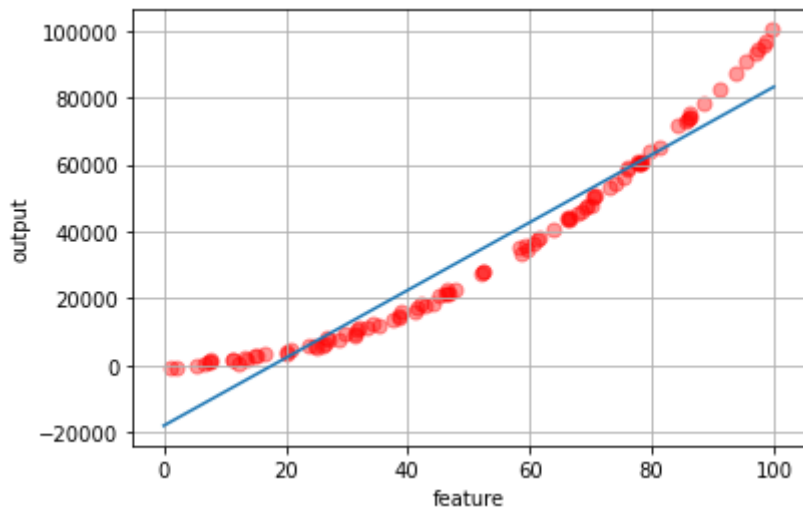
- Delen door het maximum
- Schalen naar het interval 0-1

```
1 scaler = StandardScaler().fit(X_train)
2 X_train = scaler.transform(X_train)
3 X_test = scaler.transform(X_test)
```


FEATURE ENGINEERING – HIGHER ORDER

$$x \rightarrow y \rightarrow w_0 + w_1 x = y?$$

$$x \rightarrow y \rightarrow w_0 + w_1 x + w_2 x^2 = y?$$



Kwadratische features
toevoegen

FEATURE ENGINEERING – EXTRA FEATURES

Bedenken van nieuwe features

- Oppervlakte op basis van breedte β en lengte L $\rightarrow B \cdot L$
- Uit start en eindpunt de afstand halen $\sqrt{(\beta^2 + L^2)}$
- Snelheid bereken op basis van afgelegde afstand en duur van de rit
- Dag van de week of welke maand het is uit de datum halen.

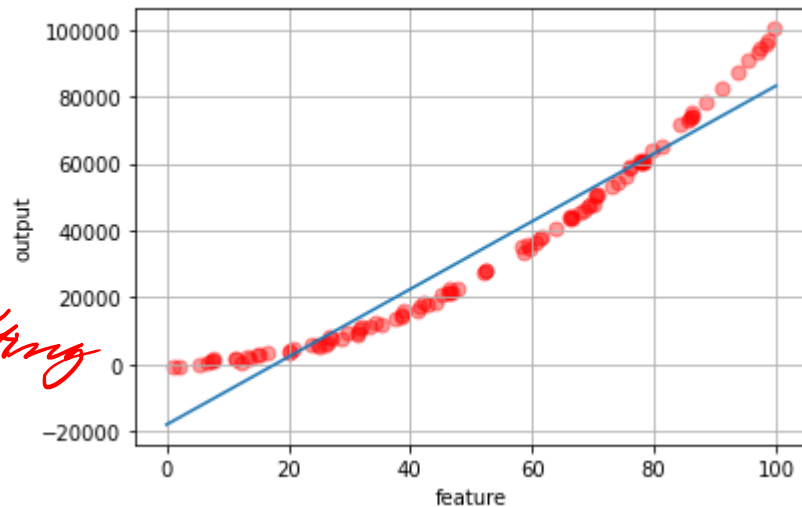
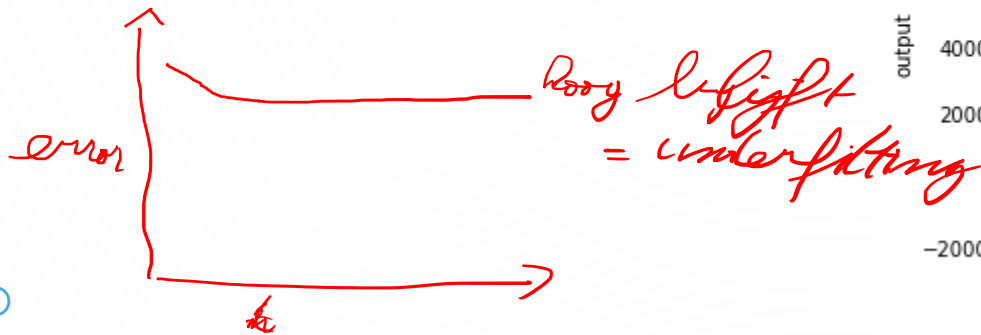
- ... *Polynomial Features*

$$x^3 y^3 \rightarrow x^3 x^2 x y^3 y^2 y xy xy^2 x^2 y$$

UNDERFITTING

= te weinig gewichten

Model is te eenvoudig om de data correct te modelleren

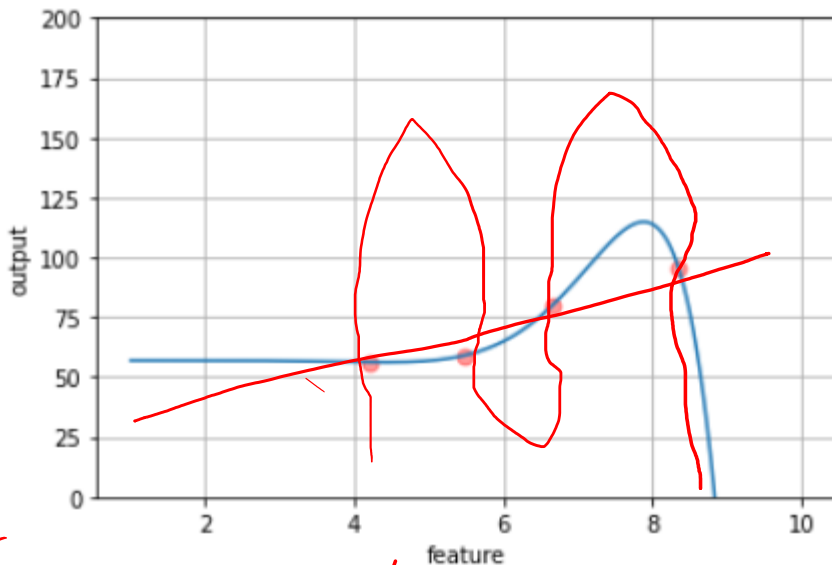
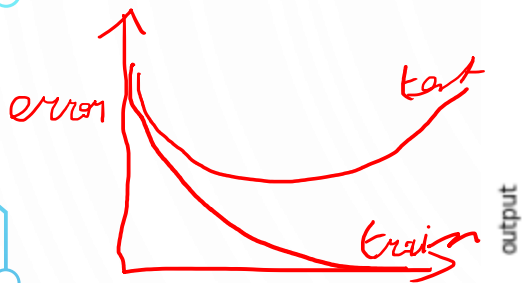


OVERFITTING

→ te gaan
detecteren

training data → model te trainen

→ test data → evalueren
→ overfitting



overfitting

↳ train - error goed
test - error slecht

OVERFITTING - REGULARISATIE

Extra term in de kostenfunctie voor het gebruik van features te penaliseren

$$L(\mathbf{w}) = \frac{1}{2N} \sum_{i=1}^N (f_{\mathbf{w}}(x^i) - y^i)^2 + \lambda R(\mathbf{w})$$

→ norm van alle w 's

De parameter λ is de mate waarin er regularisatie is

- 0 -> geen regularisatie
- ∞ -> alle gewichten zijn nul

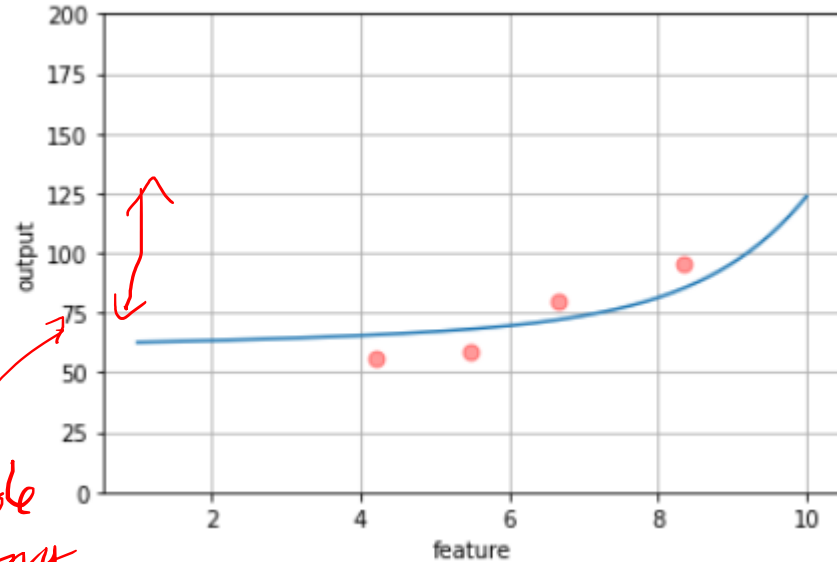
OVERFITTING – L2NORM

$$\text{Regularisatieterm} = \sum_{i=1}^N w_i^2$$

Merk op dat de som begint vanaf 1

De bias wordt niet in rekening gebracht

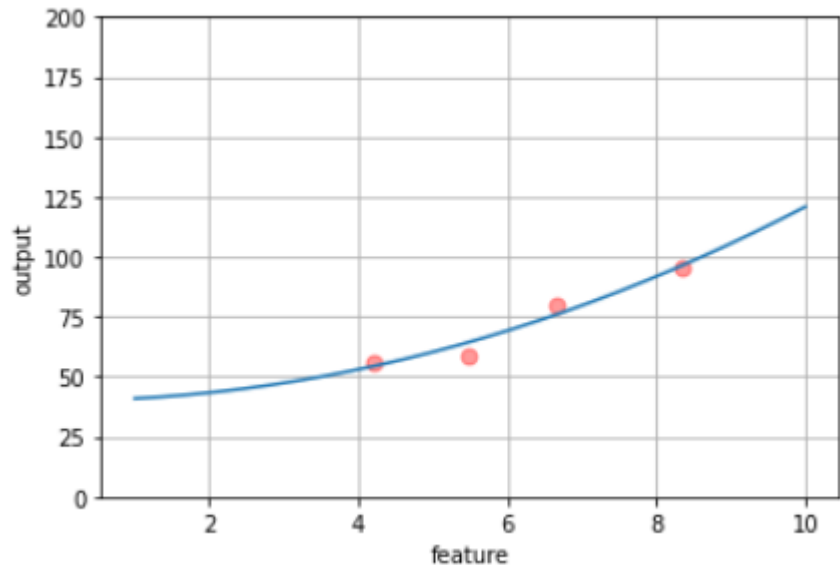
*w₀ niet
→ anders steeds door de
oorsprong*




OVERFITTING – L1NORM

$$\text{Regularisatieterm} = \sum_{i=1}^N |w_i|$$

Voordeel is dat gewichten op nul
gezet kunnen worden


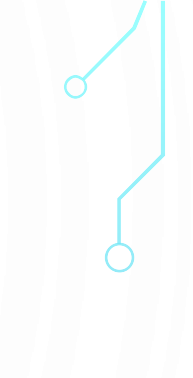


 <https://towardsdatascience.com/l1-and-l2-regularization-explained-874c3b03f668>

↳ *isognito*



GLOSSARY

- Supervised
 - Unsupervised
 - Reinforcement Learning
 - Regression
 - Overfitting
 - Underfitting
 - Learning Rate
 - Loss Function
 - Feature Engineering
 - Normalisation
 - Regularisation
 - Trainen van een model
- 
- 
- 