SUPERVISED LEARNING - REGRESSION

JENS BAETENS

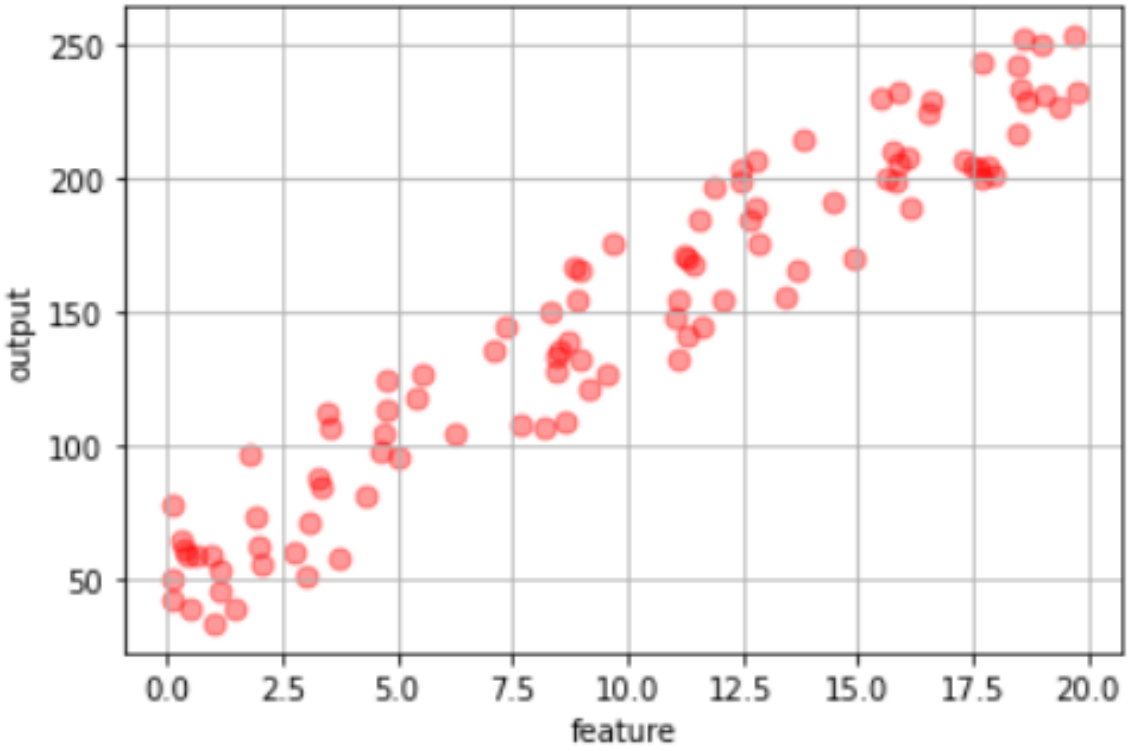
LINEAIRE REGRESSION

	feature	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

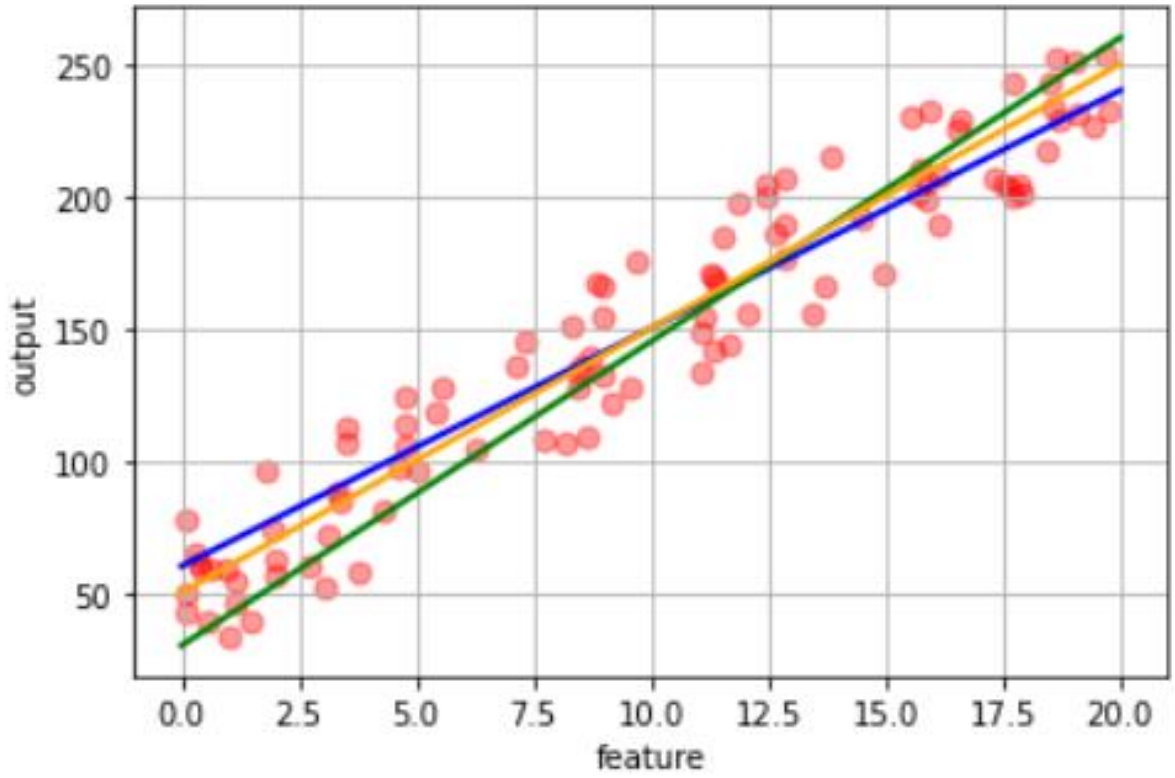
Voorspel het resultaat in de output kolom op basis van de inputs (hier de feature kolom)

Output wordt ook vaak target genoemd
Trainingsset = 10 training examples

Output is een (continue) variabele


250
200
150
100
50
output
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
feature

WAT IS HET BESTE MODEL?

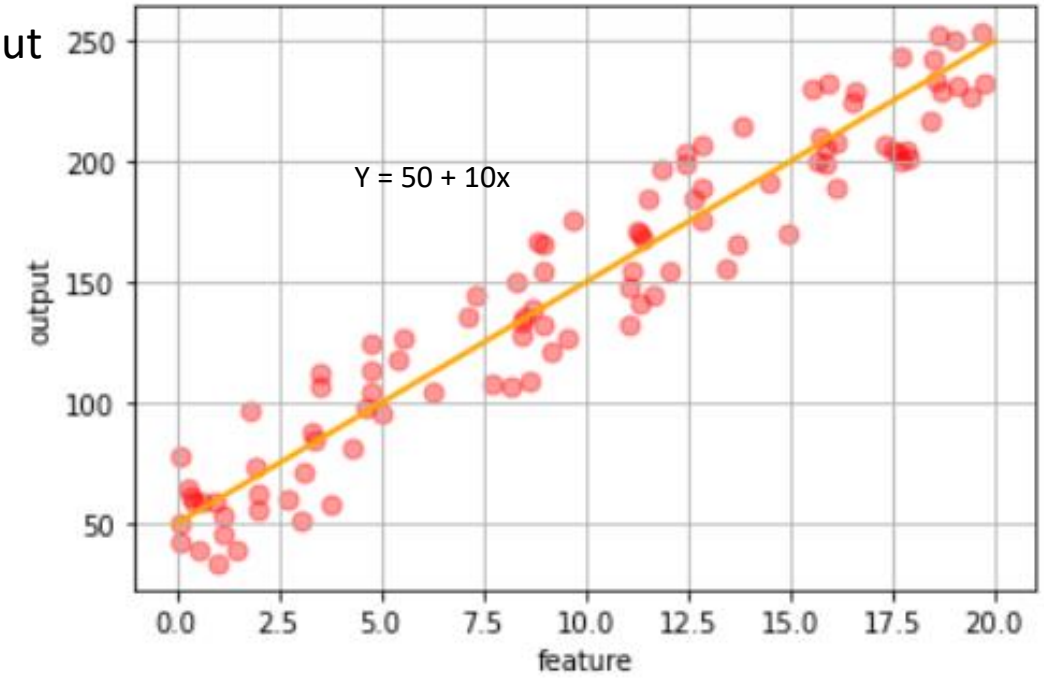


ENKELVOUDIGE LINEAIRE REGRESSIE

Zoek verband feature en output

Lineaire trendlijn f(x)

Enkelvoudig of univariate



ENKELVOUDIGE LINEAIRE REGRESSIE

De trendlijn = Het verband tussen twee waarden

$$f_{\mathbf{w}}(x) = w_0 + w_1 x = \textbf{target}$$

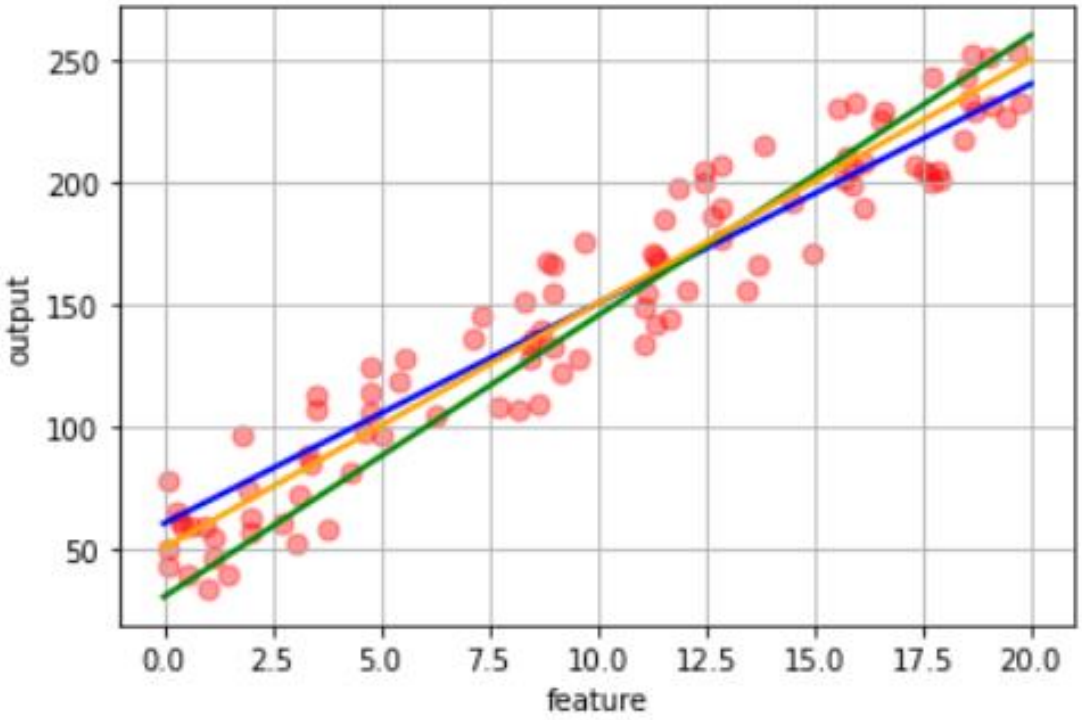
Regressie zoekt de optimale waarden voor w_0 en w_1

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**

WAT IS HET BESTE MODEL?

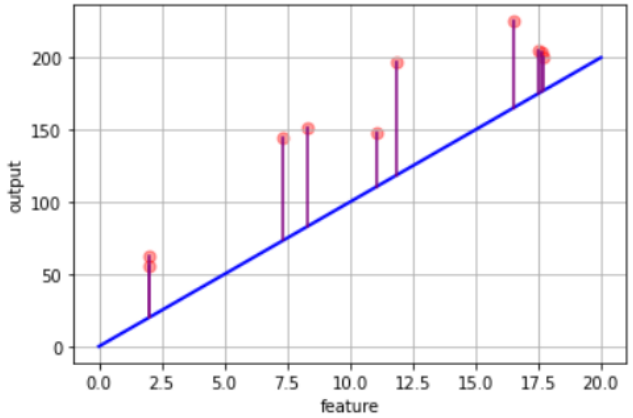


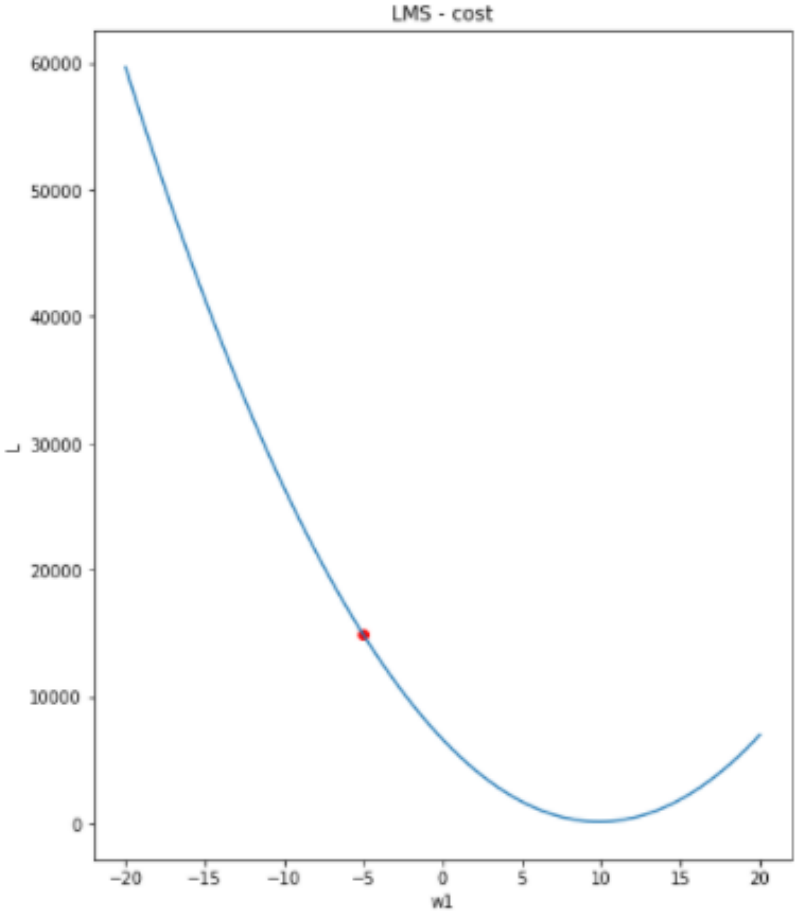
WAT IS HET BESTE MODEL?

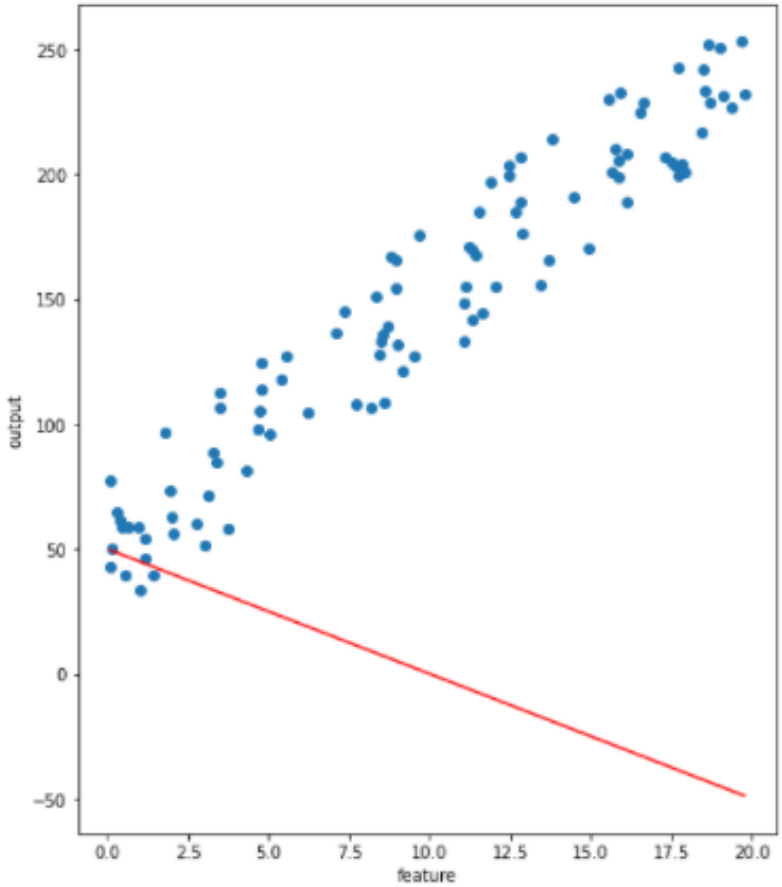
Beste model wordt gekozen door minimalisatie van een kostenfunctie.

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

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




LMS - cost
60000
50000
40000
30000
20000
10000
0
L
−20
−15
−10
−5
0
5
10
15
20
w1


250
200
150
100
50
0
−50
output
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
feature

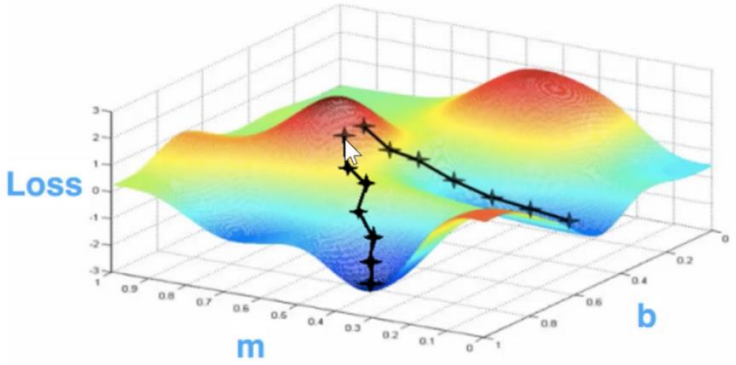
GRADIENT DESCENT

GRADIENT DESCENT – LOKAAL MINIMUM?

LMS–functie is convex

- Hierdoor altijd global minimum

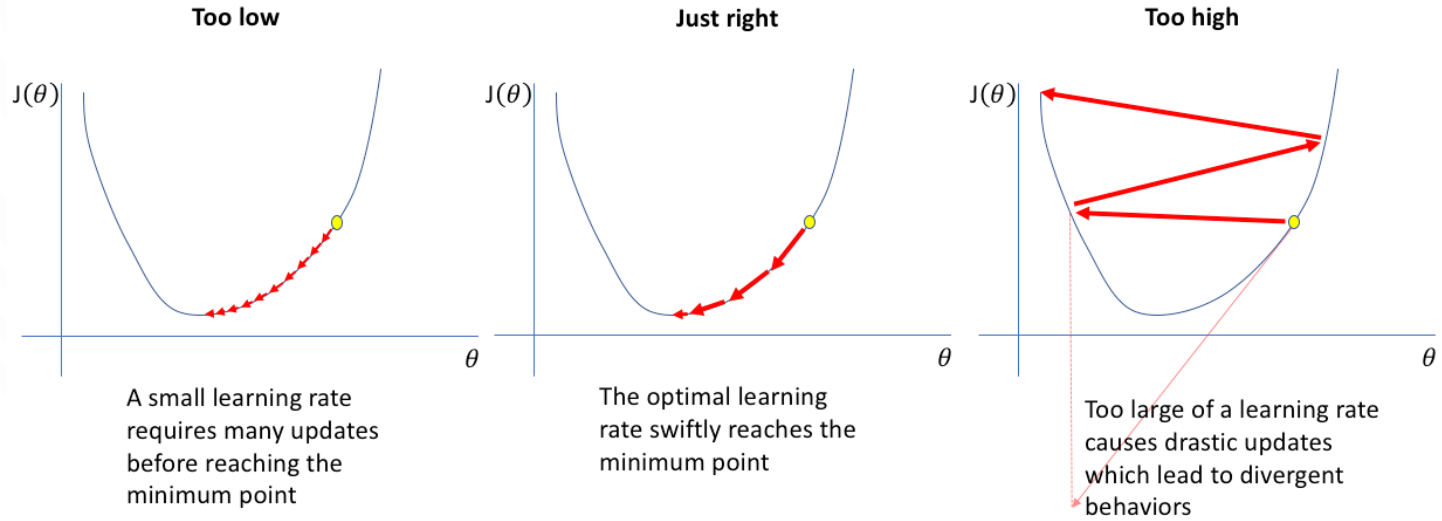
Bij neurale netwerken kan het wel



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)

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

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

Gemiddelde absolute fout

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

Determinatiecoëfficiënt

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

FEATURE ENGINEERING - NORMALISATION

Herschaal elke kolom (behalve target) zodat

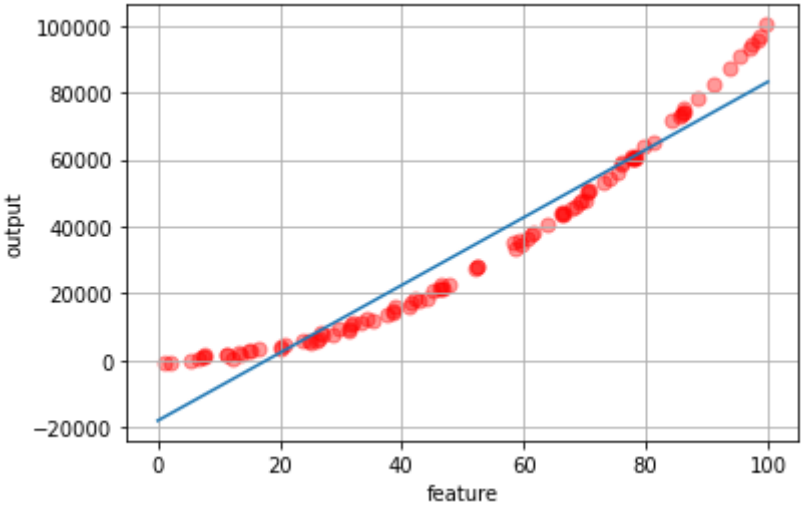
- Gemiddelde gelijk aan 0
- Standaardafwijking is 1

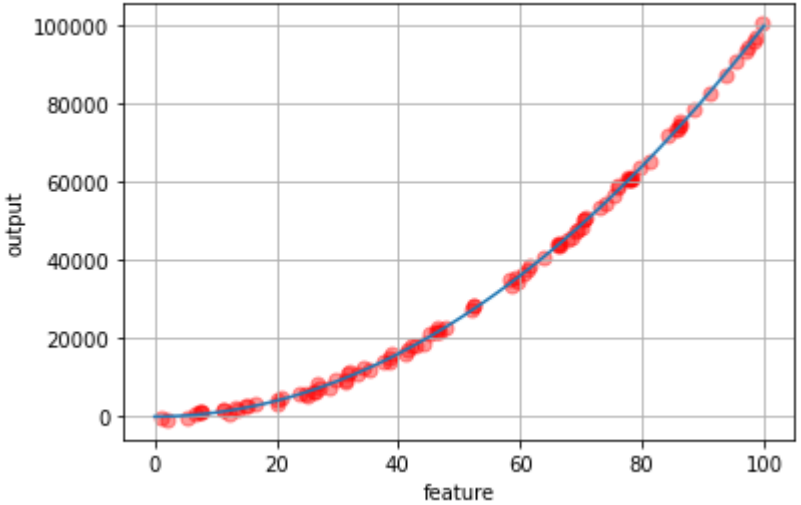
Andere vormen:

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

```
scaler = StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

FEATURE ENGINEERING – HIGHER ORDER





Kwadratische features toevoegen

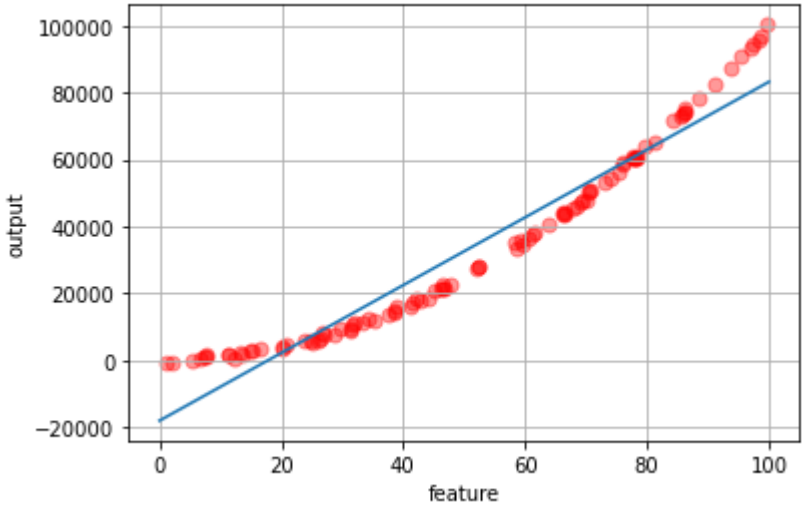
FEATURE ENGINEERING – EXTRA FEATURES

Bedenken van nieuwe features

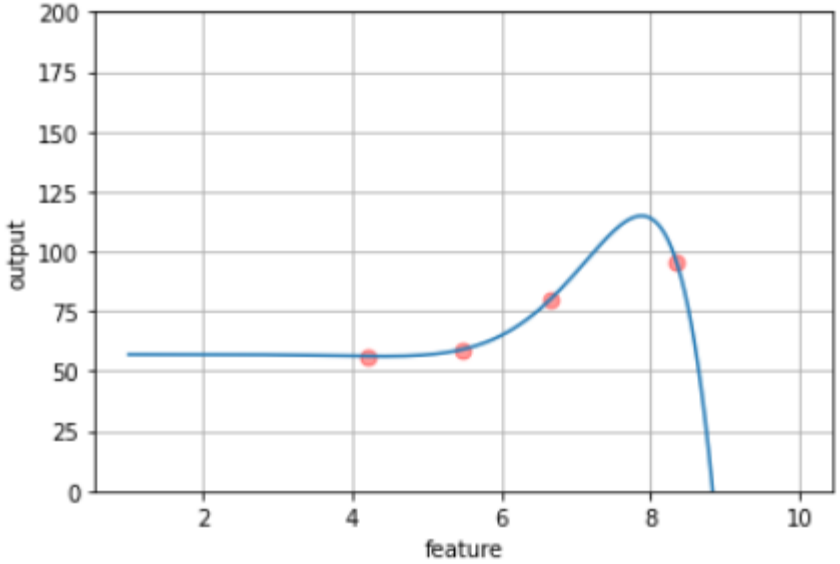
- Oppervlakte op basis van breedte en lengte
- Uit start en eindpunt de afstand halen
- Snelheid bereken op basis van afgelegde afstand en duur van de rit
- Dag van de week of welke maand het is uit de datum halen.
- …

UNDERFITTING

Model is te eenvoudig om de data correct te modelleren



OVERFITTING



OVERFITTING - REGULARISATIE

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

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

De parameter λ is de mate waarin er regularisatie is

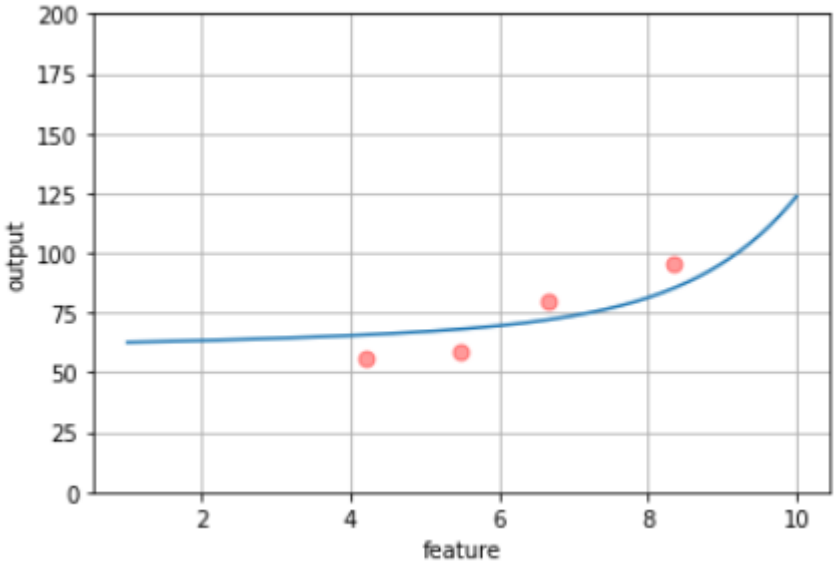
- 0 -> geen regularisatie
- ∞-> alle gewichten zijn nul

OVERFITTING – L2NORM

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

Merk op dat de som begint vanaf 1

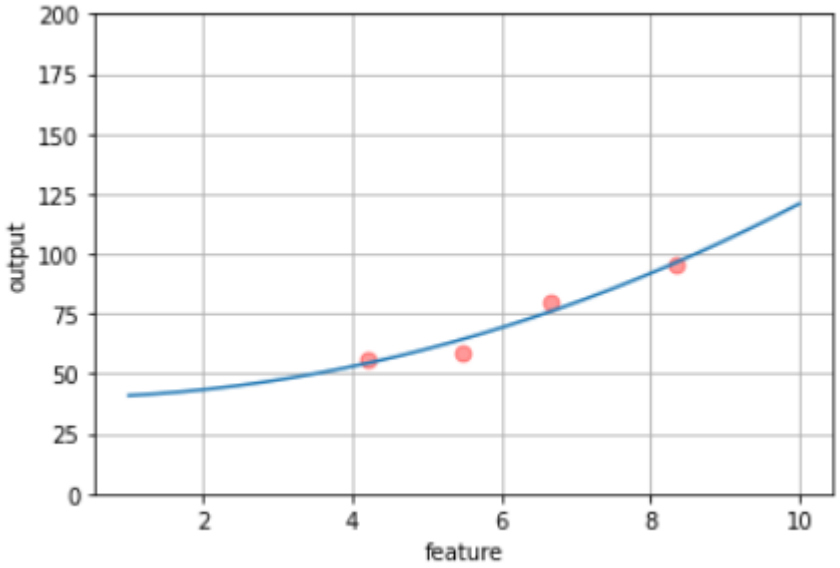
De bias wordt niet in rekening gebracht



OVERFITTING – L1NORM

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

GLOSSARY

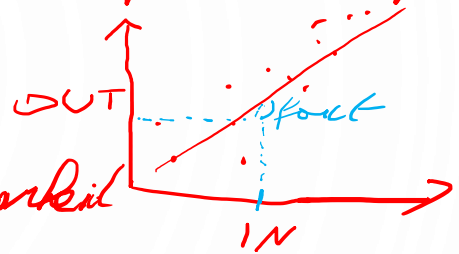
- Supervised
- Unsupervised
- Reinforcement Learning
- Regression
- Overfitting
- Underfitting
- Learning Rate
- Loss Function
- Feature Engineering
- Normalisation
- Regularisation
- Trainen van een model

Regressie → Verband zoeken tussen inputs & outputs

→ Doel: Nieuwe input

↓

Voorspelling ≈ Waarheid



Verband = Model opbouwen

(metadata)

Hyperparameters
- ML techniek
- parameters v.d. techniek
  - learning rate

(data)

parameters
- gewichten
  - $f(x) = w_0 + w_1 x$
- aantal → complexiteit model

- Lin Regr: # para = # features + 1

1) Kiezen hyperparameters / structuur algoritme

2) Model trainen : fitten fit()

↳ Parameters aanpassen - trial & error

↳ Om een kost te minimaliseren

→ MSE : gem - afstanden - fout

- kwadratisch → vermijden ✓

↳ gradient descent

3) Model evalueren : hoe accuraat is het?

- Trainingsdata

→ MSE : Hoog → Slecht model → underfitting

→ kan het verband niet weergeven

: laag : niet noodzakelijk goed

- Testdata → deel v.d data apart
  → Niet gebruikt in de fit
  → .train_test_split()
  ↳ MSE = laag → goed model

- hoog (blauwe lijn vorige slide)
  → overfitting
  → je leert te veel over trainingsdata
  → oplossingen
    - meer data
    - regularisatie (bepaalde parameters geen rol laten spelen)