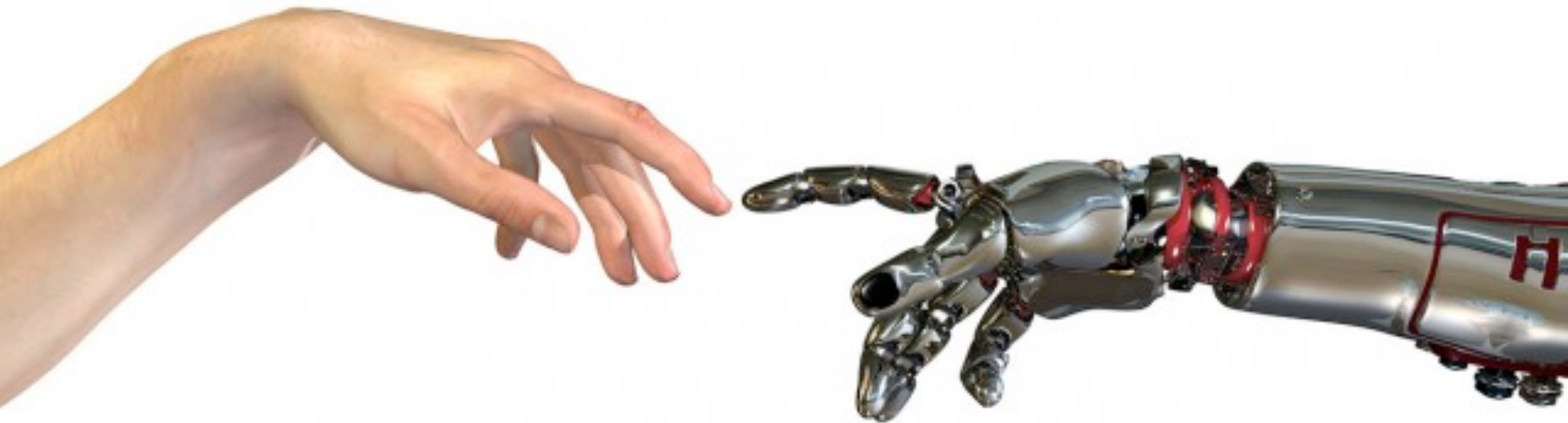
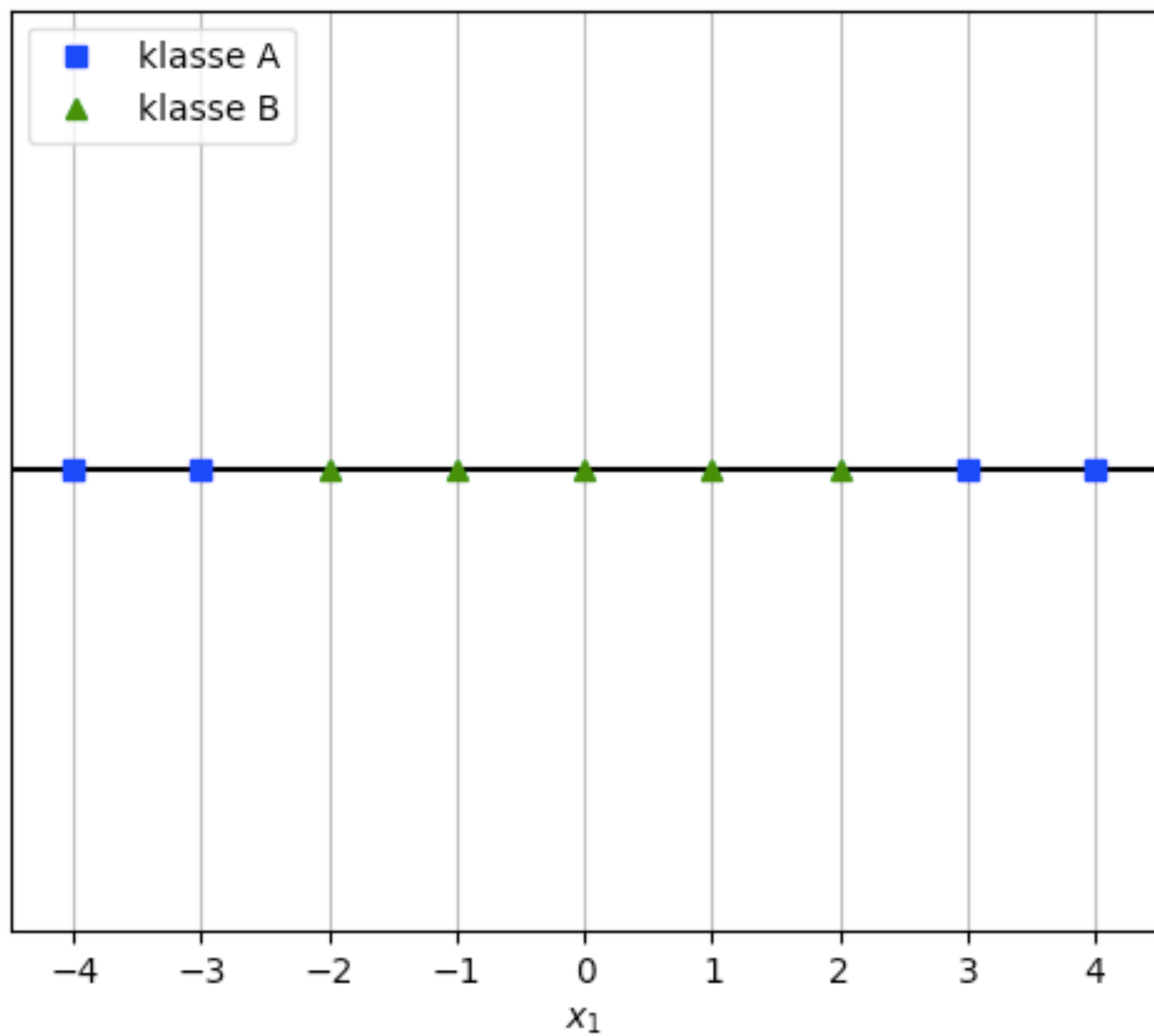


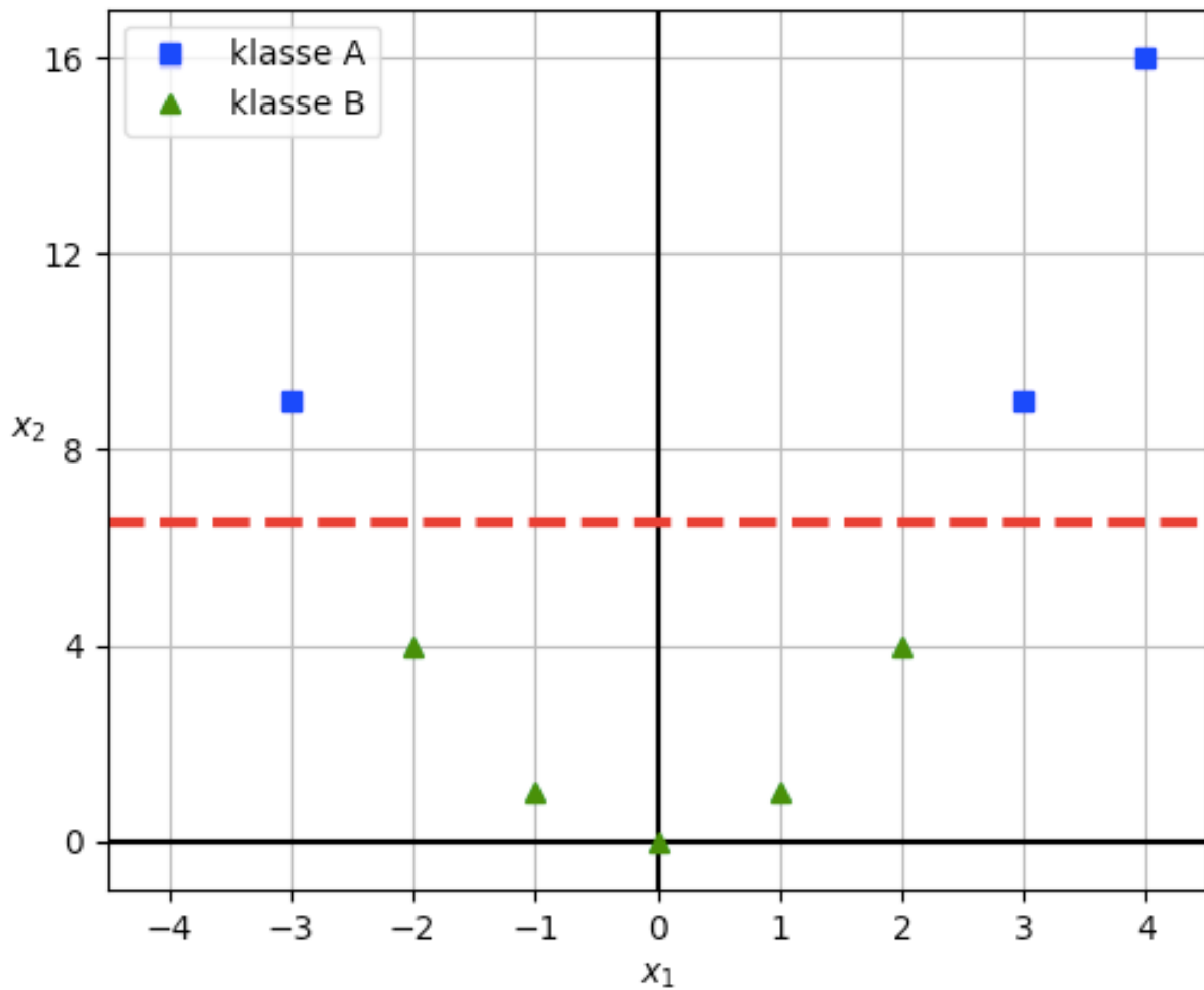
Machine Learning

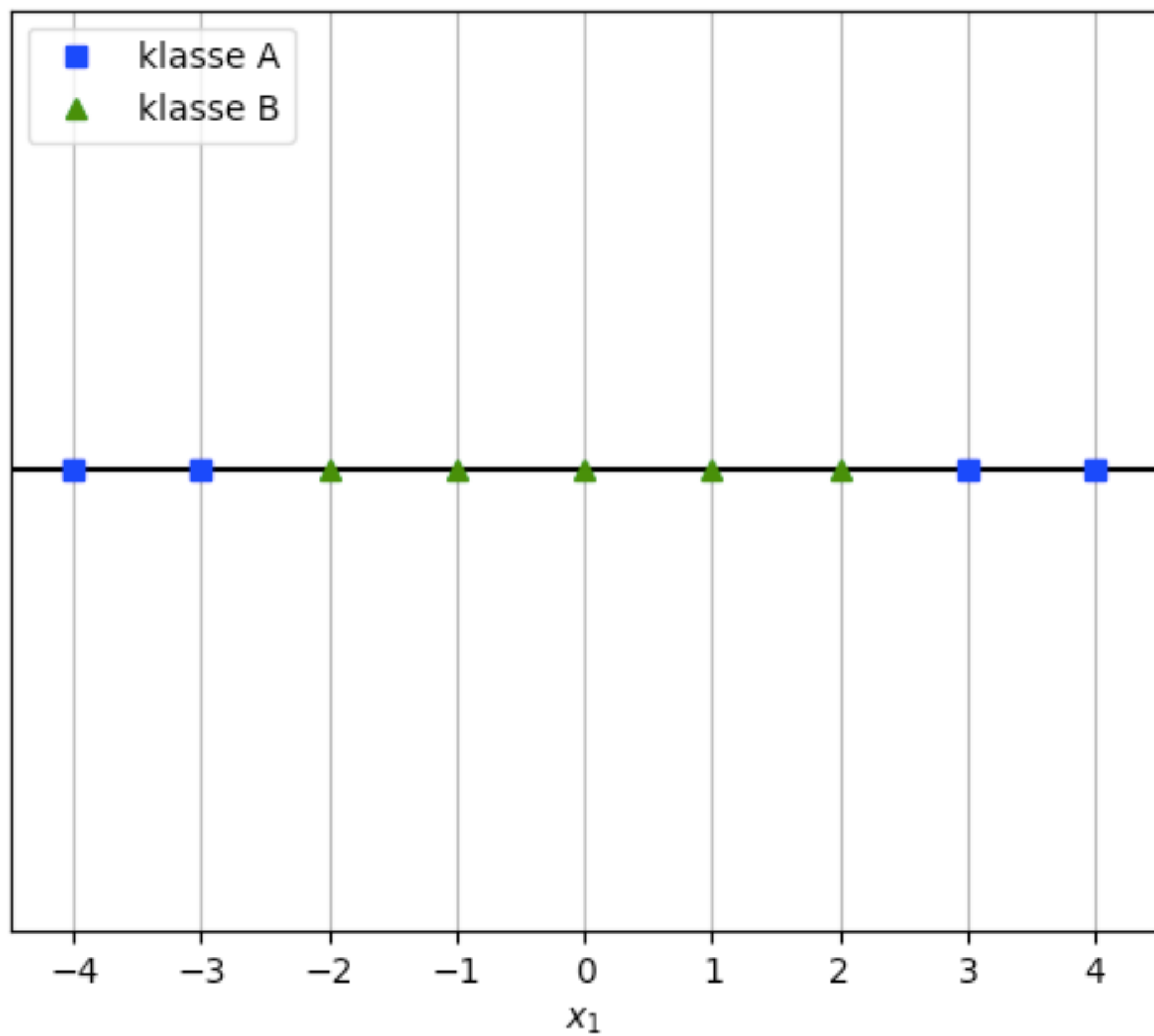
5. modevaluatie



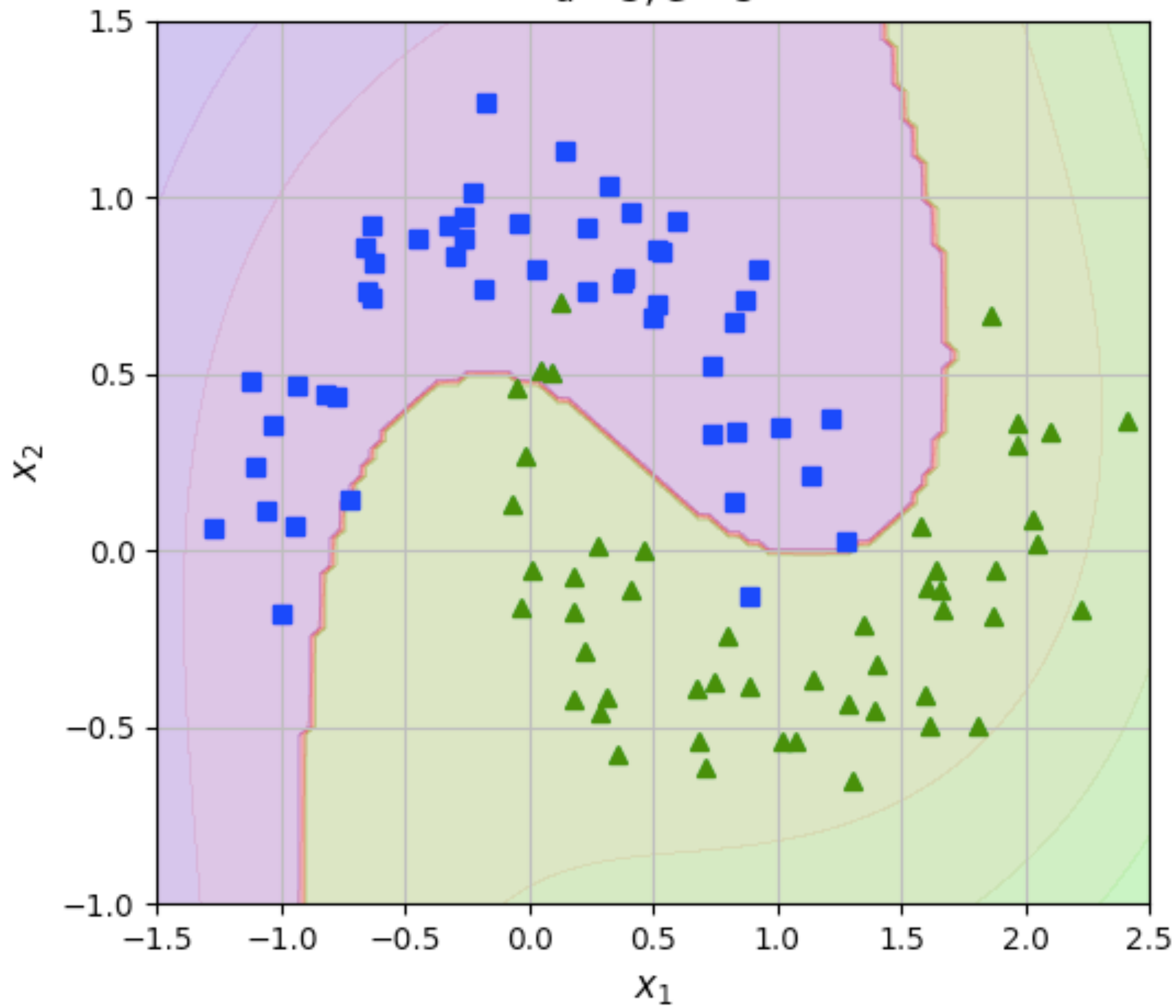
`ml: polynomial`



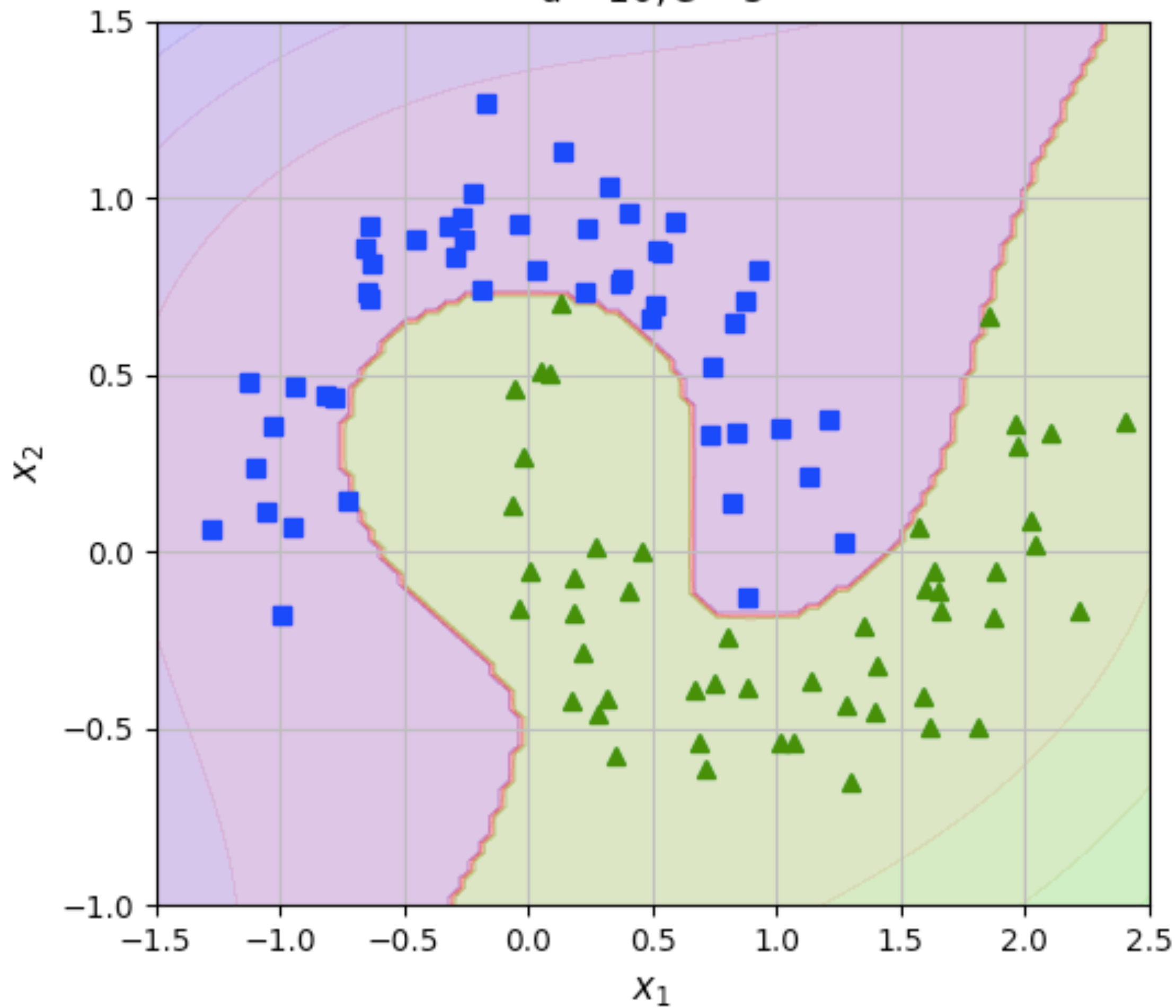




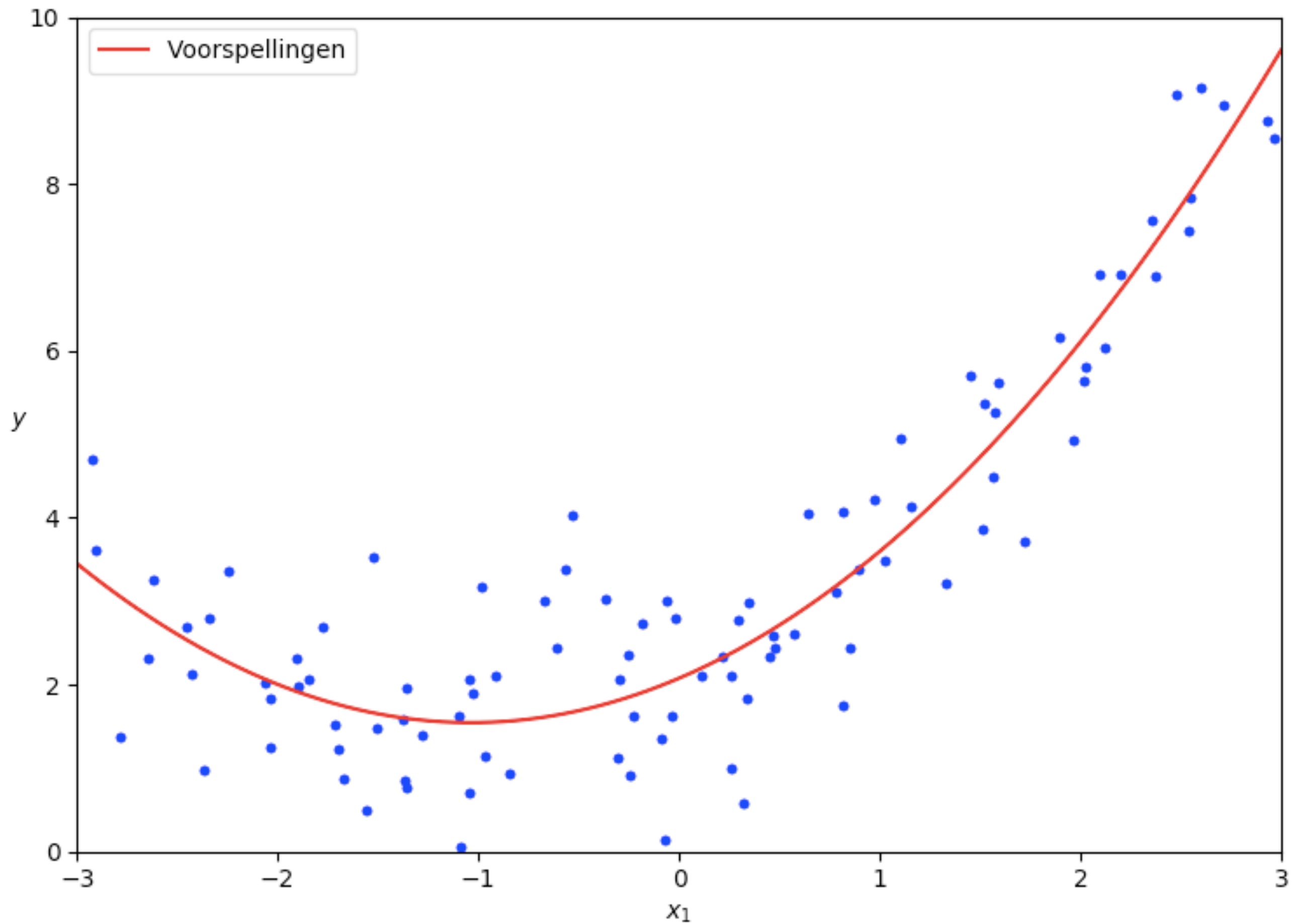
$d=3, C=5$

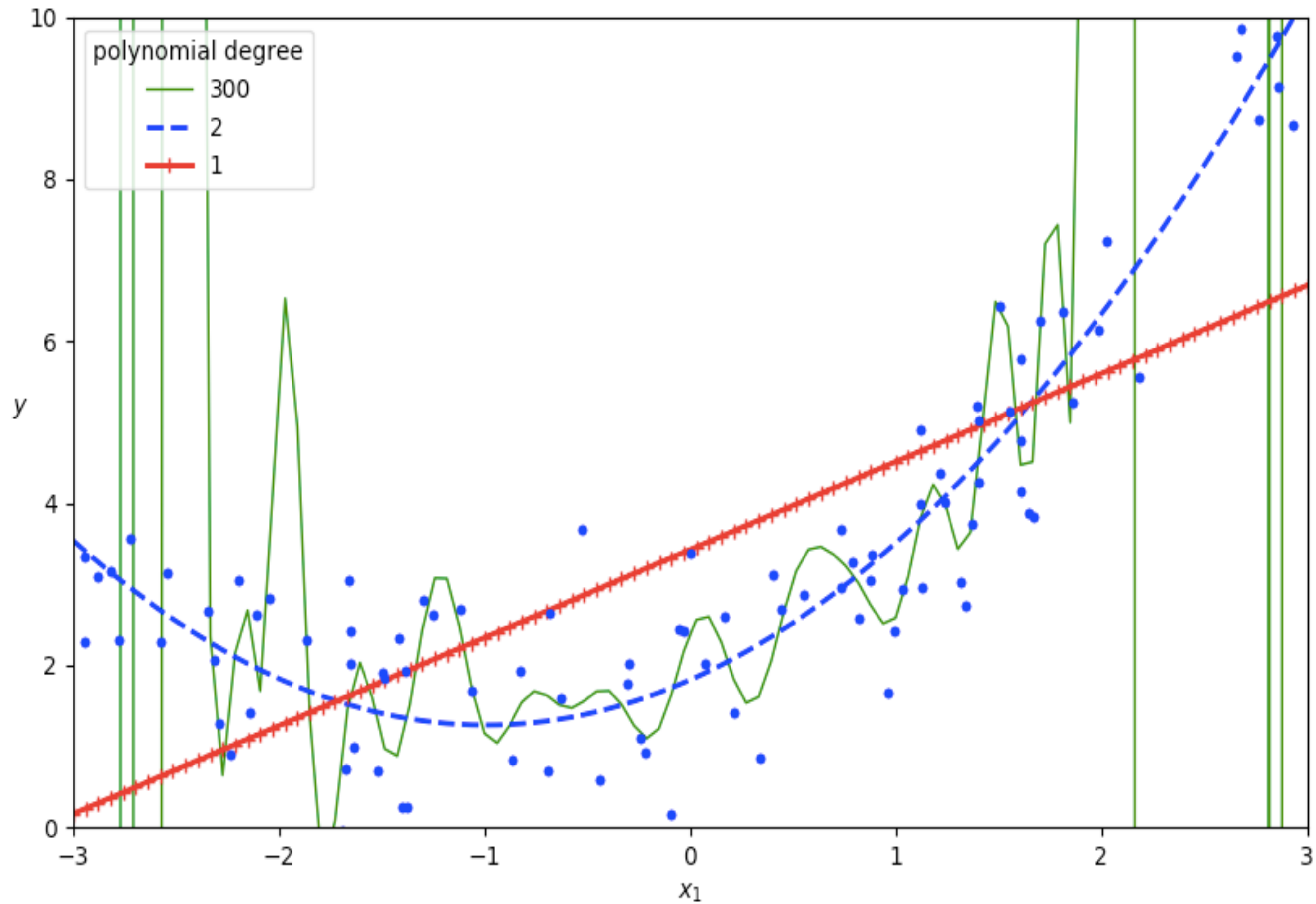


$d = 10, C = 5$



ml: overfitting and
underfitting





Voorkomen van overfitting

Verminderen van het aantal *features*

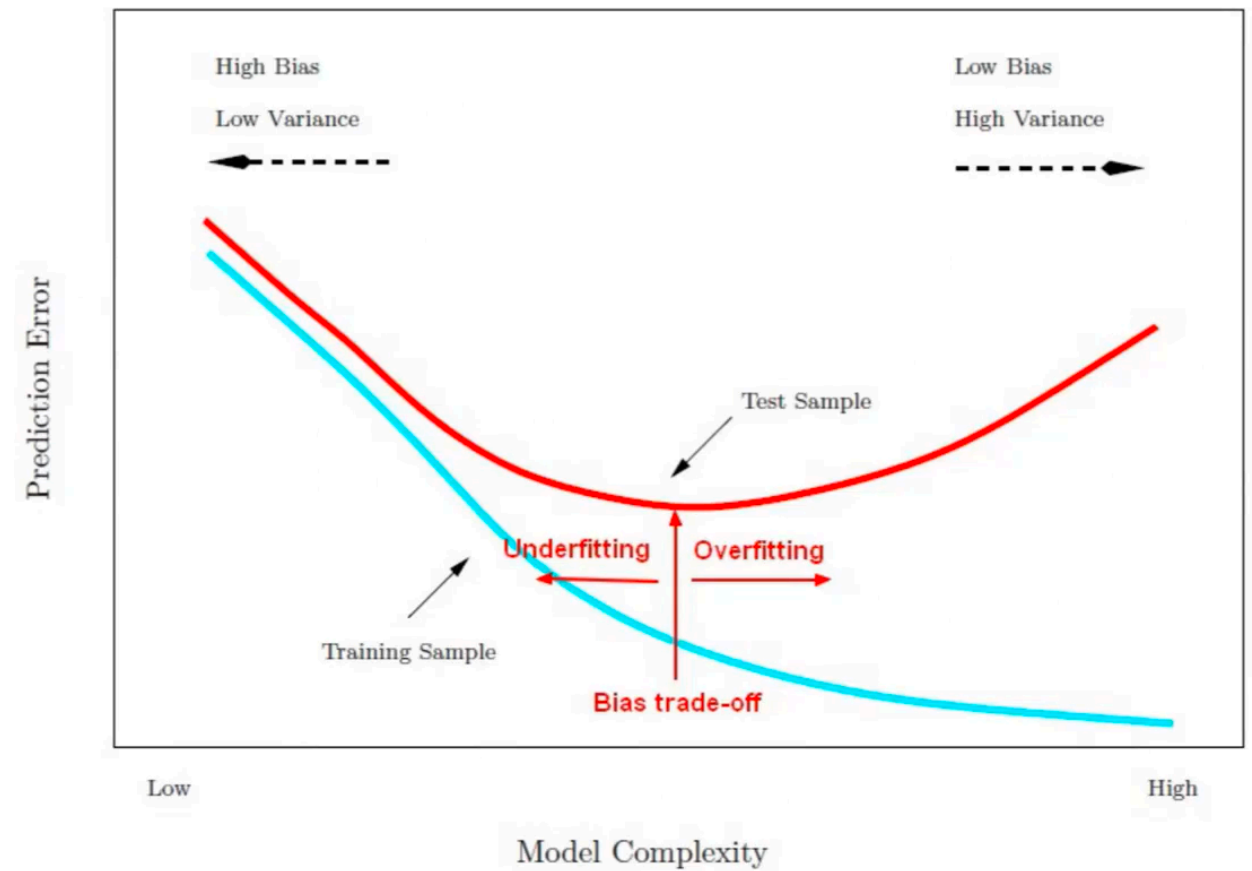
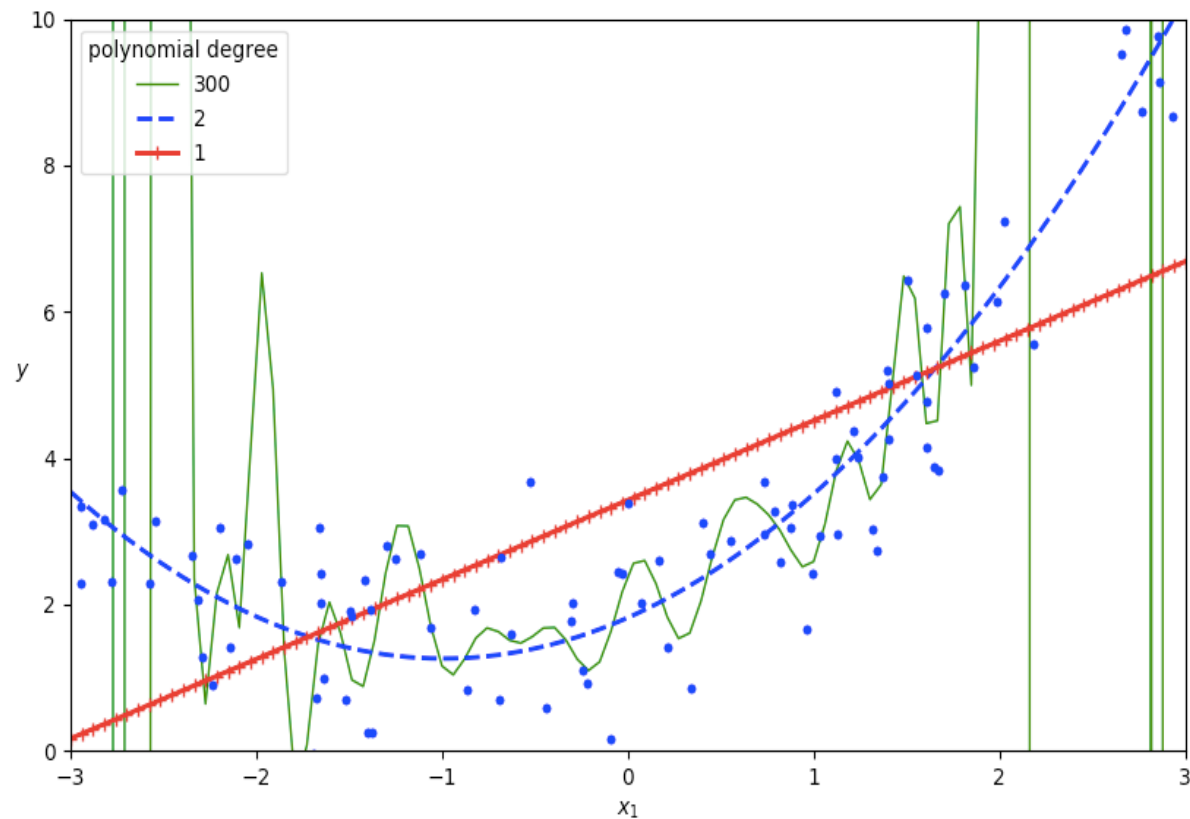
Welke *features* wel en niet te gebruiken?

Welk algoritme past het best?

Regularisatie

Verminderen van de grootte en complexiteit van θ

bias variance tradeoff

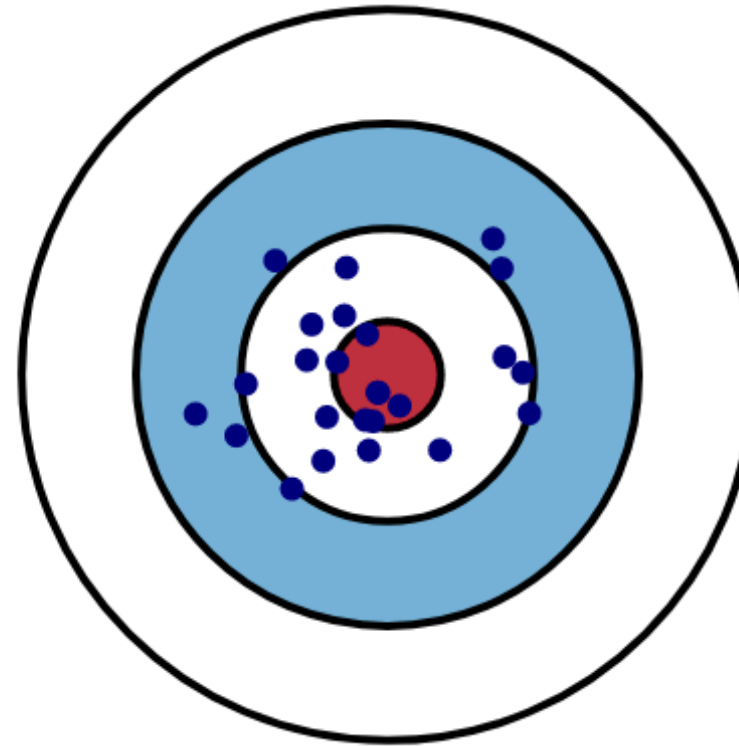
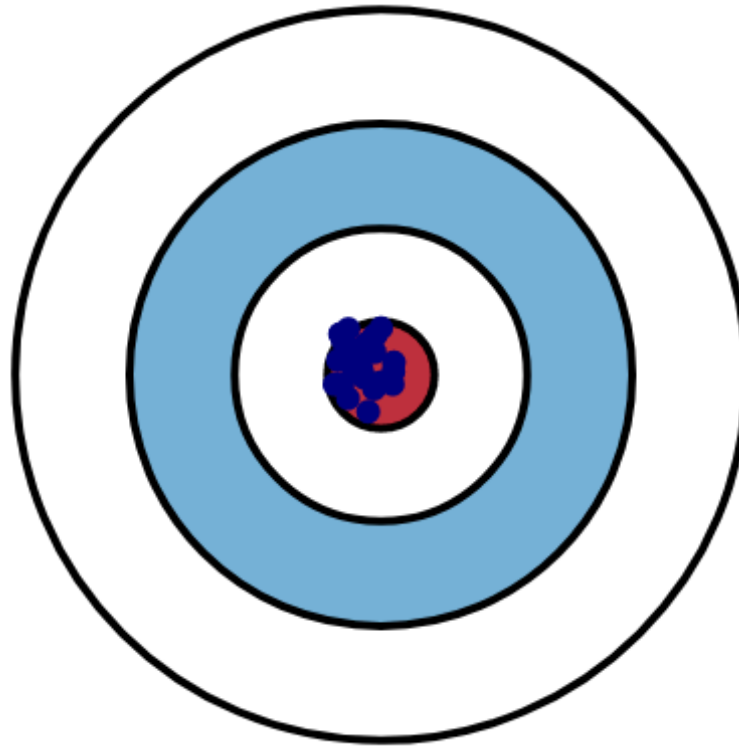


Bron: <https://datalab.com.ng/wp-content/uploads/2020/05/Bias-Variance-Tradeoff.png>

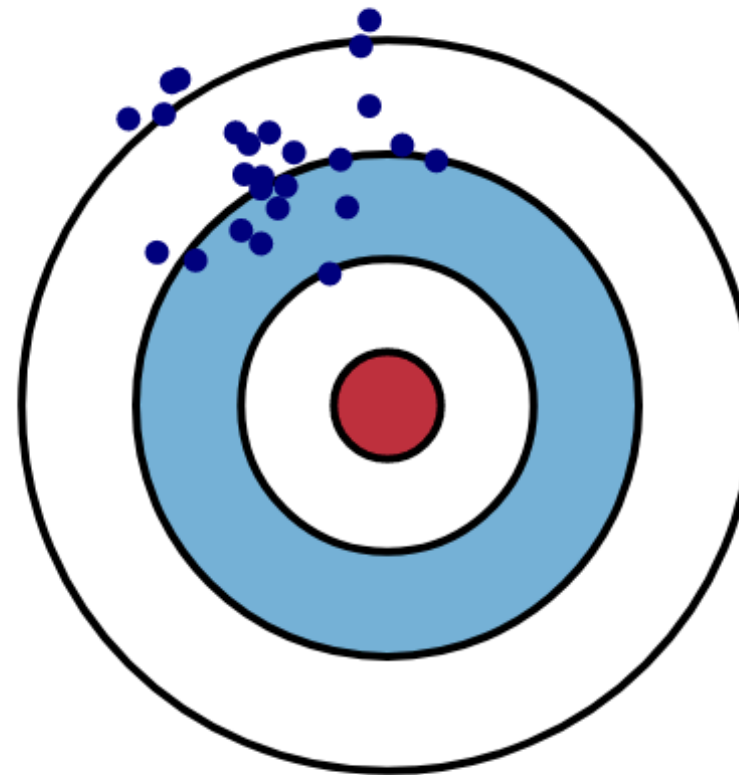
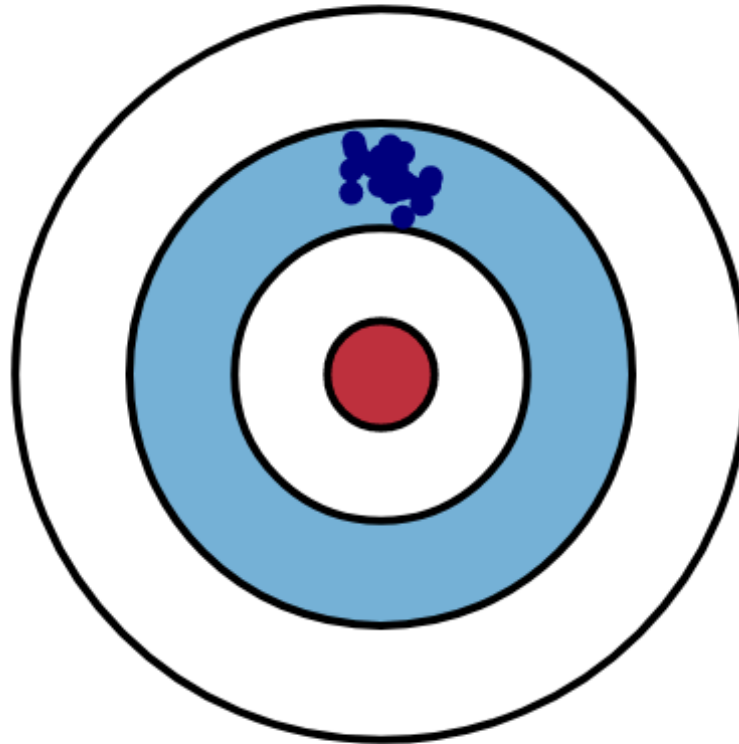
Low Variance

High Variance

Low Bias



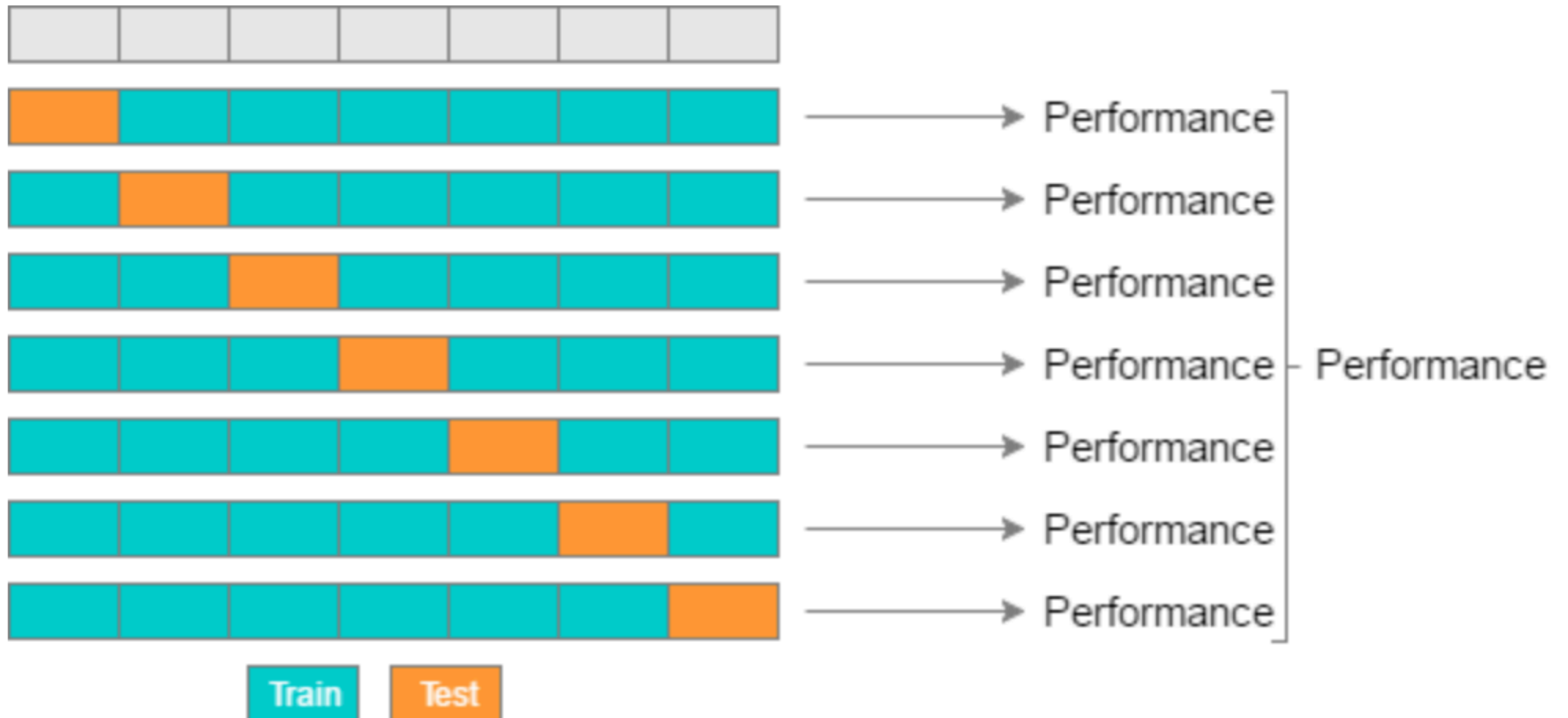
High Bias



ml:trainings-set,
cv-set en test-set

$$X = \begin{bmatrix} \text{---} & (x^{(1)})^T & \text{---} \\ \text{---} & (x^{(2)})^T & \text{---} \\ \text{---} & (x^{(3)})^T & \text{---} \\ & \vdots & \\ \text{---} & (x^{(m)})^T & \text{---} \end{bmatrix}$$

Cross-validation



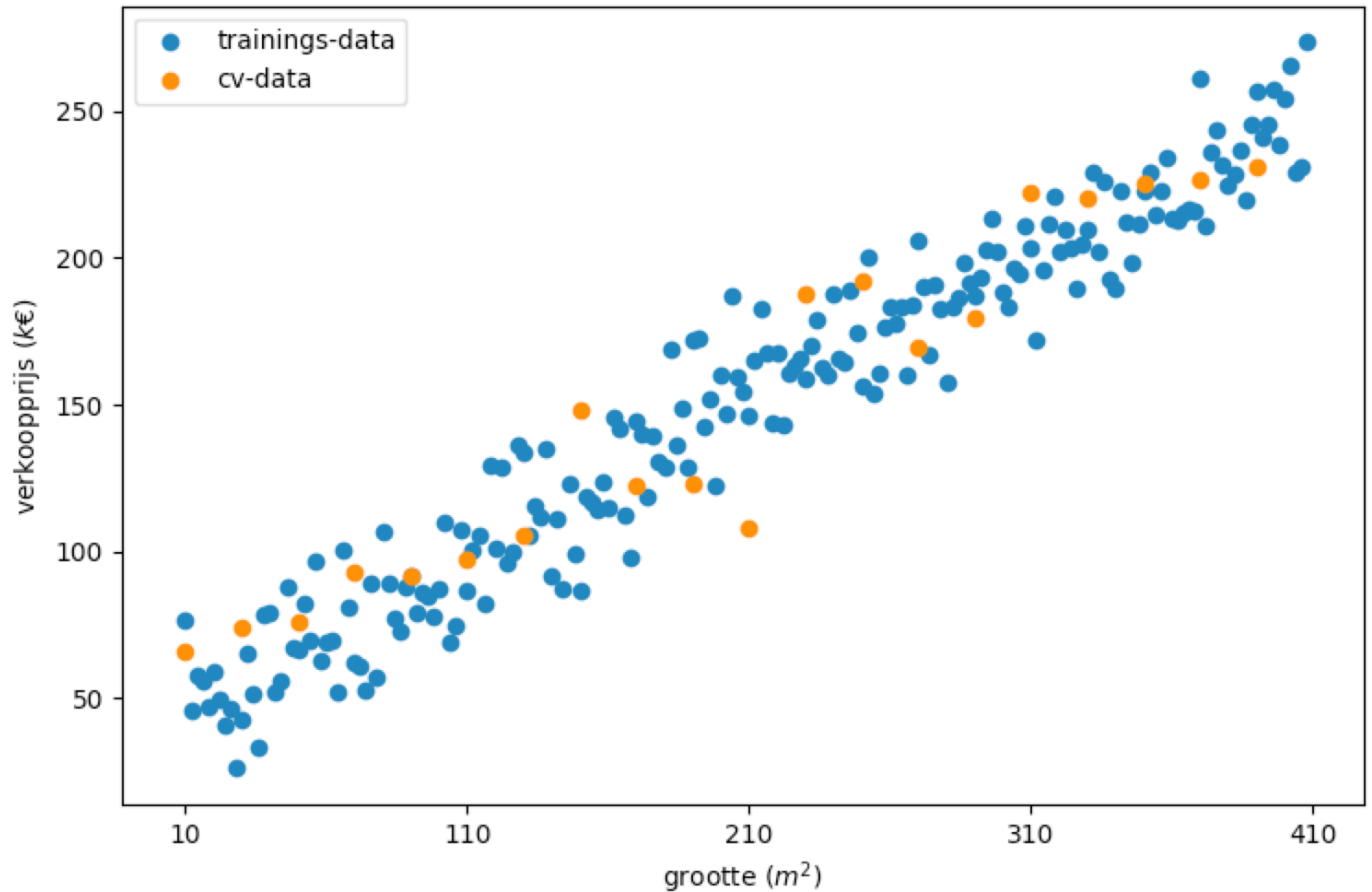
Maak verschillende hypothesen voor $\theta^{(d)}$

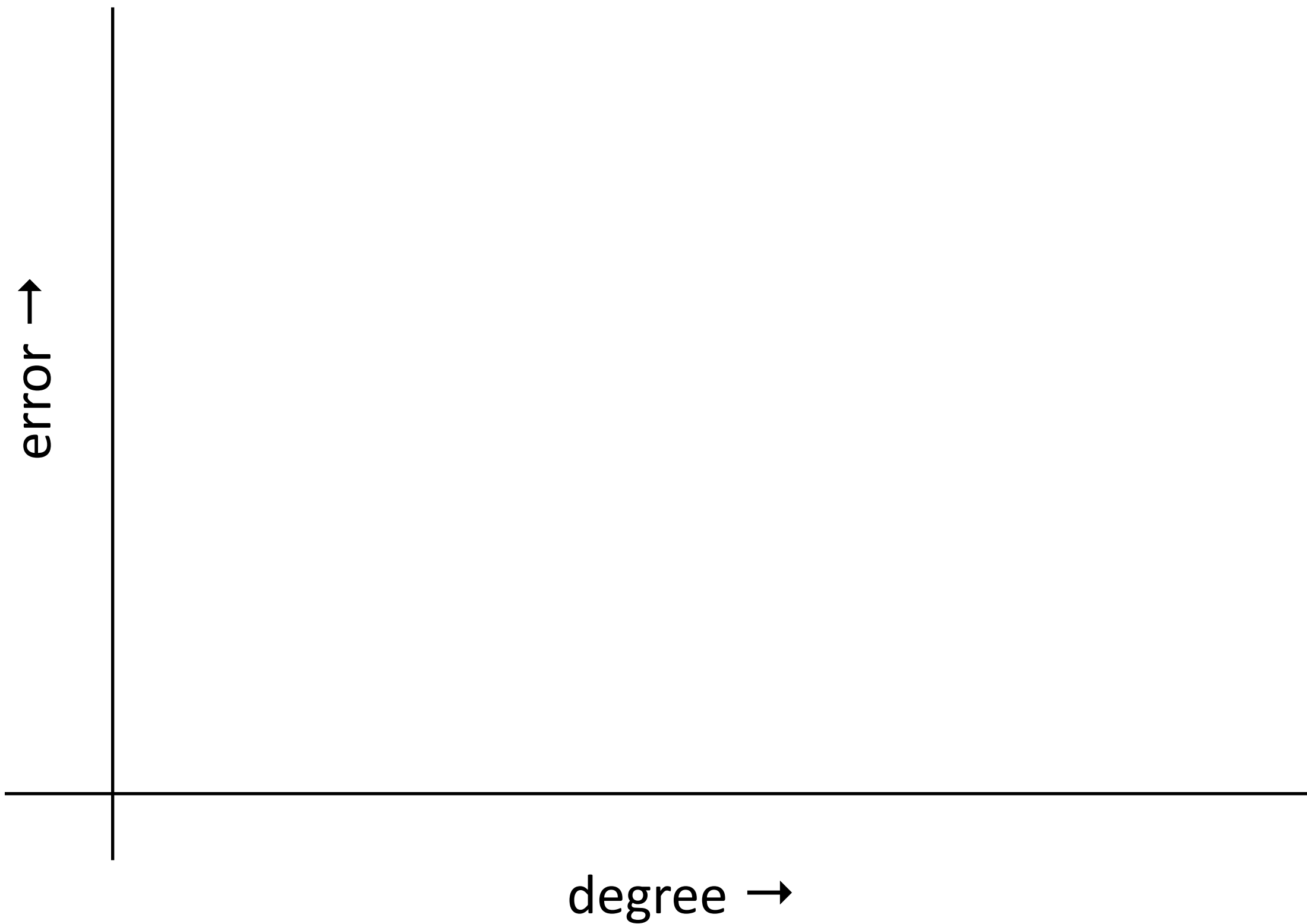
Bepaal $J_{cv}(\theta)$ voor elk van deze hypothesen

Selecteer de laagste $J_{cv}(\theta)$

Bepaal de generalisatie van deze hypothese
met behulp van $J_{test}(\theta^{(sd)})$

Verkoopprijs huizen Groningen





`ml: confusion matrix`

de confusion matrix

– voorspeld –

– werkelijk –

True Negative (TN)

False Positive (FP)

False Negative (FN)

True Positive (TP)

– voorspeld –

– werkelijk –



TN:

voorspeld als niet-spam, is
ook geen spam



FP:

voorspeld als spam, is
geen spam



FN:

voorspeld als niet-
spam, is wel spam



TP:

voorspeld als
spam, is ook spam

de confusion matrix

– voorspeld –

– werkelijk –

 <p>voorspeld als niet-5 is ook geen 5</p>	 <p>voorspeld als 5 is geen 5</p>
 <p>voorspeld als niet-5 is wel een 5</p>	 <p>voorspeld als 5 is ook een 5</p>

Beoordelingsmaten

$$Accuracy = \frac{n_{TP} + n_{TN}}{n_{total}}$$

Wanneer werkt deze maat wel en wanneer niet?

$$\textit{recall}(TPR) = \frac{TP}{TP + FN}$$

$$\textit{precision}(PPV) = \frac{TP}{TP + FP}$$

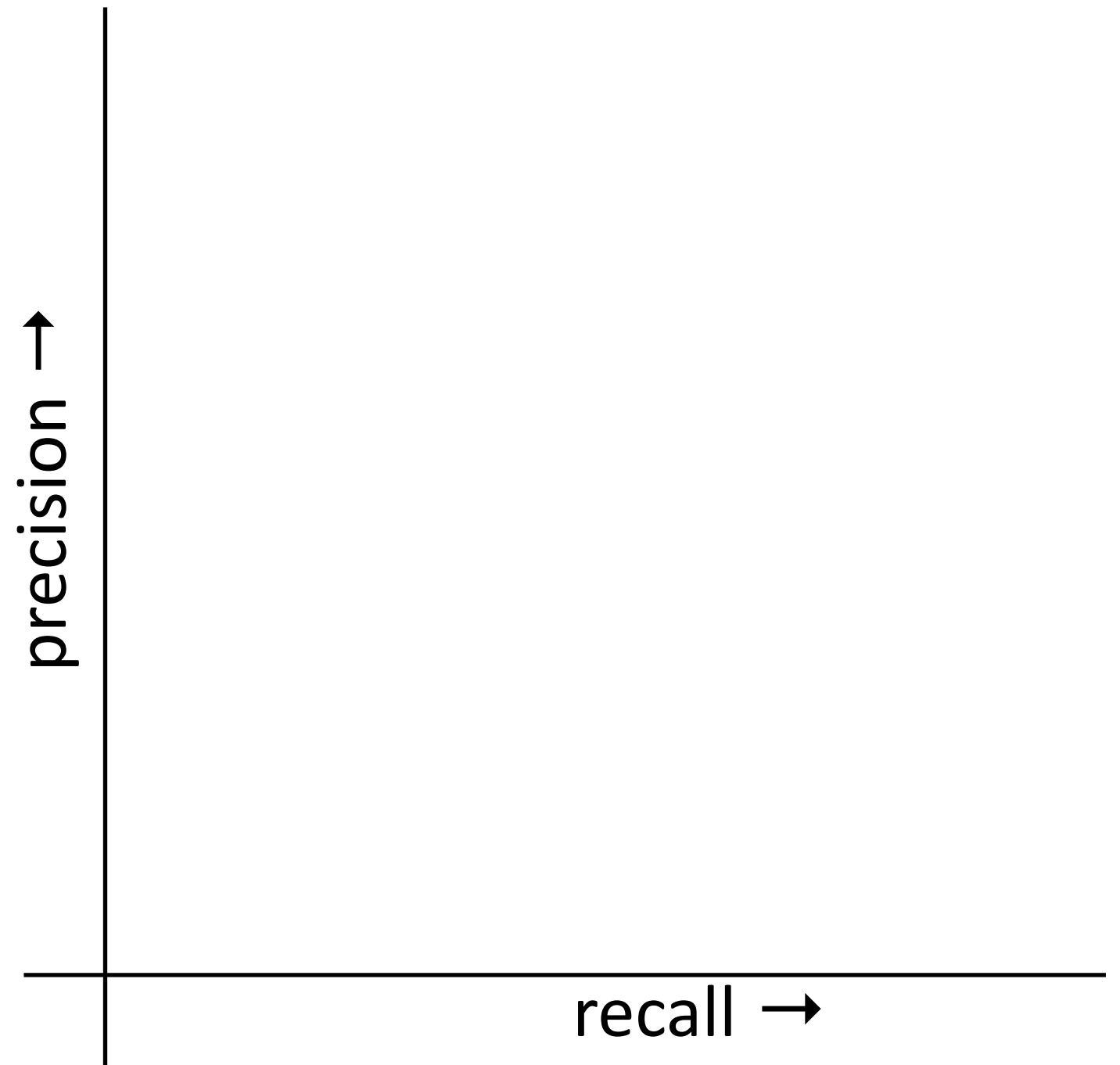
$$\textit{specificity}(TNR) = \frac{TN}{TN + FP}$$

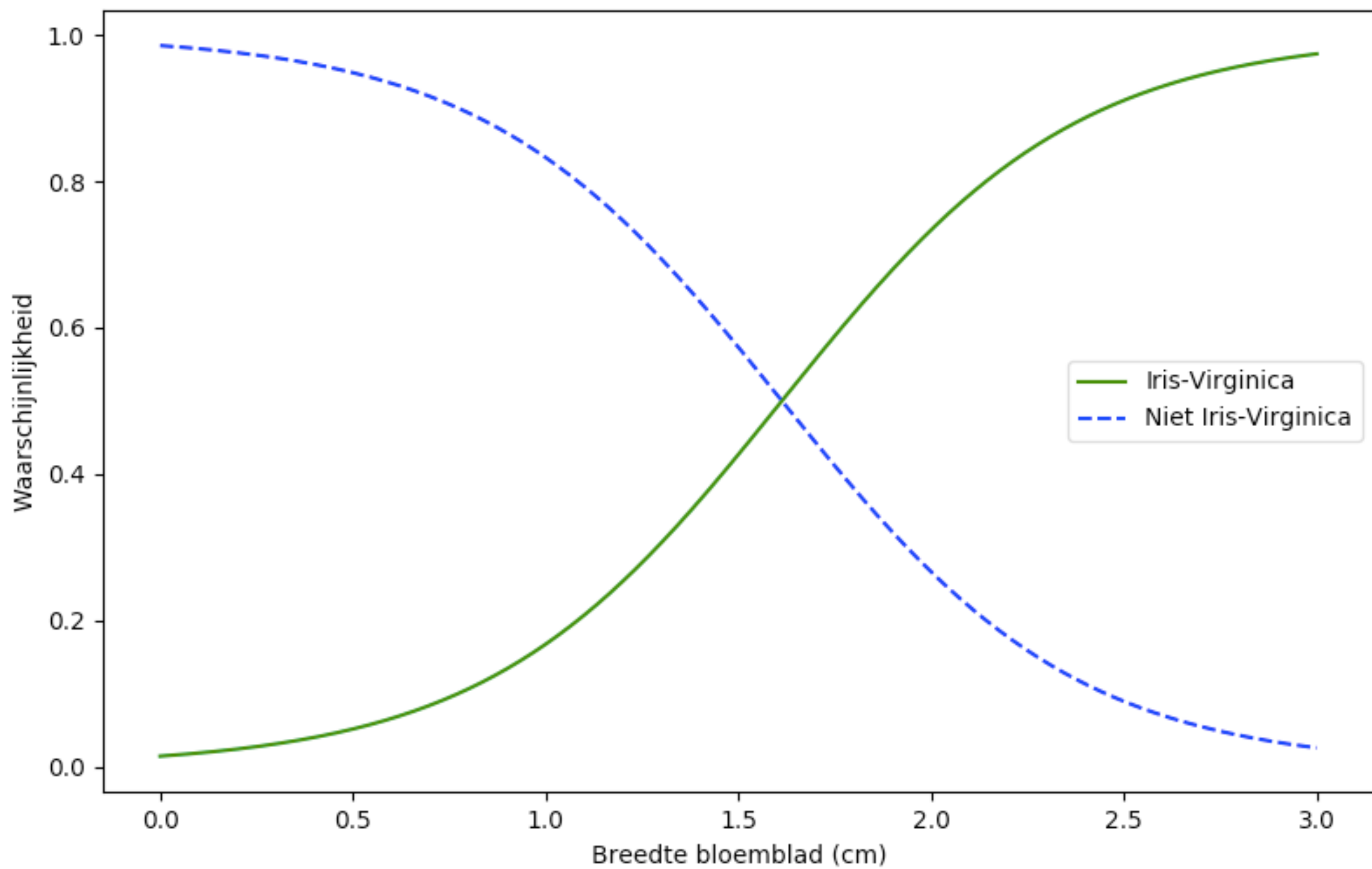
$$\textit{fall-out}(FPR) = \frac{FP}{FP + TN}$$

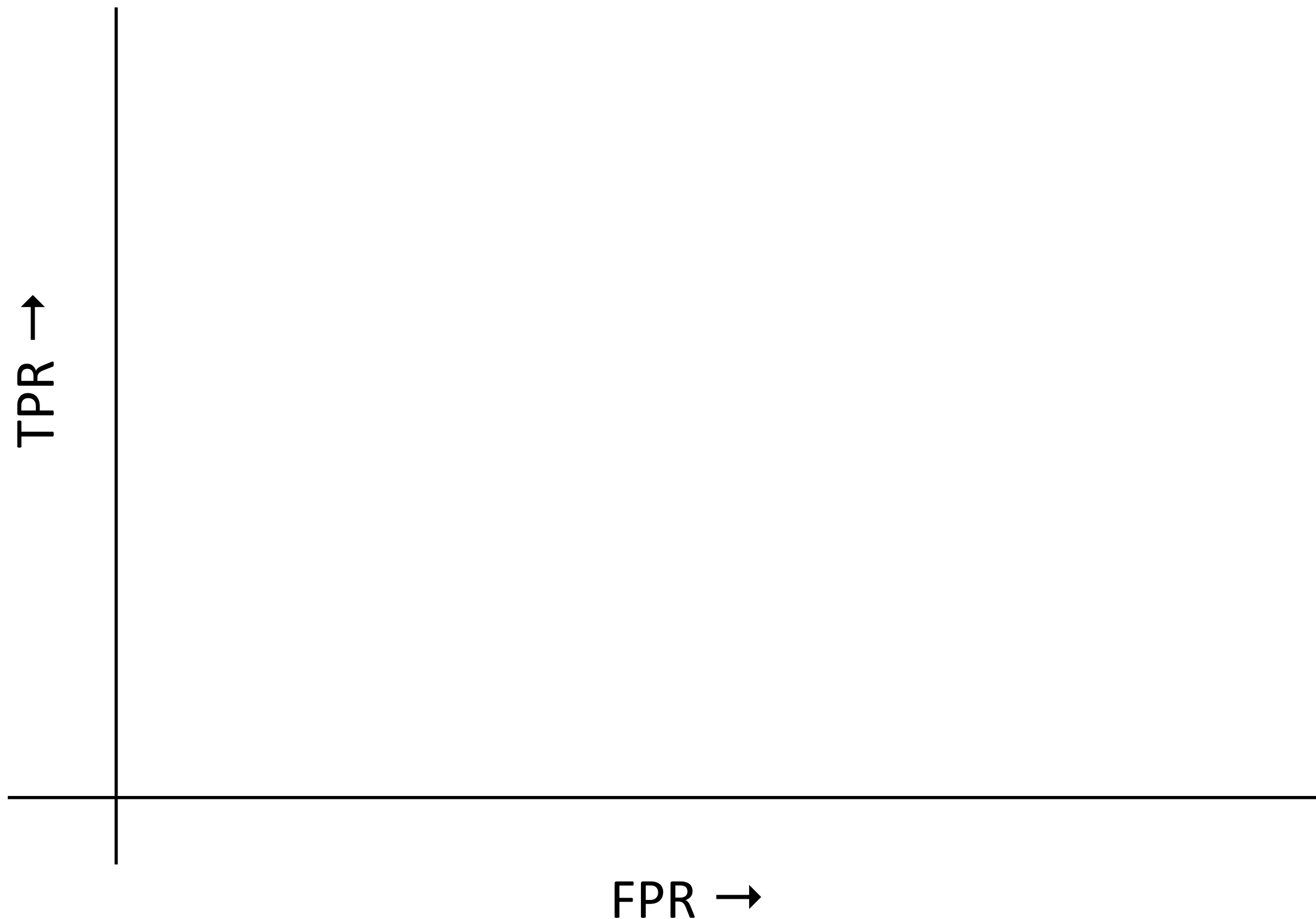
$$\text{F1-score} = 2 \times \frac{P \times R}{P + R}$$

Precision	Recall	Gemiddelde	F1-score
0,5	0,4	0,45	
0,7	0,1	0,4	
0,02	1,0	0,51	

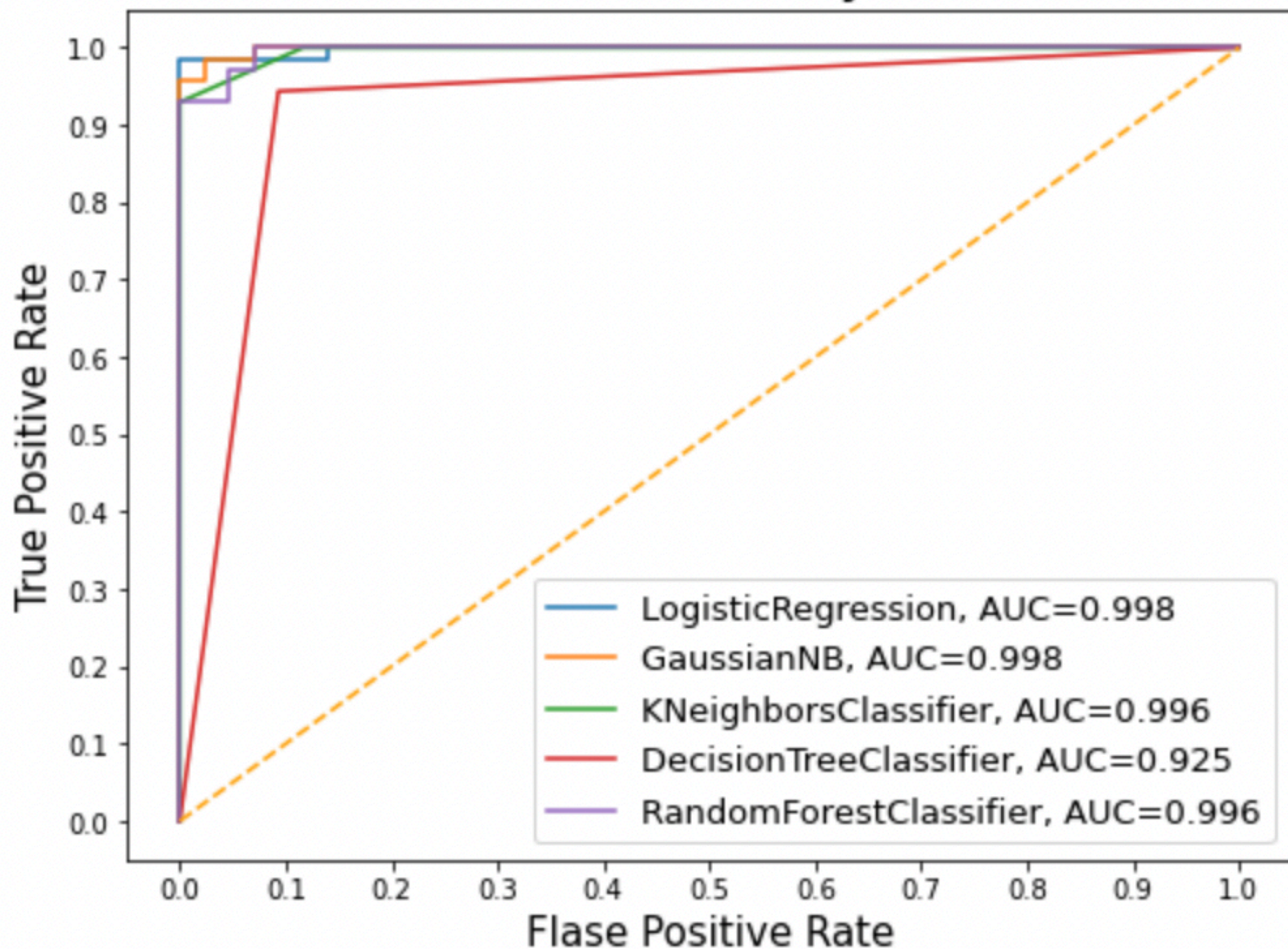
Specificity en *recall* gaan over de *zekerheid* waarmee we een voorspelling kunnen doen.







ROC Curve Analysis



Hoe nu een ML-probleem adresseren?

Welk algoritme past het best bij dit probleem?

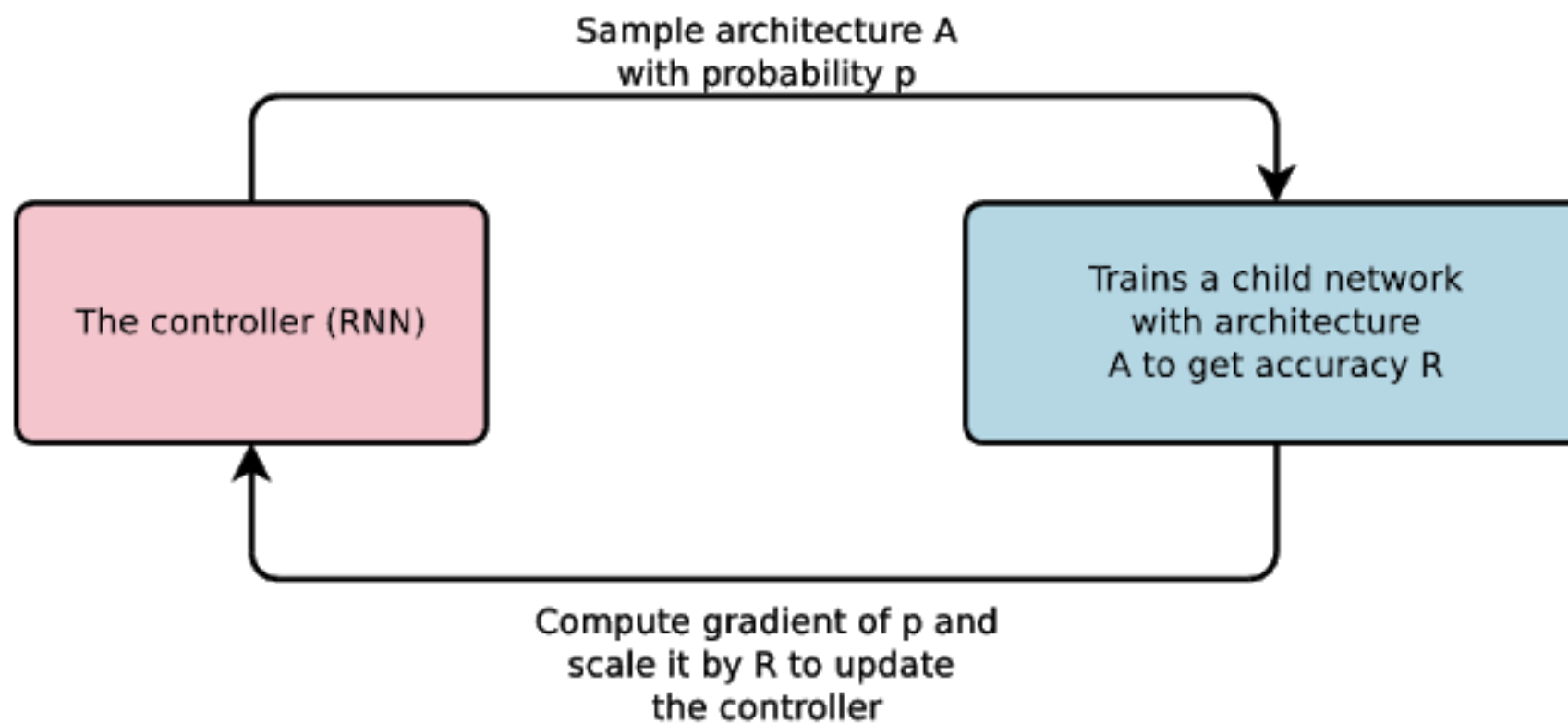
Wat zijn de beste waarden van θ ?

Wat is de beste graad van de polynoom?

Wat is de beste kernel die ik kan gebruiken?

Wat is de beste architectuur van het neurale netwerk?

Wat zijn de beste waarden van de initiële parameters?



<https://ai.googleblog.com/2017/05/using-machine-learning-to-explore.html>

