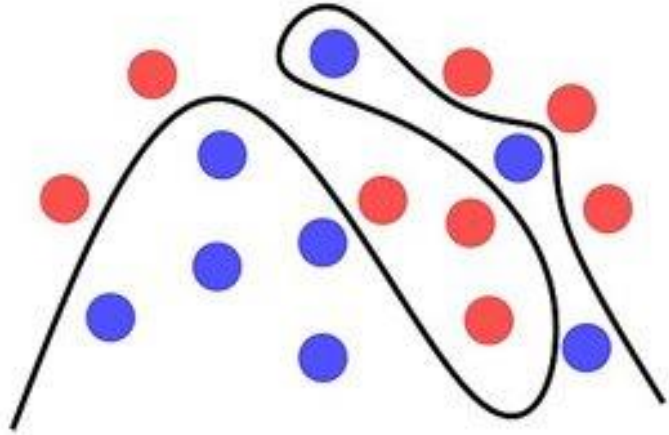


TOO LEGIT



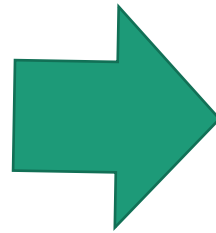
TO OVERFIT

Support Vector Machines

Jonathan Peers
IAO3B

Toepassingen

- Herkennen van handschrift
- Gezichtsherkenning
- Classificeren van documenten
- Classificatie van afbeeldingen
- Classificatie van genen (bioinformatica)

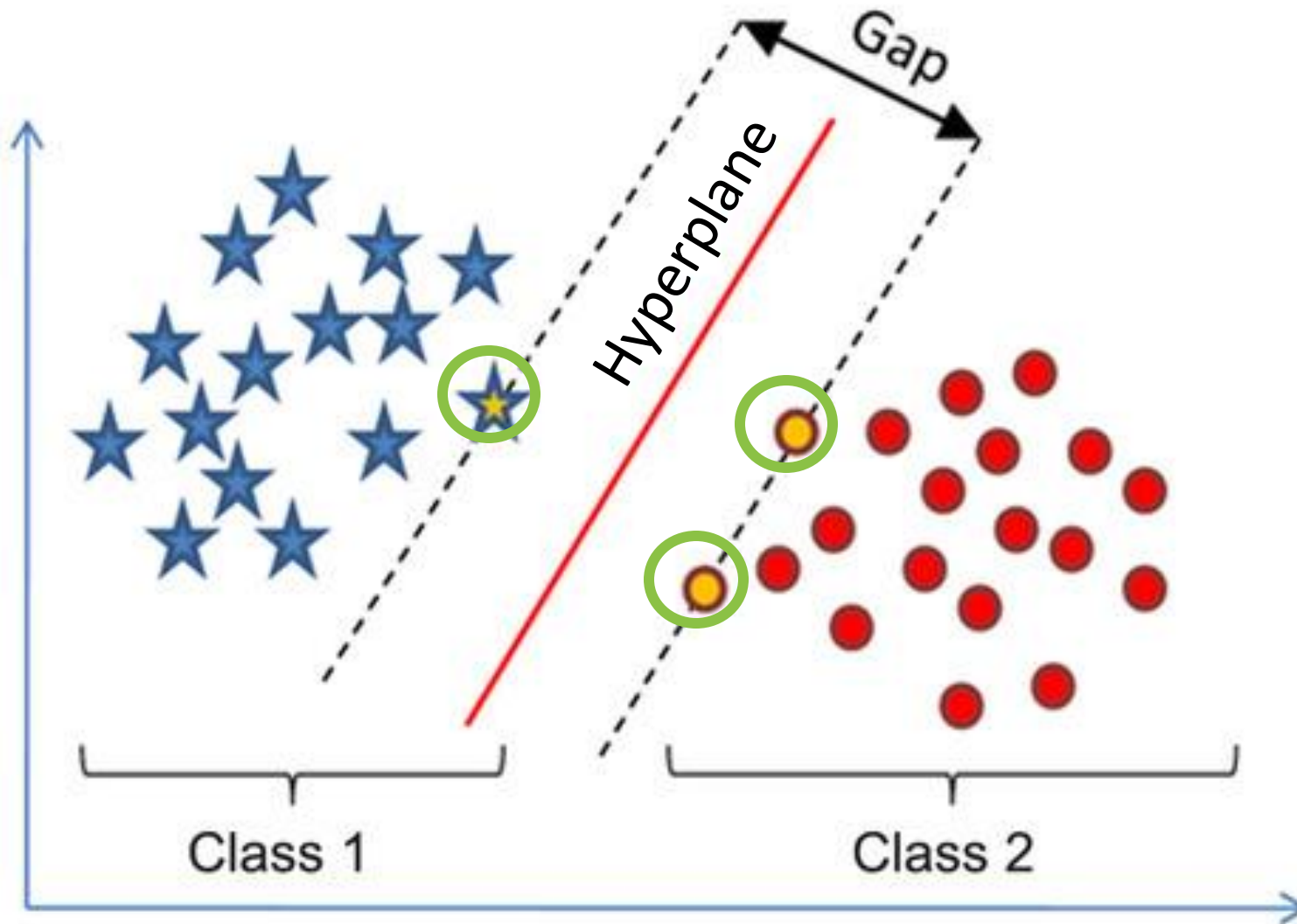


Machine learning

↳ Supervised learning

↳ **CLASSIFICATIE**

Onderdelen van een SVM



Hyperplane:

splitst klassen met
een zo groot
mogelijke margin

Margin:

afstand tussen
dichtsbijzijnde
vector v e klasse en
de hyperplane

Support vectors:

vectoren die de
hyperplane vorm
bepalen

Dus...

Een **support vector machine** probeert de **beste hyperplane** te vinden die de vectoren van klassen onderscheid.

beste hyperplane

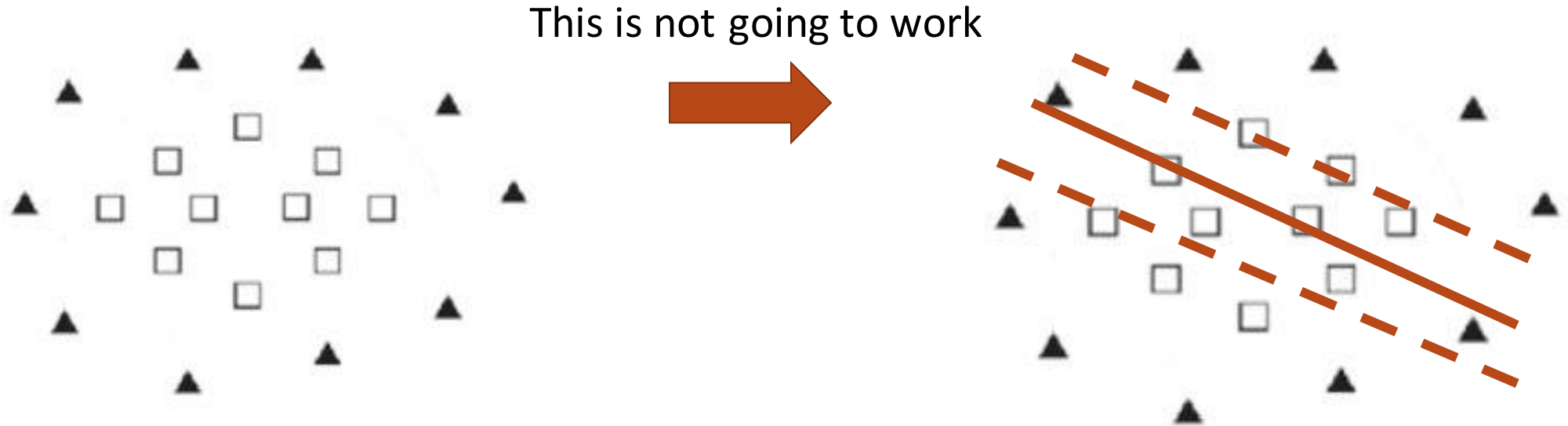
==

hyperplane grootste marge tussen de klassen

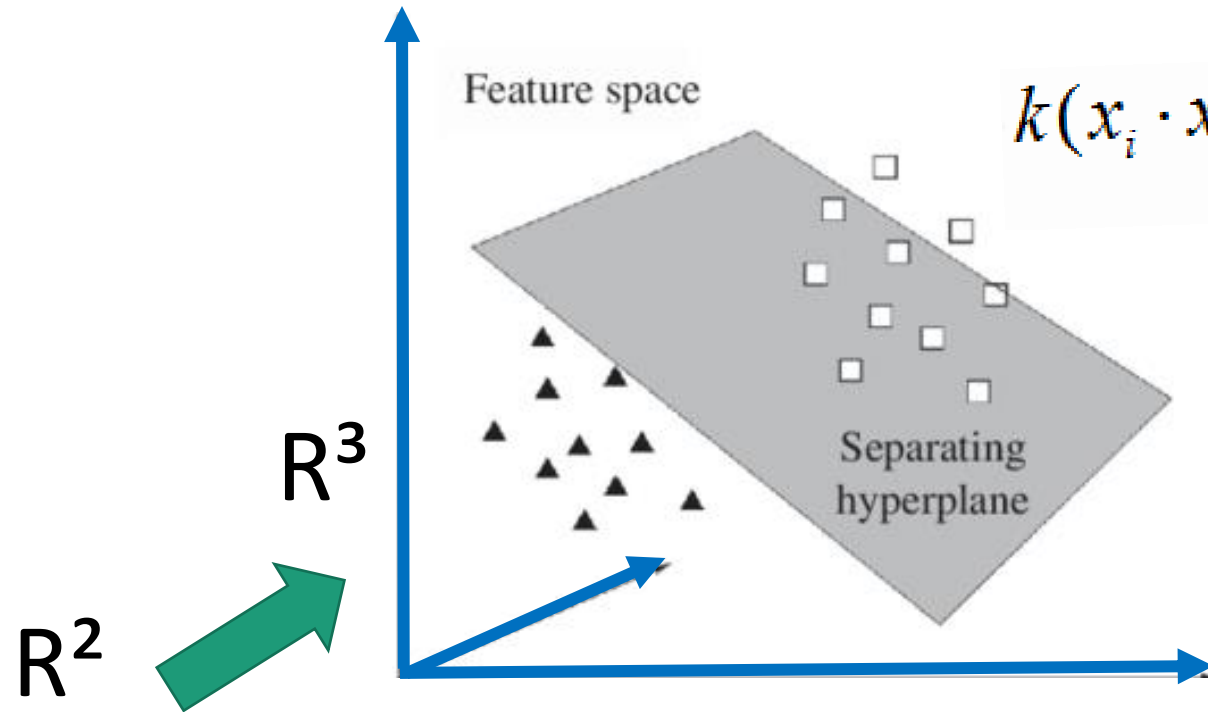
Trouble in linear paradise...

Misclassifications:

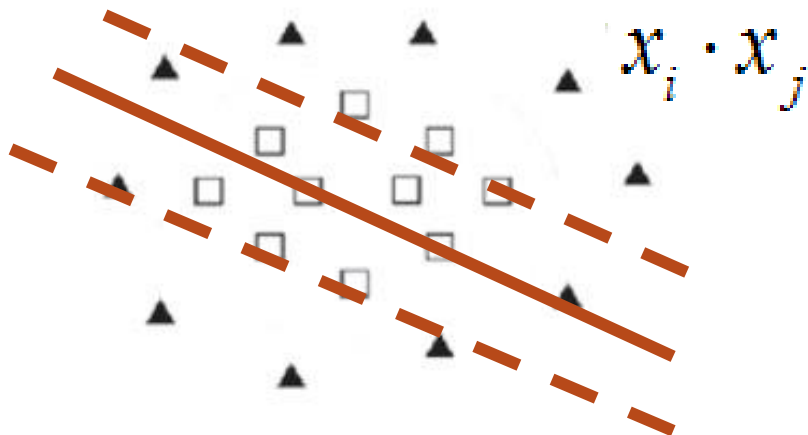
vectoren die niet aan de juiste kant vallen van een lineaire hyperplane



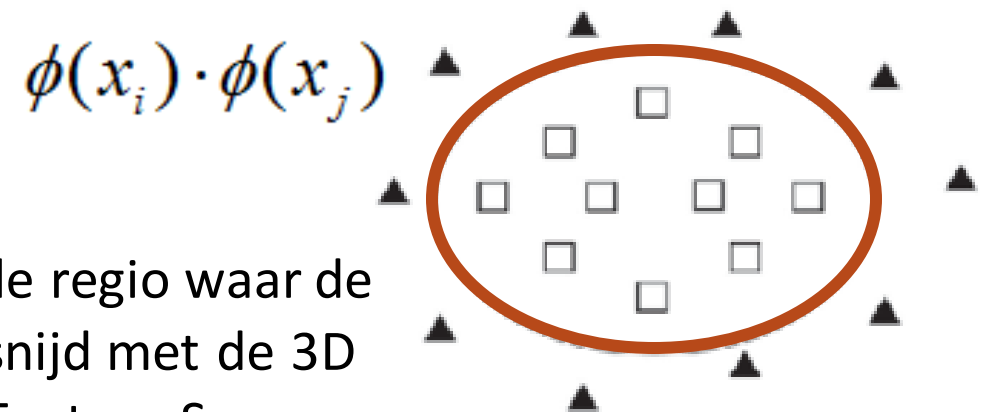
vector punten mappen in een 3D Feature Space adhv Kernel Functie



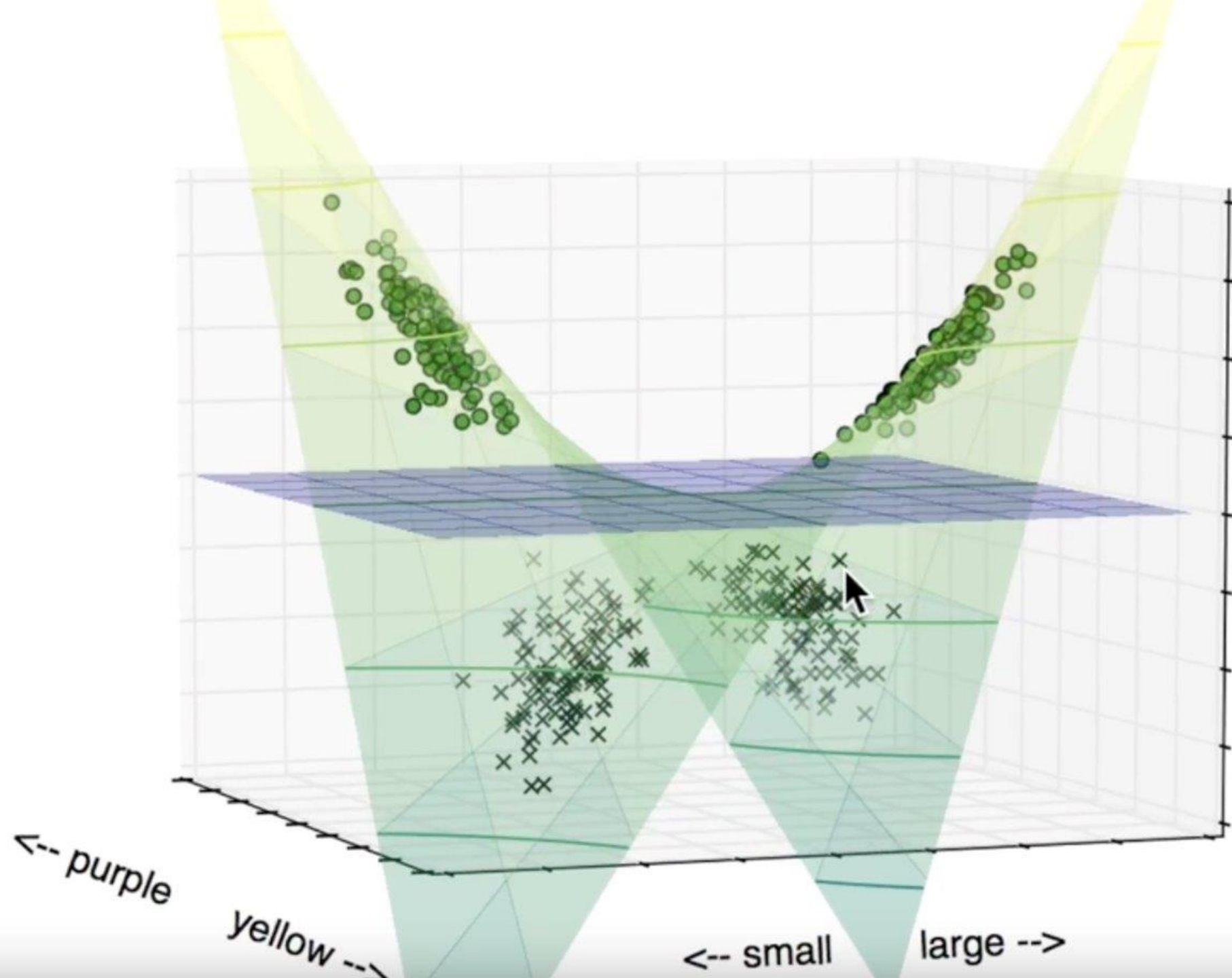
Input Space (Probleem)

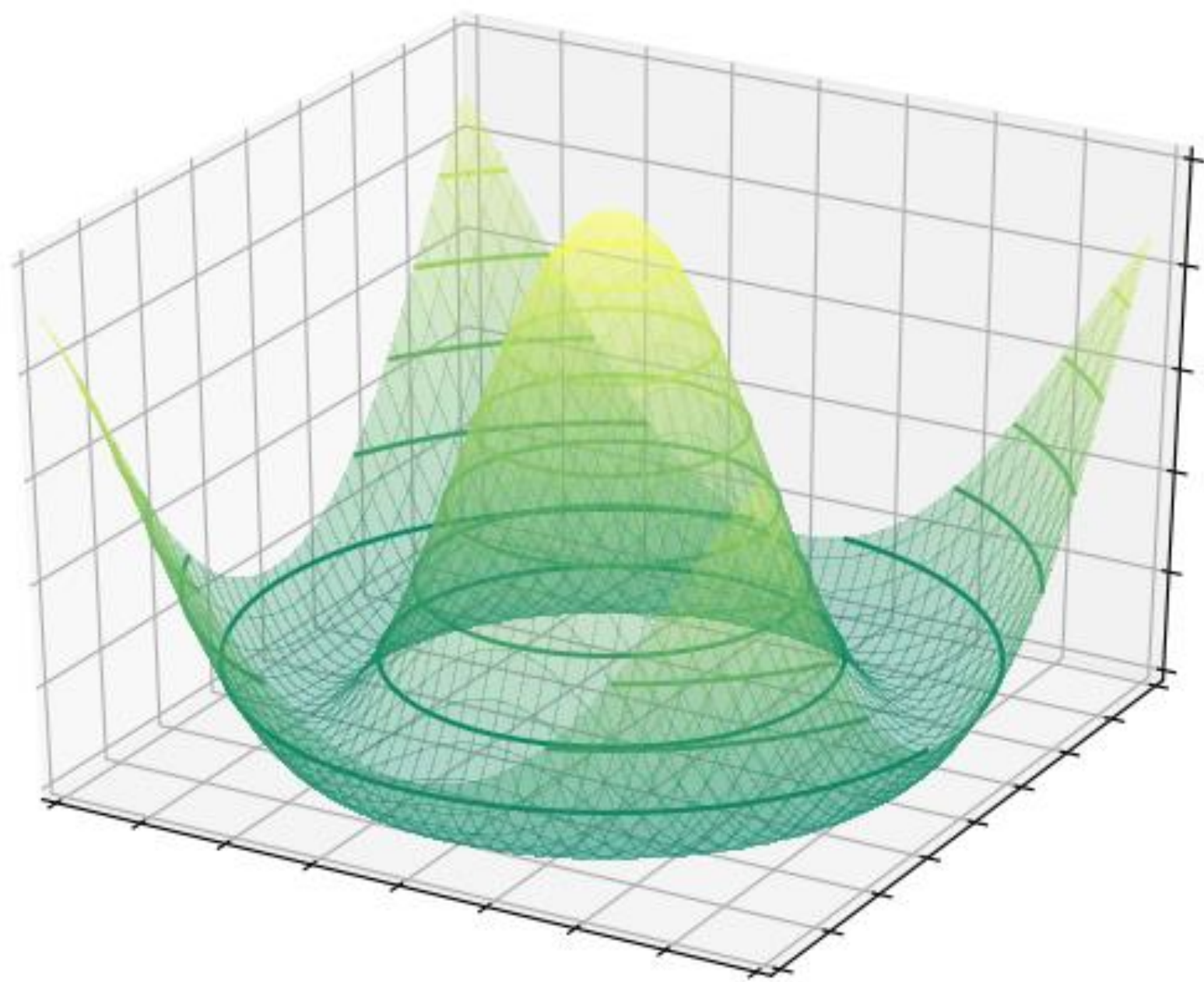


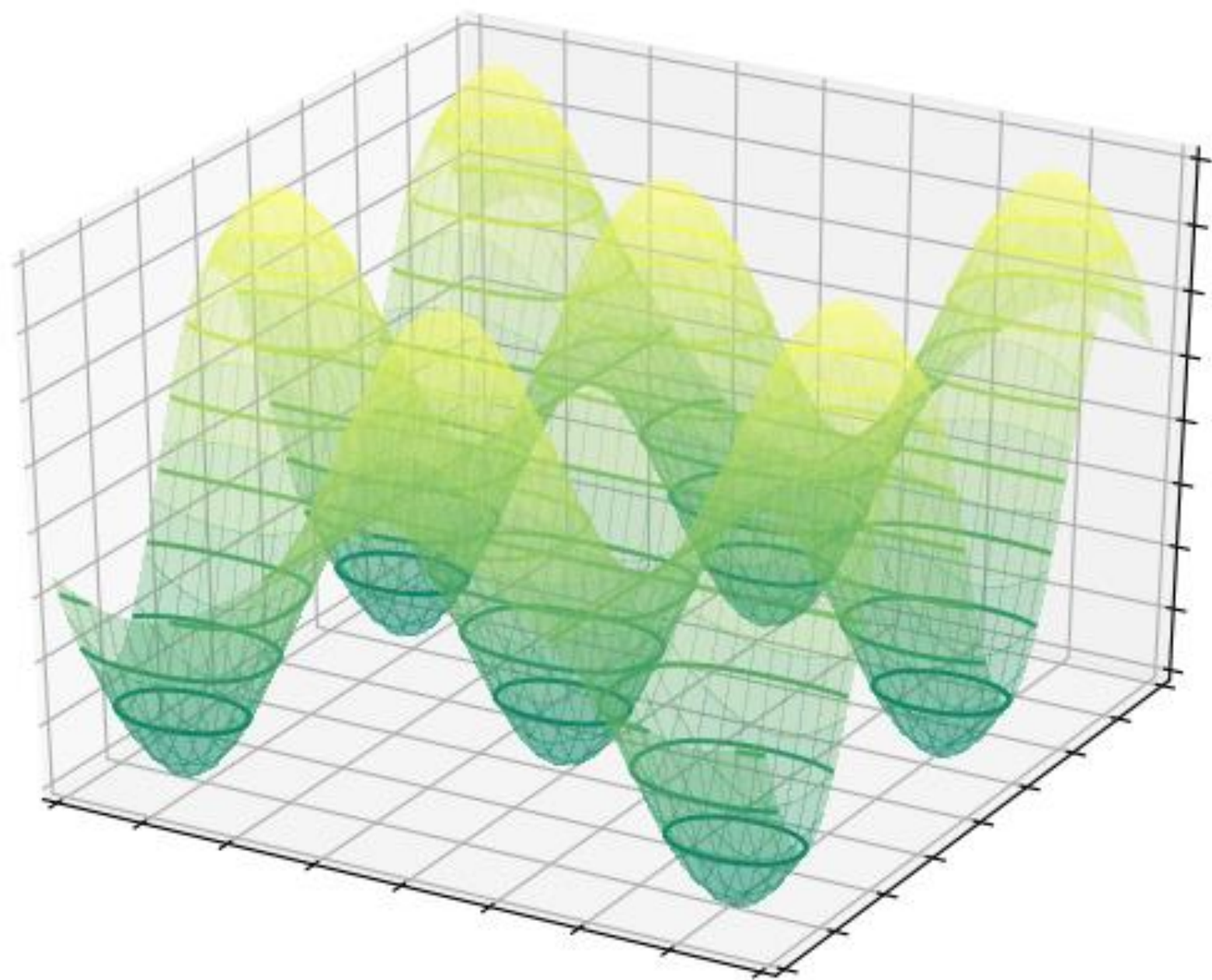
Input Space (Oplossing)



2D Ovaal is de regio waar de hyperplane snijdt met de 3D figuur in de Feature Space







Dus...

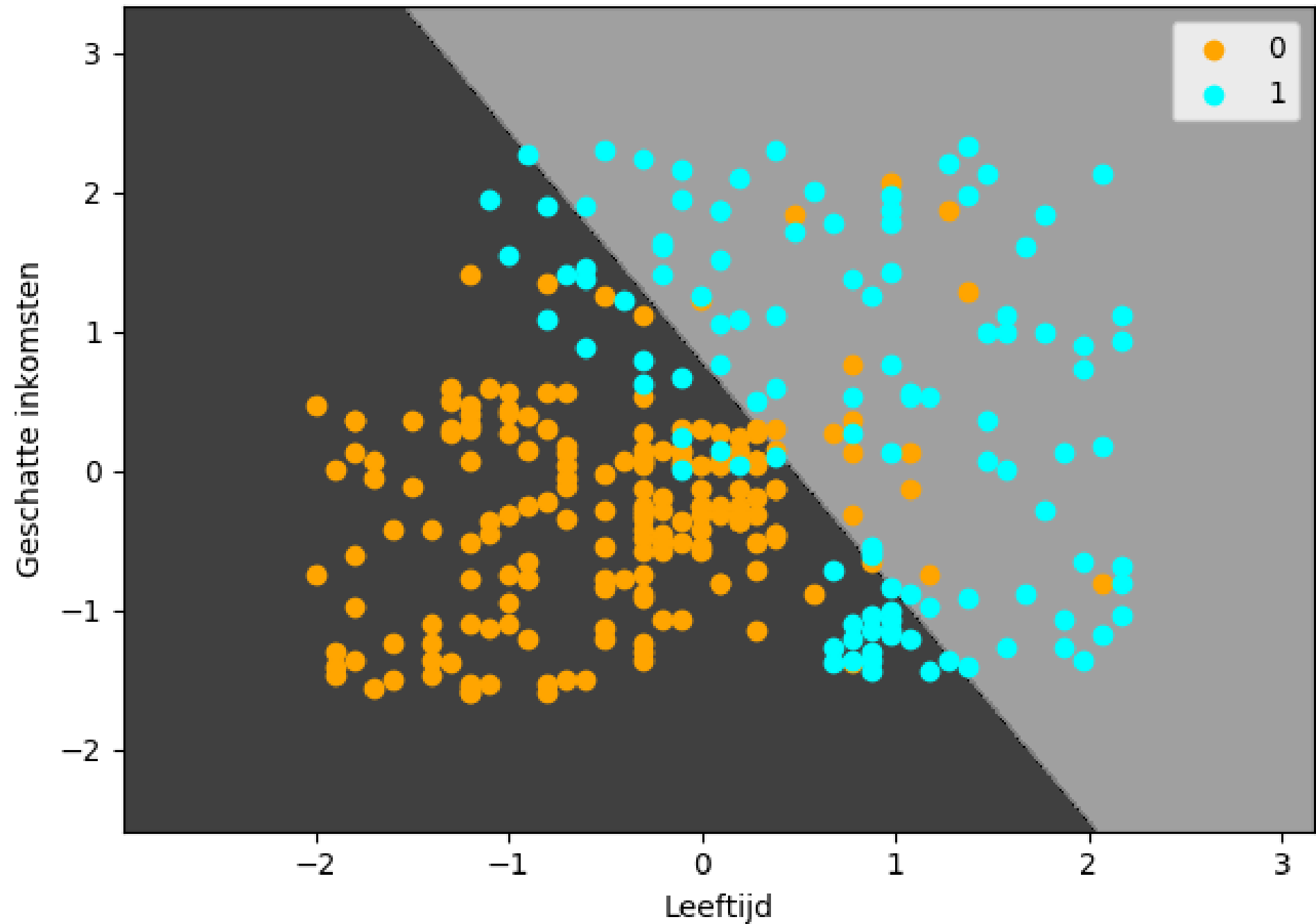
We kunnen dus ook niet-lineaire hyperplanes maken mbv de kernel functie
als er teveel misclassificatie/ruis is

extra dimensie toevoegen EN klassen boven en onder

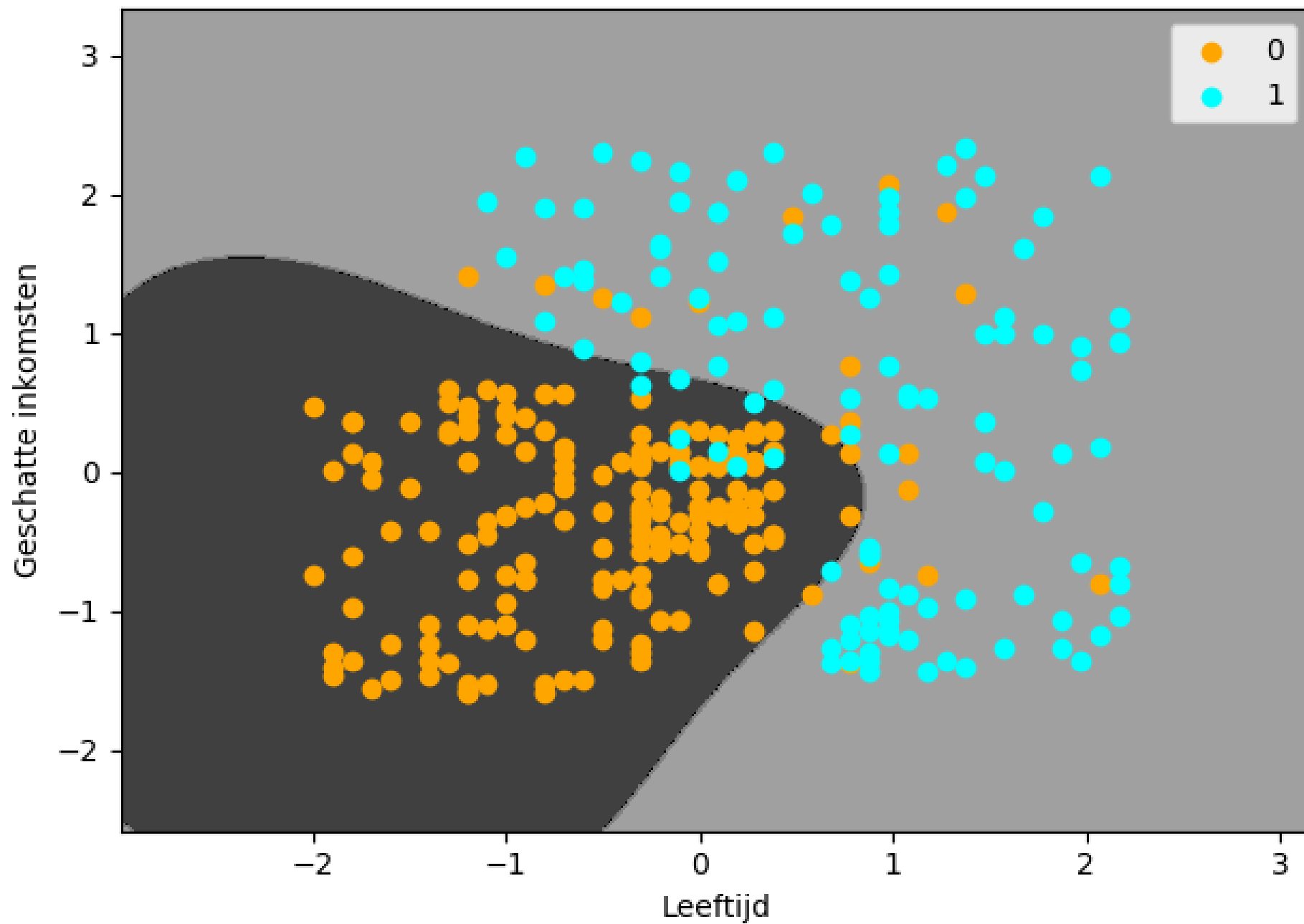
meest gebruike kernel functie voor dit probleem is de Radial Basis
Function (RBF) kernel == makkelijk!

(nog meer o.a. sigmoid, polynomial, linear...)

Standard Linear SVM (Training set)



Standard RBF kernel SVM (Training set)



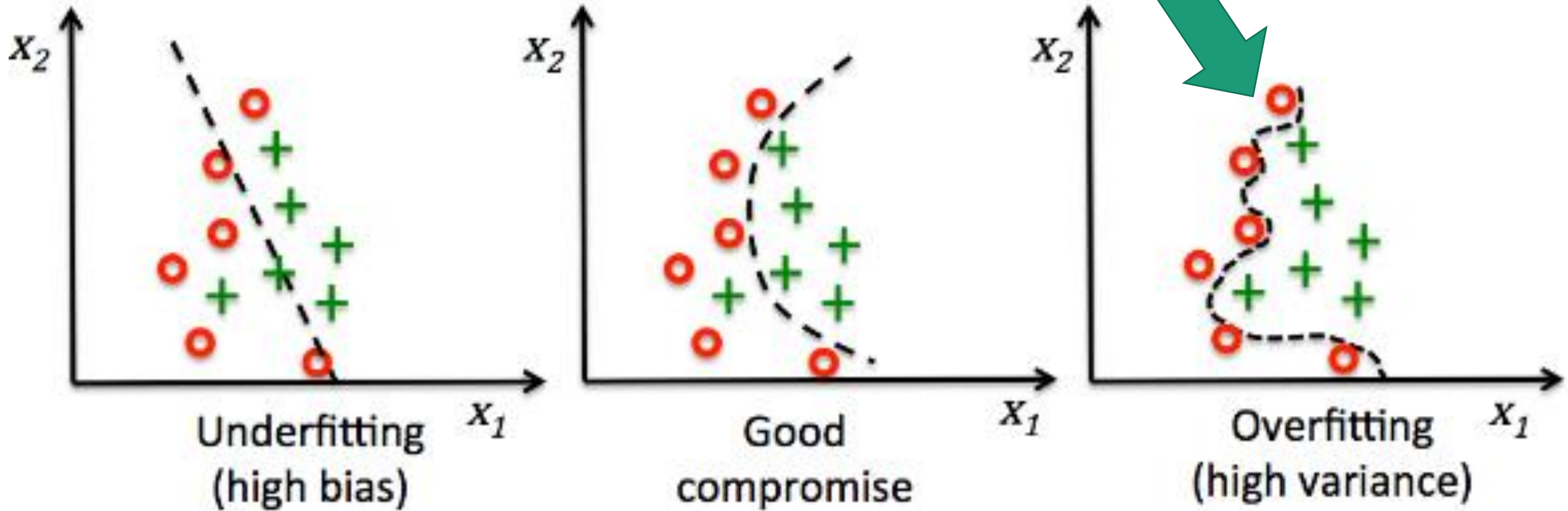
Kunnen we de groepering van
klasvectoren nog verder optimaliseren?

Optimaliseren classificatie

Non-lineaire SVM's kunnen we verder optimaliseren door de **Kernel**, **C** en **γ -waarde** aan te passen.

- **C**
 - laag \rightarrow zachte classificatie (grote margin)
 - hoog \rightarrow agressieve classificatie (kleine margin)
- **γ (gamma)**
 - bereik van een vector tov hyperplane , denk: magnetisme rondom vector
- **Kernel**
 - strategie om dot/inner/inwendig product te berekenen,
 - gebaseerd op convex optimalisatie (minimaliseren convex functies)

Pas op voor **overfitting**!



Parameters 'finetunen' of...

ScikitLearn biedt een manier om de optimale hyperparameters voor een dataset te zoeken

```
"""best params for my set"""
from sklearn.model_selection import GridSearchCV
from sklearn.svm import SVC

svc = SVC()

param_grid = {
    "kernel": ['linear', 'poly', 'rbf', 'sigmoid'],
    "gamma": [1e-1, 1e-2, 1e-3, 1e-4],
    "C": [1, 10, 100, 1000]}

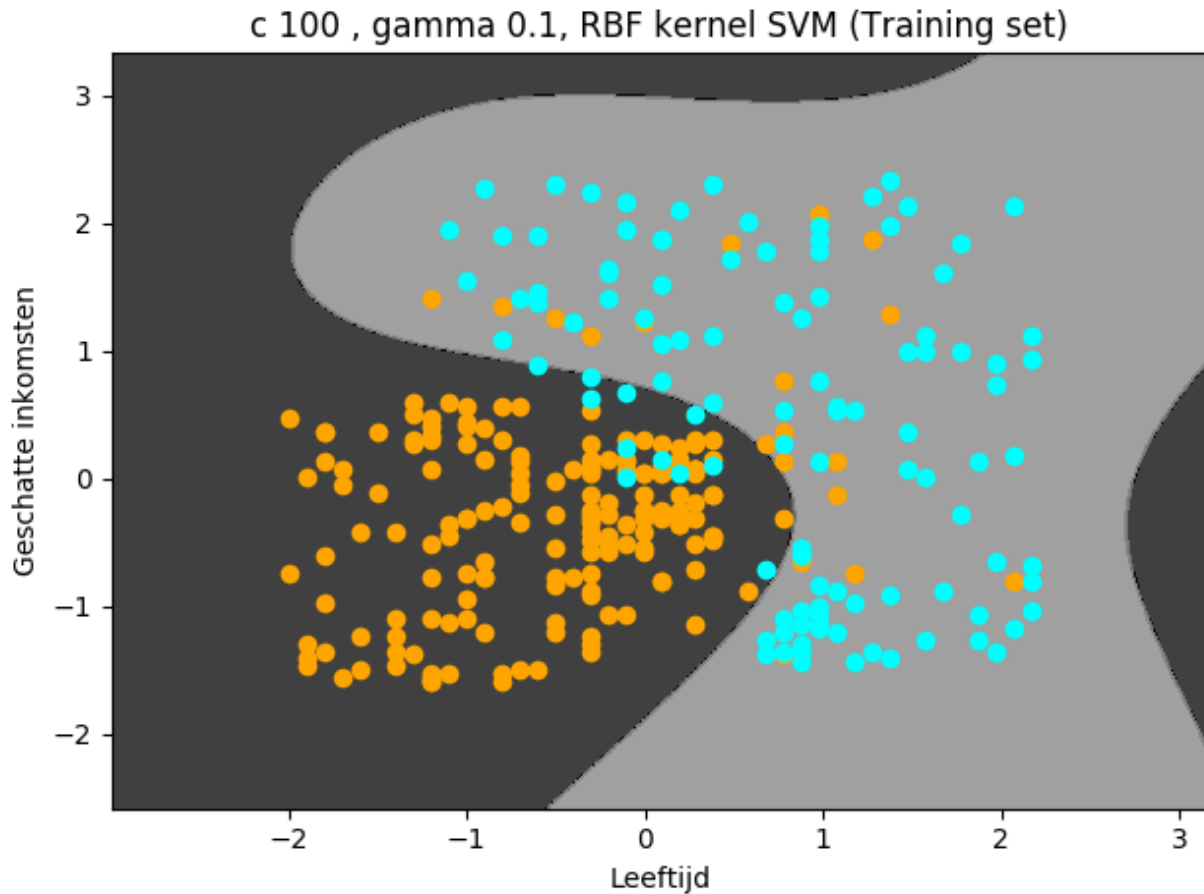
CV_svc = GridSearchCV(estimator=svc, param_grid=param_grid, cv=10)
CV_svc.fit(X_train, y_train)
print (CV_svc.best_params_)
```

Dictionary als returnwaarde

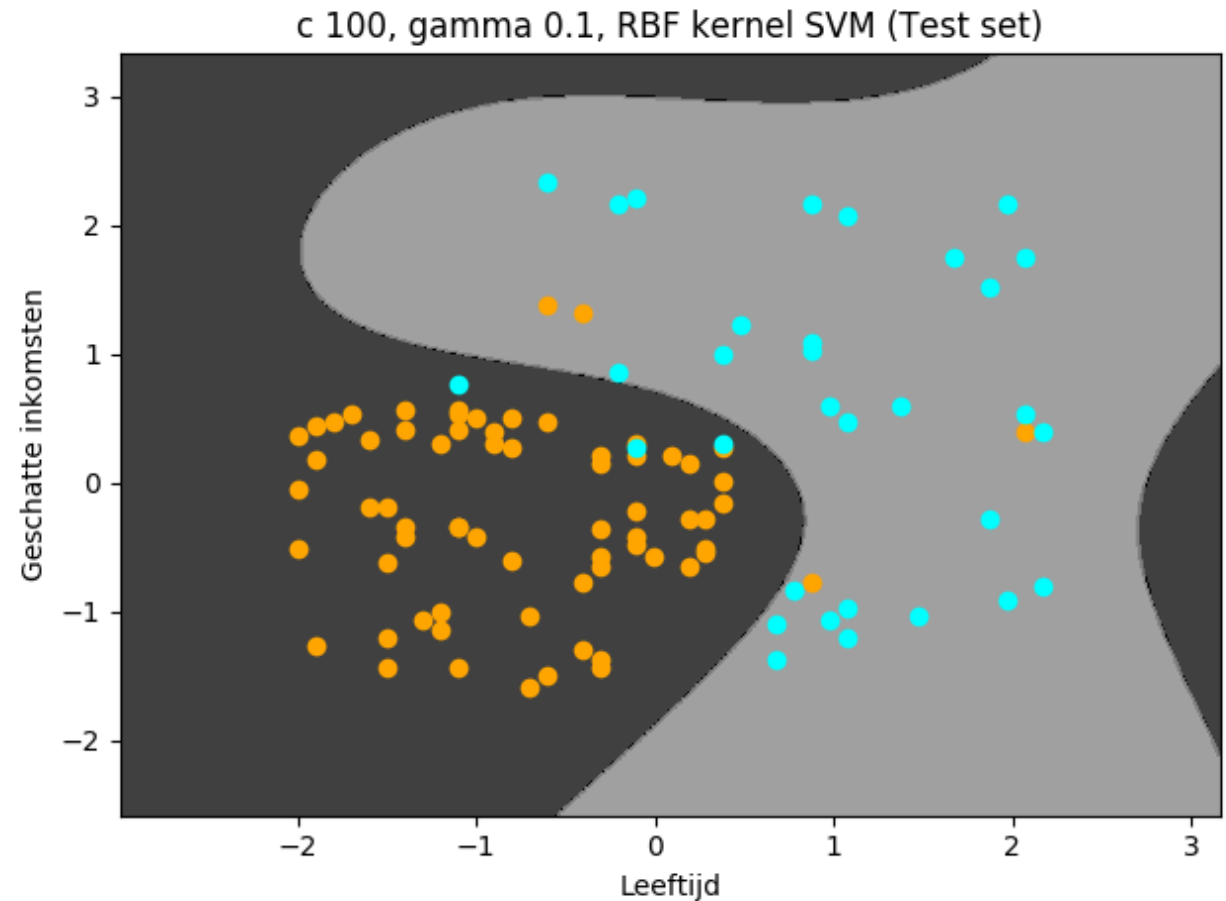
```
{  
    'kernel': 'rbf',  
    'C': 100,  
    'gamma': 0.1  
}
```

Implementatie van de door GridSearchCV voorgestelde parameters

TRAINING DATASET



TEST DATASET



Waarnemingen



Positief

- + Werkt het best op kleinere datasets
- + Goed voor binaire classificatie
- + Hoge dimensionele data

Negatief

- Grote datasets -> meer resources
- Minder accuraat bij ruis

Dank u voor uw aandacht

De code, data en presentatie kan men op deze open repository vinden:

<https://github.com/TheRealJonathanPeers/SVM>