

NEAREST NEIGHBOURS

JENS BAETENS

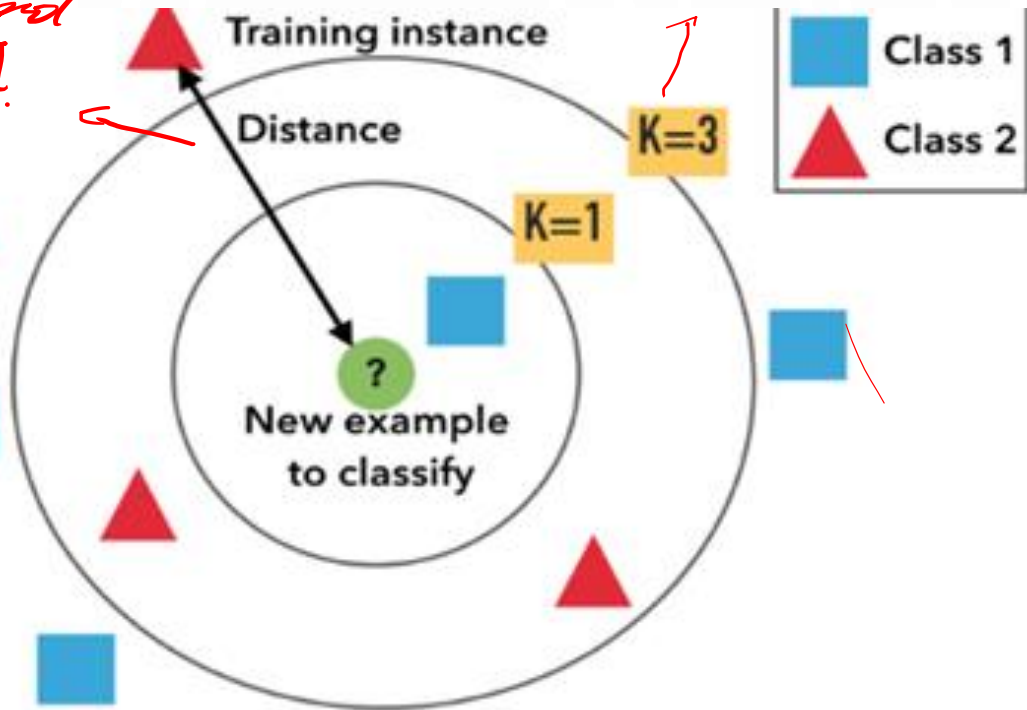
Euclidean distance

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Scaling!

regularisation / overfitting

belongs hyperparameters



KENMERKEN

*Werk
uitgevoerd* Supervised learning

training is snel
predict is traag
Lazy Learning / Geen trainingsfase →

Heel eenvoudige techniek door te kijken naar meest gelijkaardige observaties.

Classifier: Klasse is meest voorkomende in de buurt

Regressie: Gemiddelde van de targets van de buren

↳ ensemble ook voor regressie

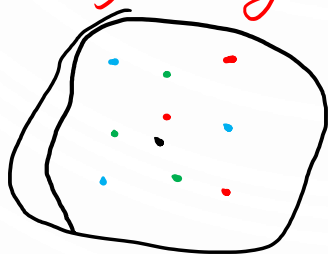
WERKING

Selecteer K dichtste buren (K is hyperparameter)

- In het geval van twee klassen: neem aantal oneven

Laat elke buur stemmen voor het resultaat

↳ gem. afstand
↳ ~~may~~ majority voting om conflicten op te lossen



VOORDELEN

*dikts te overen zullen complexer
↑ - meer ~~keerpunten~~ punten overlopen*

Accurater met meer data (maar berekeningstijd neemt wel toe)

Kan goed om gaan met ruis door meerdere burens te bekijken

Geen trainingsfase en kan goed complexe modellen leren

NADELEN

afstand berekenen complex

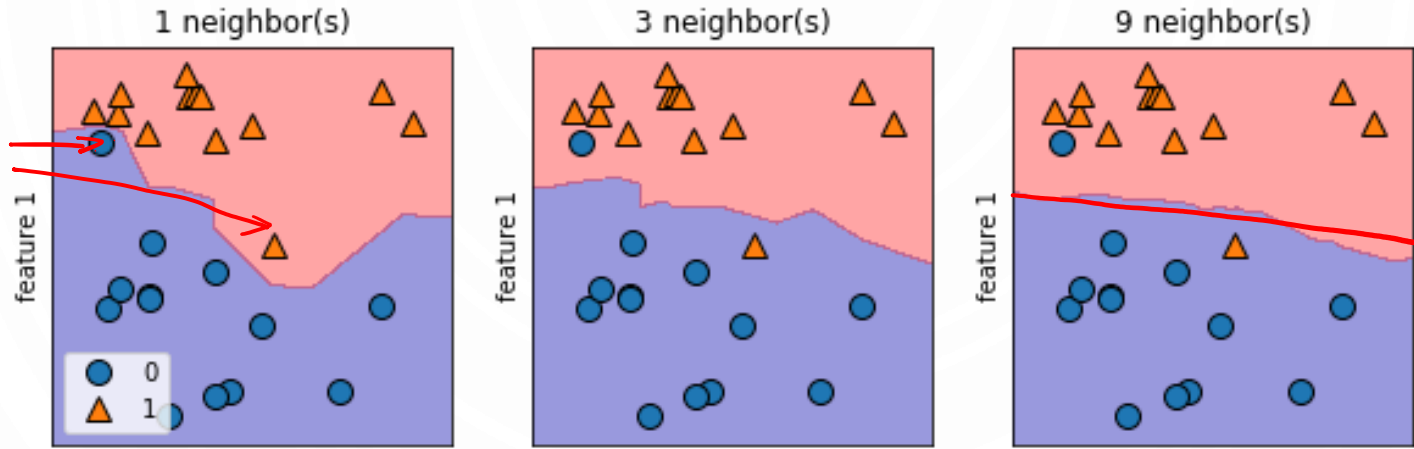
Minder efficient bij veel features

Weinig mogelijkheden om het algoritme te optimaliseren met hyperparameters aan te passen.

(bijna onbekijk)

(hoe afstand berekend wordt.)

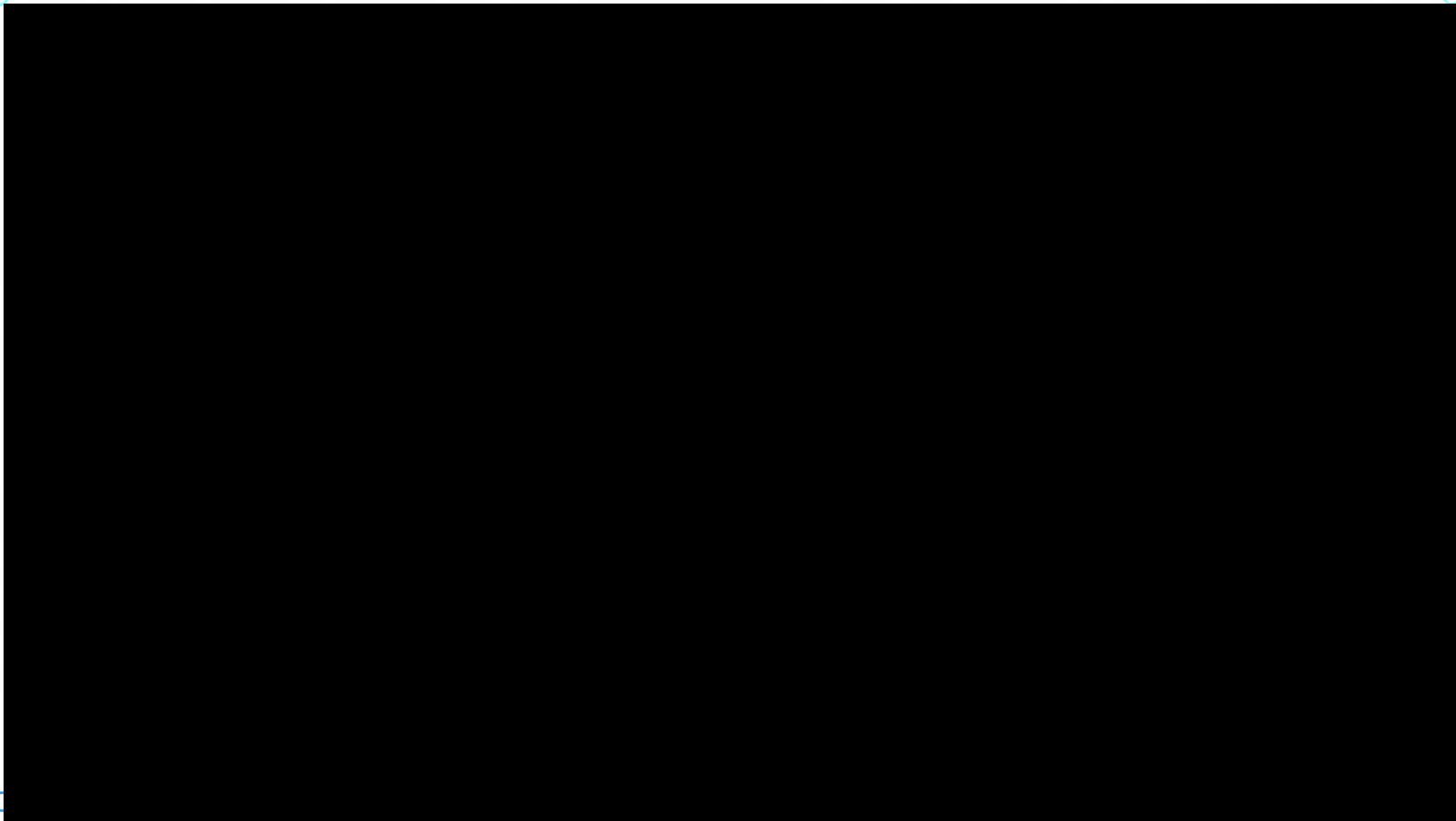
TUNING OF HYPERPARAMETER K



Lage K-waarde -> neiging tot overfitting

Grote K-waarde -> neiging tot underfitting

↳ K ook impact op efficiëntie → begin met kleine K



<https://www.youtube.com/watch?v=MDniRwXizWo>