

# Opdracht 2 – Bouw een Decision Tree voor Product Aankopen

## Doel van deze opdracht

In deze opdracht ga je zelf een Decision Tree bouwen voor een productaankoopscenario. Je leert:

- hoe een Decision Tree werkt door er zelf één te maken
- hoe de boom kiest welke vraag eerst gesteld wordt
- welke features (kenmerken) sterk of zwak zijn
- hoe een model automatisch regels maakt
- waarom modellen uitzonderingen nooit perfect kunnen oplossen

Je gebruikt hiervoor een dataset met 20 personen die wel of niet een product kopen.

## Trainingsdataset (20 personen)

Persoon	Price_range	Reviews	Discount	Categorie	Koopt_product
A	Laag	Goed	Ja	Elektronica	Ja
B	Laag	Goed	Nee	Elektronica	Ja
C	Laag	Gemiddeld	Ja	Kleding	Ja
D	Laag	Gemiddeld	Nee	Kleding	Ja
E	Laag	Slecht	Ja	Boeken	Nee
F	Laag	Slecht	Nee	Boeken	Nee
G	Middel	Goed	Ja	Elektronica	Ja
H	Middel	Goed	Nee	Elektronica	Ja
I	Middel	Gemiddeld	Ja	Kleding	Ja
J	Middel	Gemiddeld	Nee	Kleding	Nee

Persoon	Price_range	Reviews	Discount	Categorie	Koopt_product
K	Middel	Slecht	Ja	Boeken	Nee
L	Middel	Slecht	Nee	Boeken	Nee
M	Hoog	Goed	Ja	Elektronica	Ja
N	Hoog	Goed	Nee	Elektronica	Nee
O	Hoog	Gemiddeld	Ja	Kleding	Nee
P	Hoog	Gemiddeld	Nee	Kleding	Nee
Q	Hoog	Slecht	Ja	Boeken	Nee
R	Hoog	Slecht	Nee	Boeken	Nee
S	Laag	Goed	Ja	Boeken	Ja
T	Middel	Gemiddeld	Nee	Boeken	Nee

 **Let op:** net als in echte data zijn er uitzonderingen:

- Persoon **H** koopt wél zonder korting
- Persoon **O** koopt niet, ondanks reviews Gemiddeld + korting

Hou dit in gedachten: een model kan nooit 100% perfect zijn.



## Jullie opdracht

Werk in groepjes. Je gaat *stap voor stap* een Decision Tree bouwen.

## Stap 1 – Kies de eerste splitsing

Kies een feature voor de **eerste vraag** in de boom.

Denk na:

Welke feature verdeelt de dataset het **netst** in homogene groepen?

### **Jullie eerste splitsing:**

- Feature: \_\_\_\_\_
- Vraag: \_\_\_\_\_
- Waarom deze feature?

#### **Tip:**

Als één groep bijna volledig **Ja** of **Nee** is, dan is dit vaak een sterke eerste keuze.

## **Stap 2 – Maak 2 groepen**

Verdeel de dataset in 2 groepen op basis van jullie eerste splitsing.

### **Groep A:**

- Personen: \_\_\_\_\_
- Aantal personen: \_\_\_\_\_
- Aantal Ja: \_\_\_\_\_
- Aantal Nee: \_\_\_\_\_
- Homogeen? (Ja/Nee): \_\_\_\_\_

### **Groep B:**

- Personen: \_\_\_\_\_
- Aantal personen: \_\_\_\_\_
- Aantal Ja: \_\_\_\_\_
- Aantal Nee: \_\_\_\_\_
- Homogeen? (Ja/Nee): \_\_\_\_\_

#### **Tip:**

Een groep is homogeen als bijna iedereen dezelfde uitkomst heeft.

## Stap 3 – Kies een tweede splitsing

Kies nu een splitsing voor **één van de groepen die nog niet homogeen is.**

- Groep: \_\_\_\_\_
- Feature: \_\_\_\_\_
- Vraag: \_\_\_\_\_
- Waarom deze feature?

**Let op:**

- De tweede splitsing hoeft **niet** dezelfde feature te gebruiken als de eerste.
- Stoppen mag ook, als de groep voldoende “netjes” verdeeld is.

## Stap 4 – Eventueel een derde splitsing

Indien nodig kies je nog een splitsing.

- Groep: \_\_\_\_\_
- Feature: \_\_\_\_\_
- Vraag: \_\_\_\_\_
- Waarom deze feature?

**Richtlijn:**

- Bomen met **3–4 lagen** zijn oké
- Bomen met **5+ lagen** zijn vaak overfitting

# **Stap 5 – Teken de Decision Tree**

Teken de boom:

- bovenaan de eerste vraag
- splitsingen eronder
- eindpunten (bladeren) met “Ja” of “Nee”

# **Stap 6 – Schrijf de regels op**

Elke eindpunt is één regel.

**Regel 1:**

Als \_\_\_\_\_ → \_\_\_\_\_

**Regel 2:**

Als \_\_\_\_\_ EN \_\_\_\_\_ → \_\_\_\_\_

**Regel 3:**

Als \_\_\_\_\_ EN \_\_\_\_\_ → \_\_\_\_\_

**Regel 4:**

Als \_\_\_\_\_ EN \_\_\_\_\_ EN \_\_\_\_\_ → \_\_\_\_\_

# **Stap 7 – Reflectie**

Beantwoord:

1. Welke features waren sterk in jullie boom?

**2. Welke features waren minder belangrijk, zwak of redundant?**

(Zwak = weinig informatie, Redundant = informatie overlapt met een andere feature)

**3. Waarom waren deze features zwak of redundant?**

**4. Is jullie boom overfit, underfit of OK? Leg uit.**