Data Foundations

Hoofdstuk 5

Leren uit data

Hassan Haddouchi

Introductie

- Dit hoofdstuk behandelt hoe we nog meer inzichten halen uit enorm veel data.
- Het proces van data omzetten in kennis en actie.

Wat is het probleem?

Fraudedetectie bij een bank

- Banken verliezen jaarlijks miljoenen door fraude.
- Fraudegevallen kunnen variëren van gestolen creditcards tot valse transacties.
- **Vraag**: hoe kunnen we deze fraude sneller opsporen en voorkomen?

Leren uit data voor fraudedetectie

- Machine learning wordt gebruikt om verdachte patronen in data te herkennen.
- Voorbeelden:
  - Ongebruikelijke tijdstippen van transacties.
  - Grote uitgaven in korte tijd.
  - Locaties die niet overeenkomen met de klantgeschiedenis.
- Door data te analyseren, kunnen banken verdachte activiteiten automatisch signaleren.

Het proces: van data naar actie

1. **Data verzamelen**: transacties, klantgegevens, en historische fraudegevallen.
2. **Feature engineering**: patronen en kenmerken zoals transactiebedrag, tijdstip, en locatie.
3. **Model bouwen**:
    - **Supervised learning**: trainen met gelabelde gegevens (fraude/niet-fraude).
    - **Unsupervised learning**: opsporen van afwijkende patronen zonder labels.
4. **Model toepassen**: real-time monitoring van transacties.

Een eenvoudig voorbeeld in Python

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

# Transactiegegevens simuleren
import pandas as pd
data = pd.DataFrame({
    'bedrag': [100, 5000, 20, 1500],
    'tijdstip': [2, 23, 13, 4],
    'locatie_afwijkend': [0, 1, 0, 1],
    'fraude': [0, 1, 0, 1]
})

X = data[['bedrag', 'tijdstip', 'locatie_afwijkend']]
y = data['fraude']

# Trainen van een model
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=42)
model = RandomForestClassifier()
model.fit(X_train, y_train)

# Resultaten
predictions = model.predict(X_test)
print(classification_report(y_test, predictions))
```

Nog een voorbeeld: voorspellen van restaurantbeoordelingen

Stel je voor dat je een platform ontwikkelt voor restaurantreserveringen.
Gebruikers laten recensies en beoordelingen achter over hun ervaringen.

Probleem

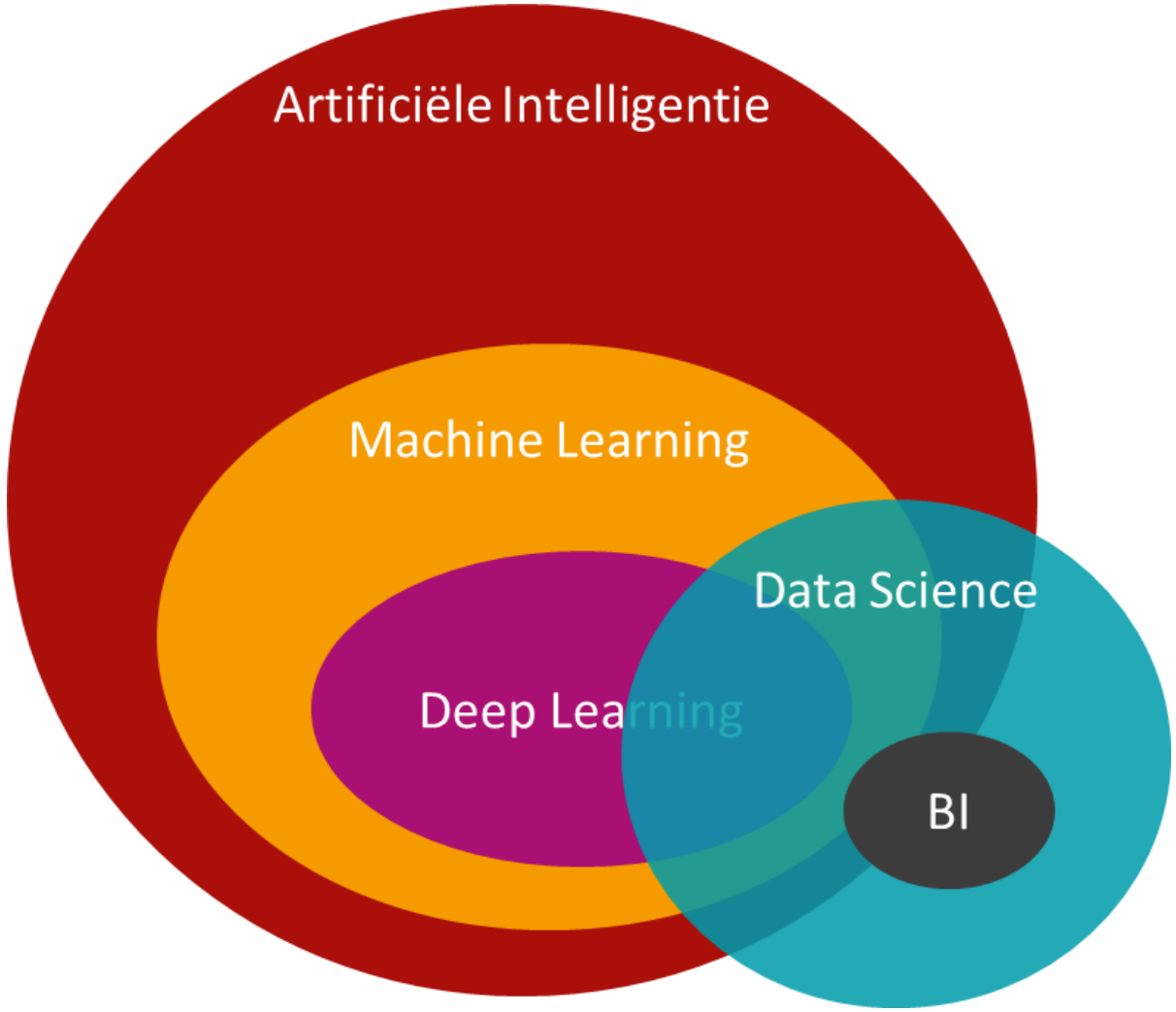
Hoe kun je voorspellen of een restaurant een hoge of lage beoordeling krijgt op basis van gegevens zoals:

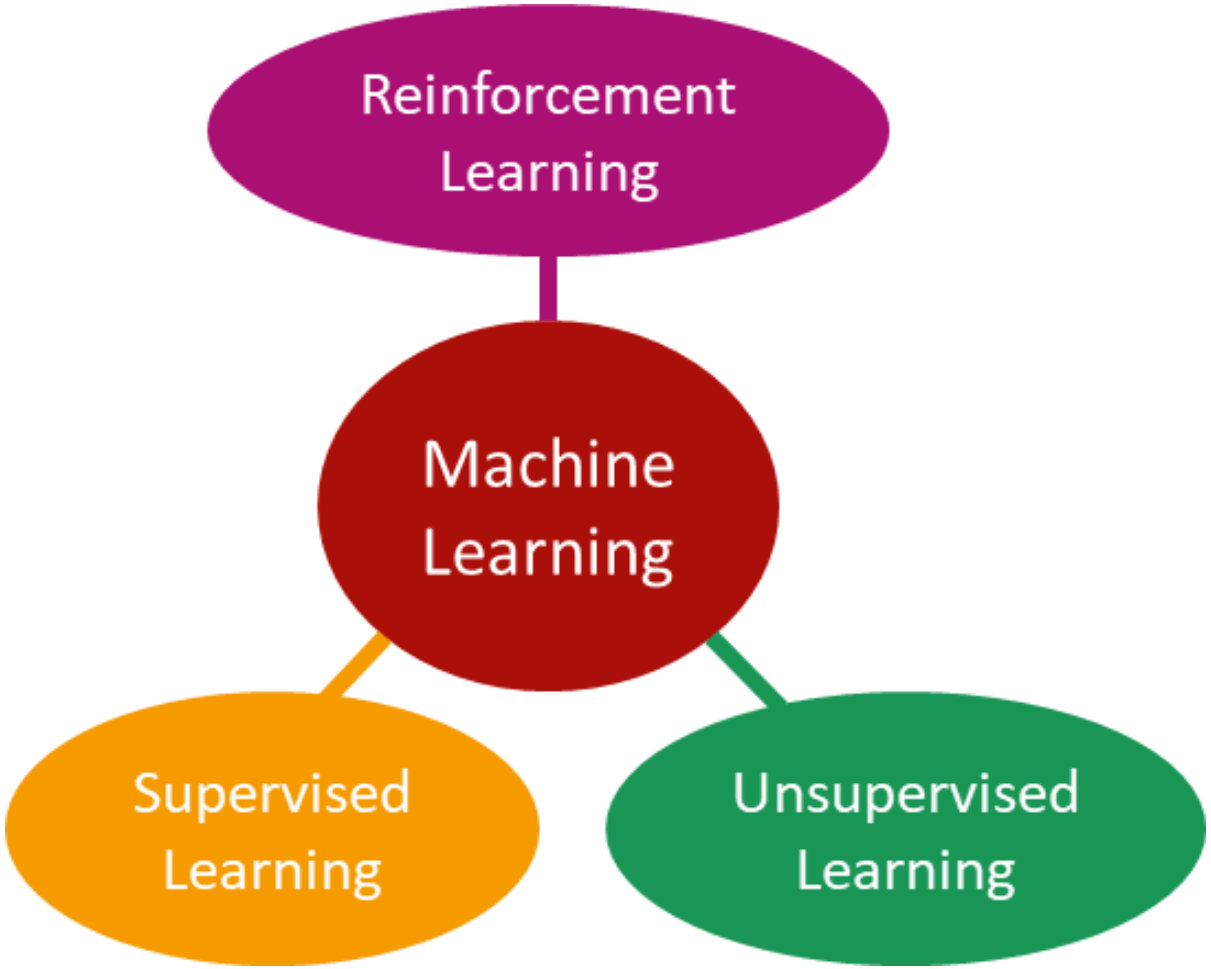
- Prijsklasse
- Bereidingstijd
- Aantal keren dat het restaurant is bezocht

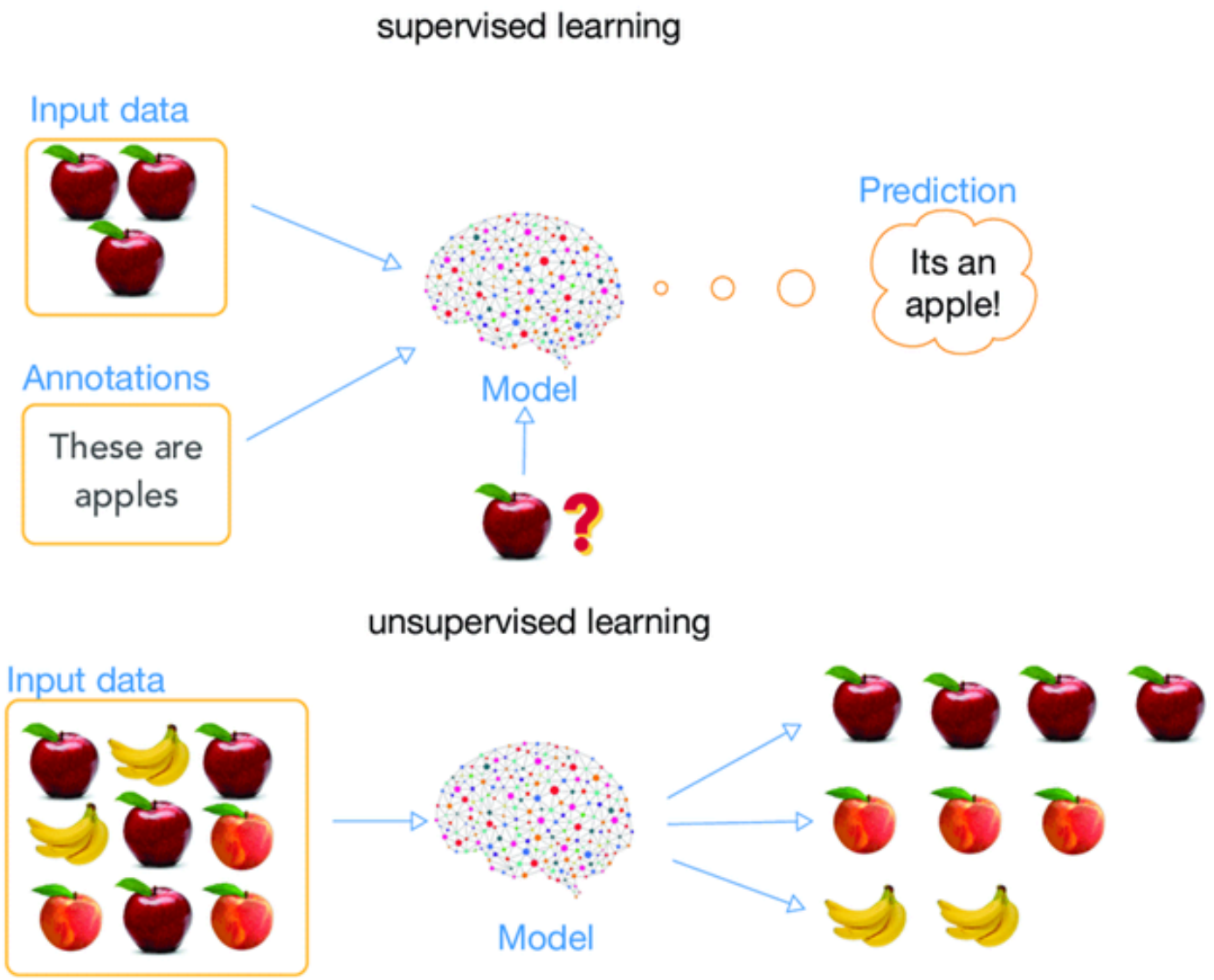
Oplossing

Leer uit data om recensies te analyseren en nieuwe beoordelingen te voorspellen.

Context





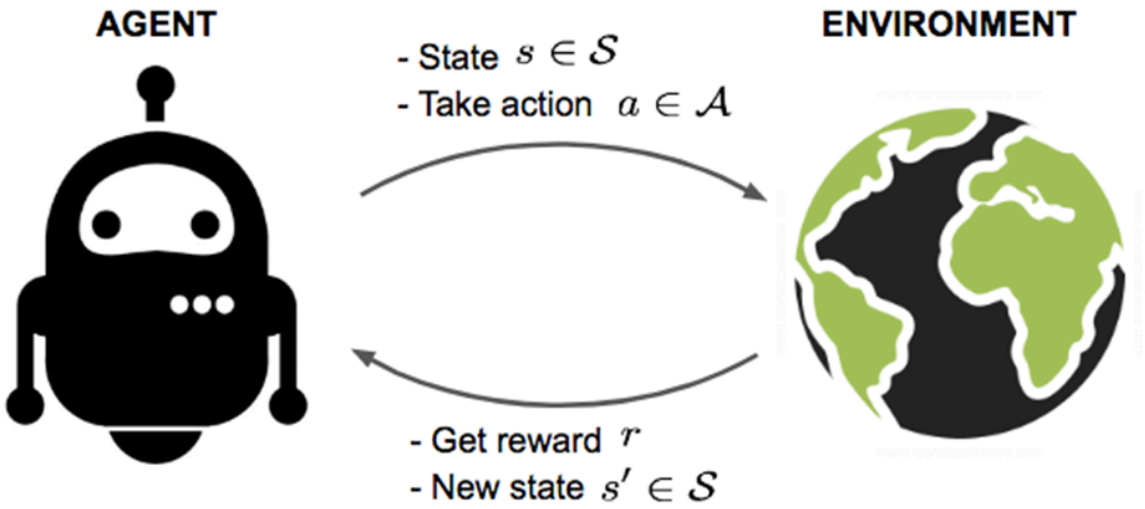


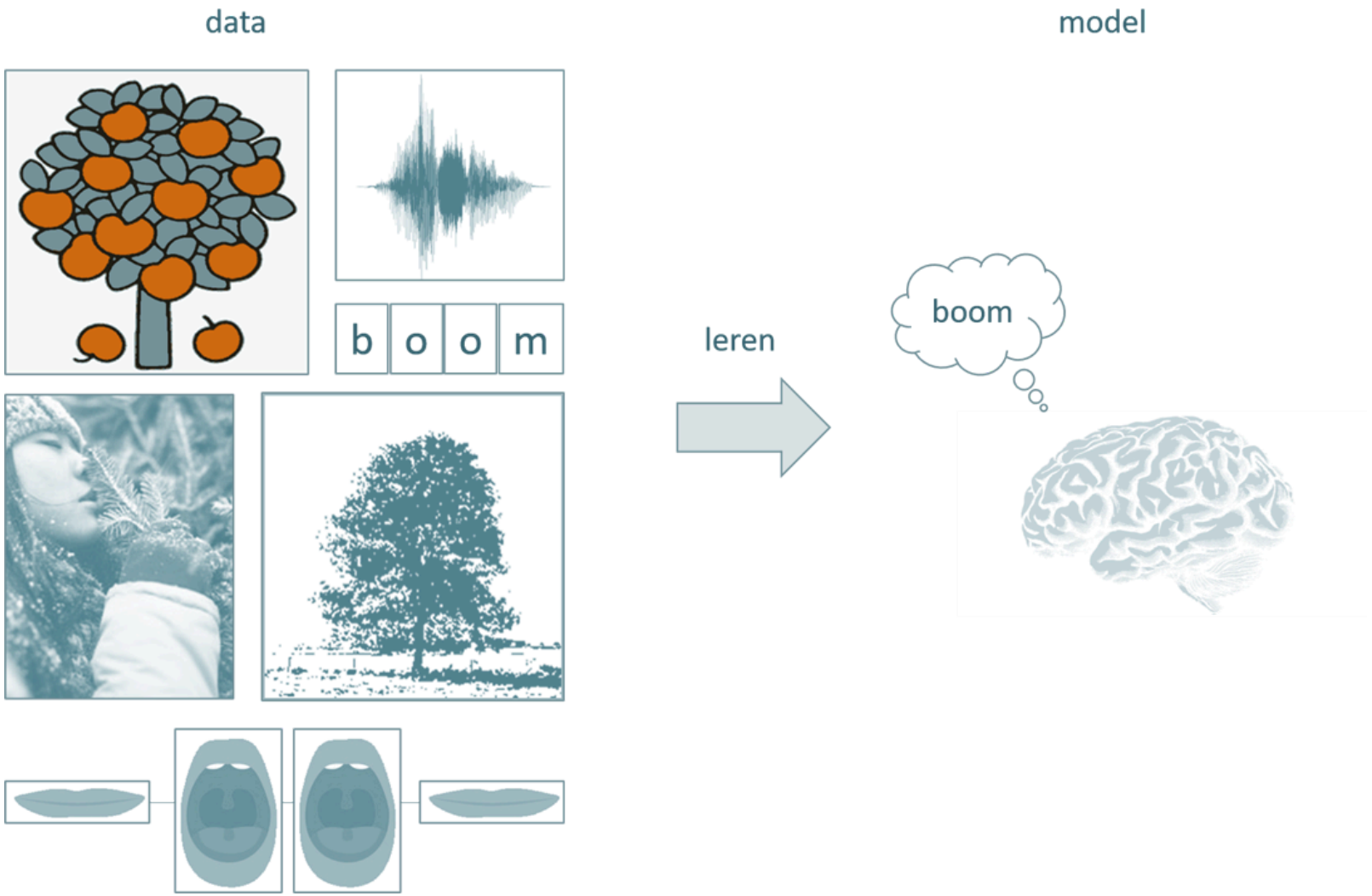
Supervised vs unsupervised learning

- Supervised learning (begeleid leren)
  Maakt gebruik van een dataset met labels. Het doel is om voor nieuwe data dit label te bepalen.
- Unsupervised learning (onbegeleid leren)
  Gebruikt data zonder labels om patronen te detecteren in de data.

Reinforcement learning (conditionering)

Een agent interageert met een omgeving en leert via trial and error door acties te ondernemen een beloning te maximaliseren.





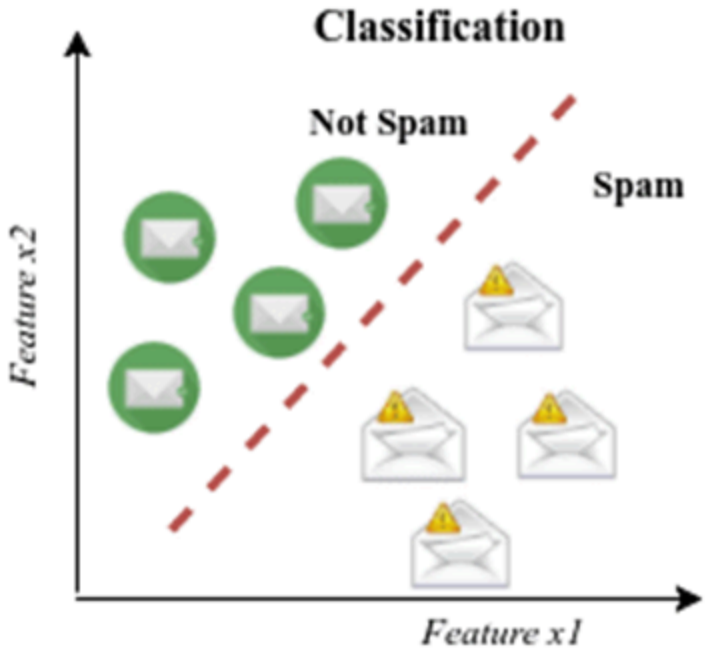
Classificatie vs regressie

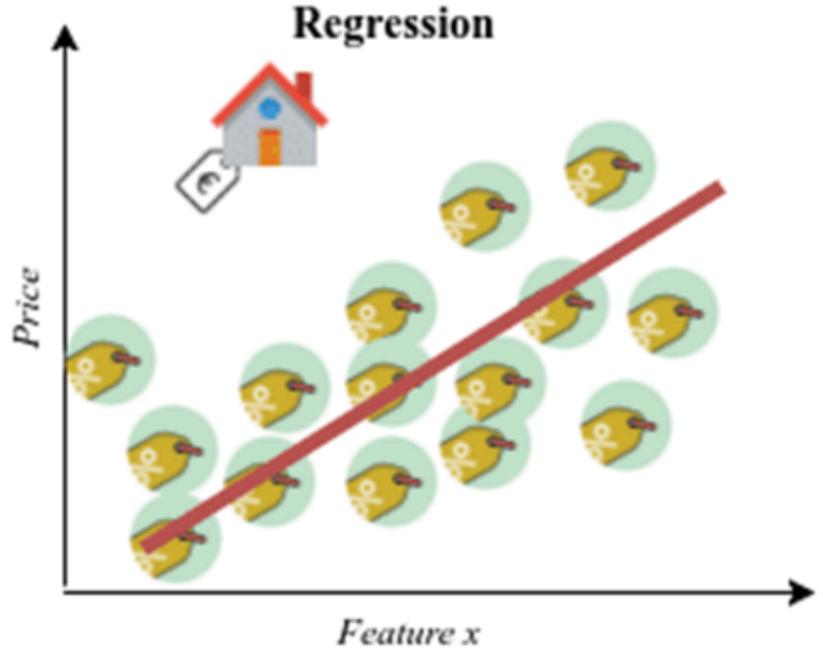
Classificatie:

Het doel is om de data op te splitsen in verschillende categorieën.

Regressie:

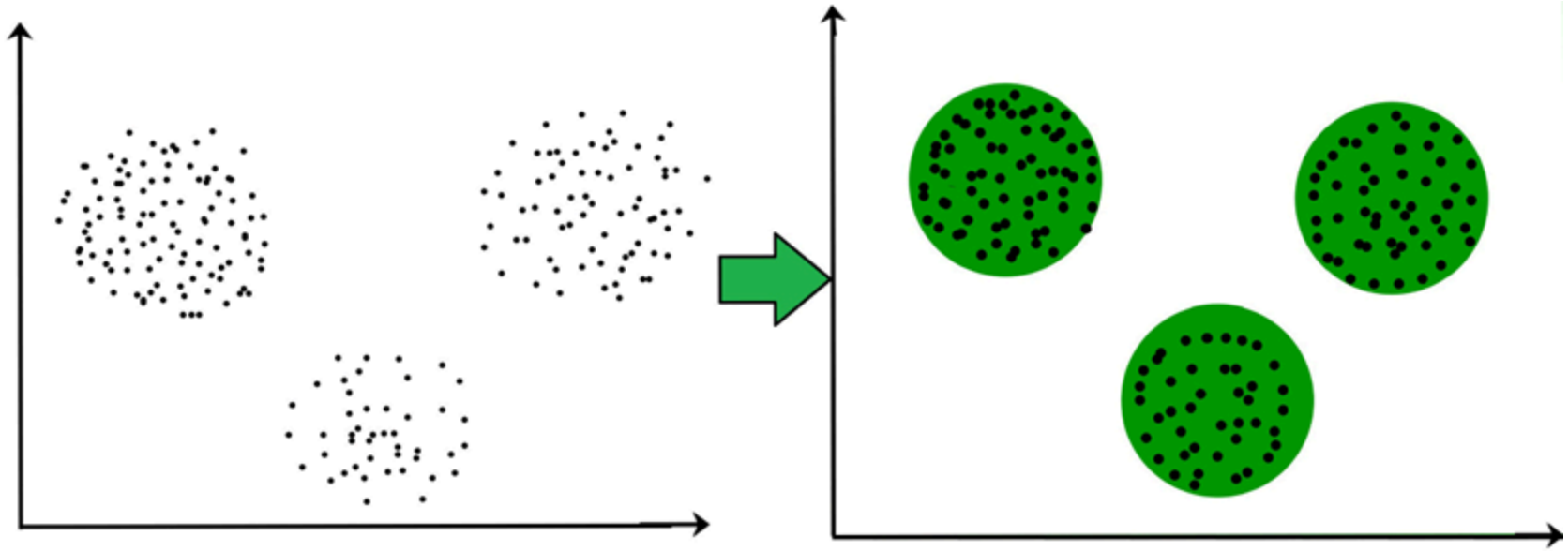
Hierbij wordt er een getal voorspeld.





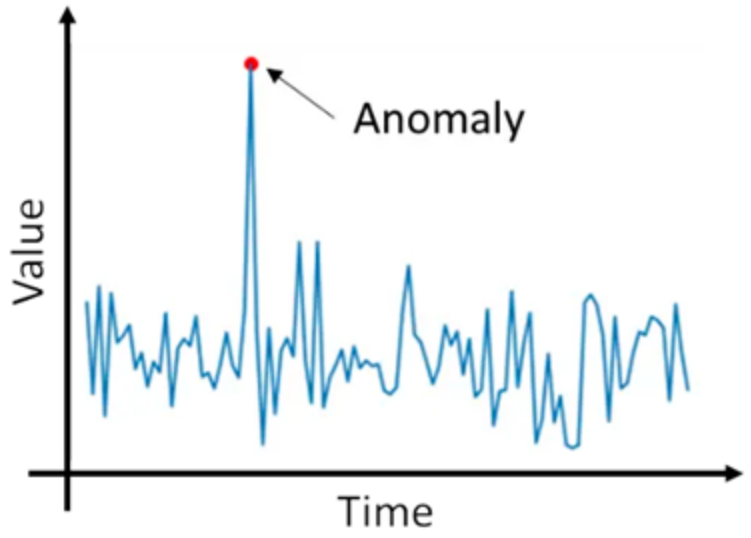
Clustering

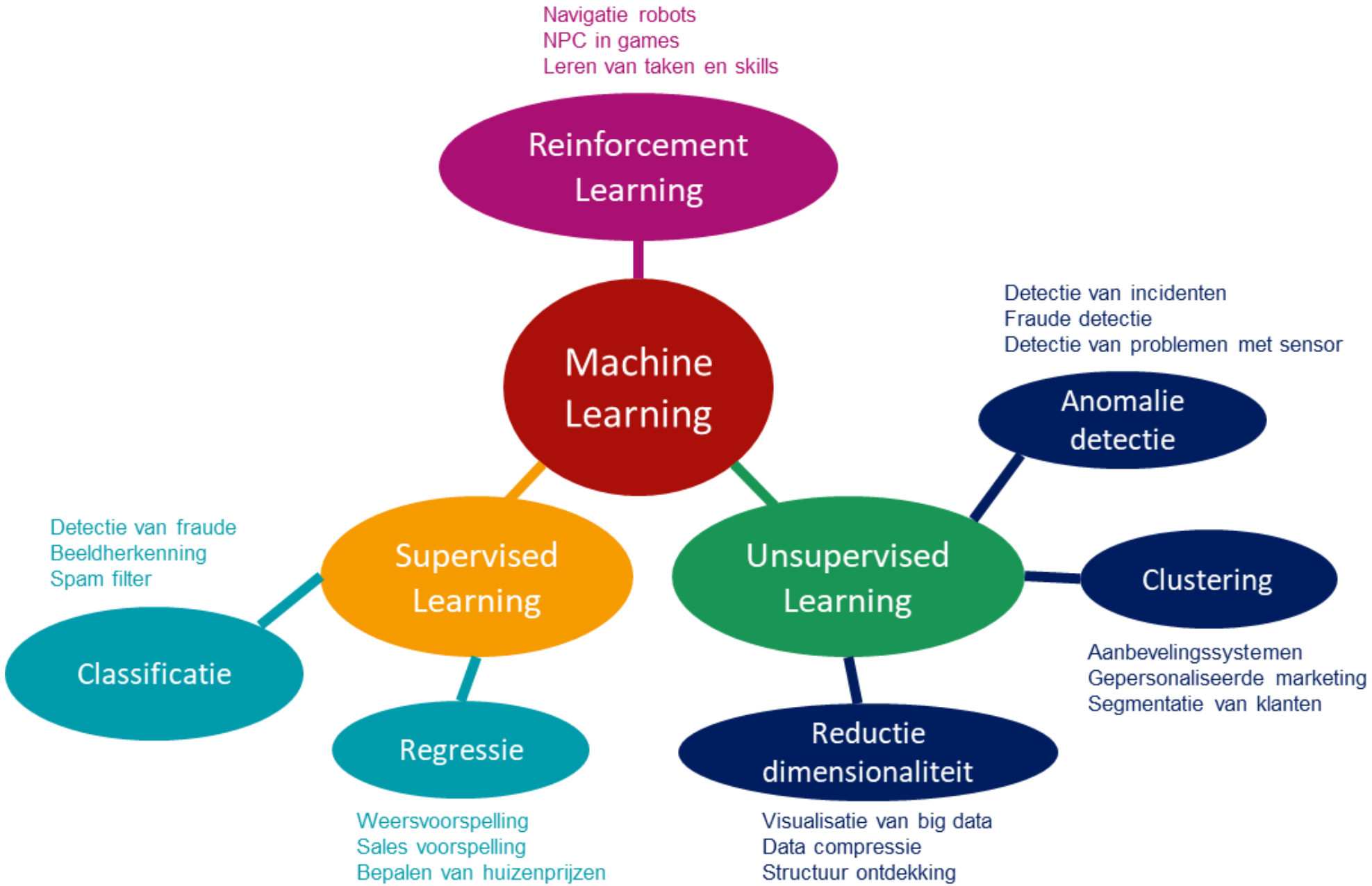
Het groeperen van objecten zodat objecten in dezelfde groep (of cluster) meer op mekaar lijken dan op objecten uit andere groepen.



Anomalie detectie

Het identificeren van uitzonderlijke items, gebeurtenissen of observaties die significant afwijken van de rest van de data en niet het gewone gedrag volgen.



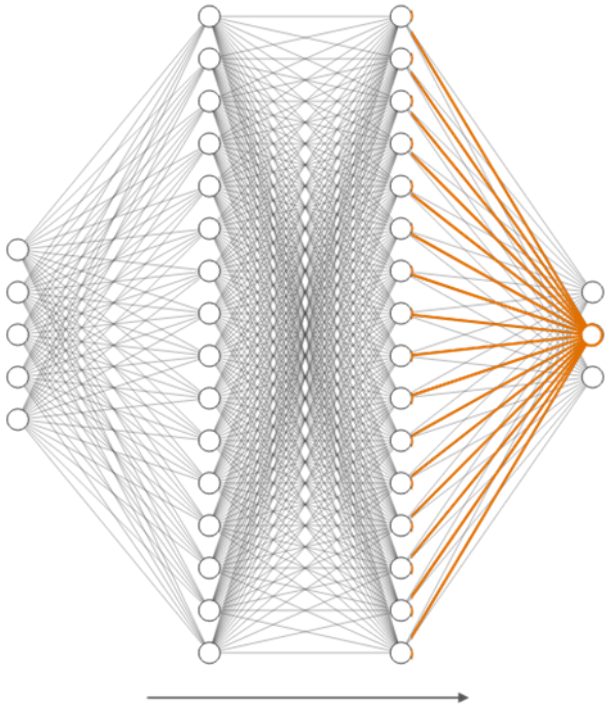


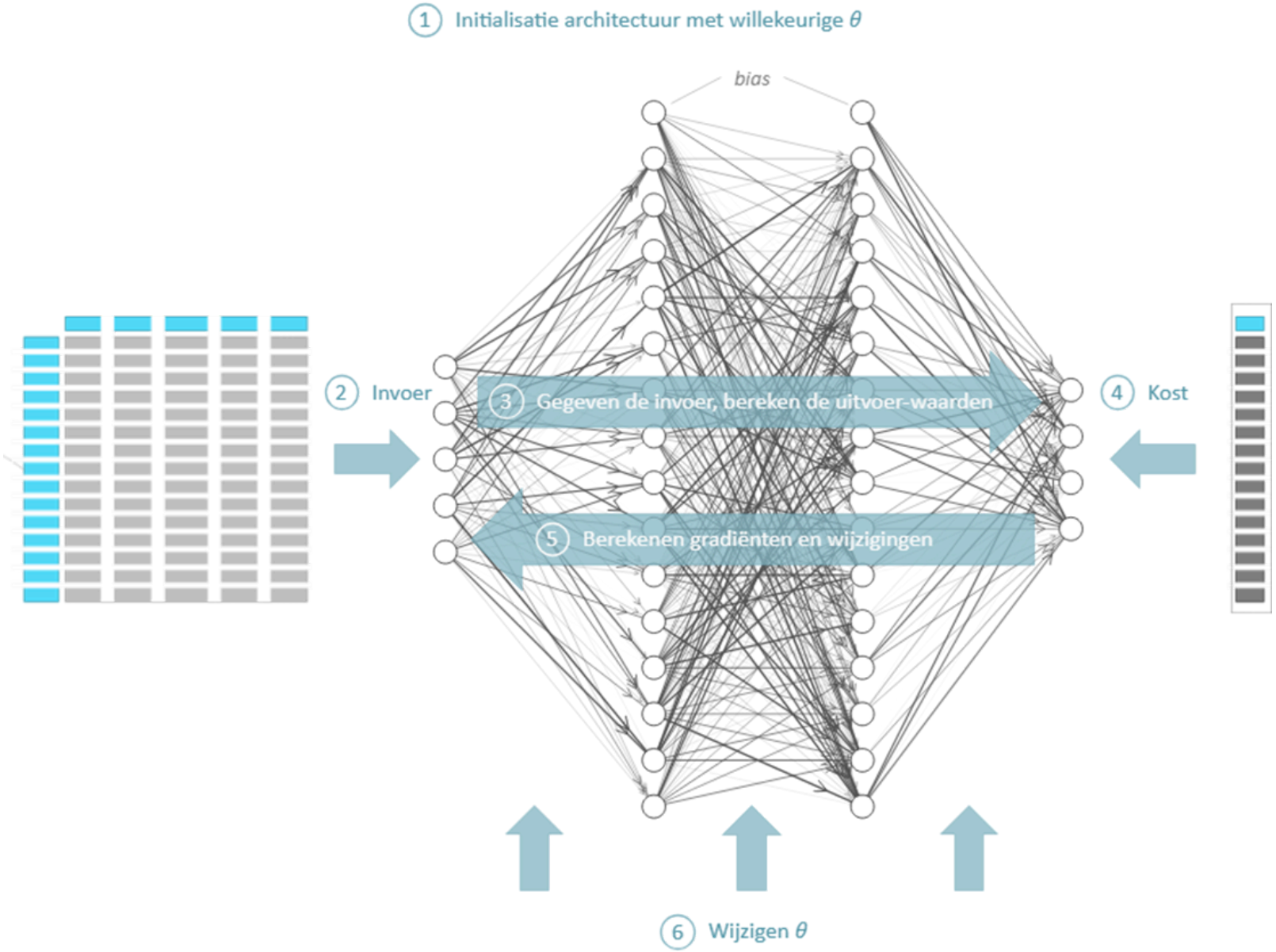
ML algoritmen

- Lineaire regressie (regressie)
  Verband tussen de variabelen x en labels y
  Het te voorspellen label wordt ook wel de afhankelijke variabele genoemd en de data die gebruikt wordt de onafhankelijke variabelen.
- Logistische regressie (classificatie)
  Geeft de waarschijnlijkheid dat een datapunt tot een klasse behoort.
- Beslissingsboom (classificatie en regressie)
  Een opeenvolging van ja/nee vragen die tot een voorspelling leiden.

Neuraal netwerk

Bestaan uit verschillende lagen (bij meerdere lagen spreekt men van deep learning). In elke node worden de output van de vorige laag samengebracht en gecombineerd.





Hoe evalueren we kennis uit data?

Overfitten

Het fenomeen wanneer je een goede evaluatie krijgt, maar het algoritme enkel goed werkt op de training data omdat het deze onthoudt.

Daarom moeten we onze dataset splitsen.

- Training set: om het algoritme te trainen.
- Test set: om de finale evaluatie van je algoritme te doen.
- Validatie set (optioneel): om te kijken hoe goed je algoritme werkt

Quiz

Met welke data werken neurale netwerken het best?

Wat is een typisch kenmerk van een neuraal netwerk?

Wat is een voorbeeld van anomalie detectie?

Hoe heet het trainen van een agent met trial en error?

Labo

Doel van het Labo

- Begrijpen hoe gegevens ons kunnen helpen voorspellingen te doen.
- Een eenvoudig voorspellingsmodel bouwen met Python.
- Ervaring opdoen met dataverwerking en het toepassen van ML-bibliotheken.

Dataset

- Simuleer een dataset met de volgende kolommen:
  - `prijs` : Prijsklasse van het restaurant (1 = goedkoop, 3 = duur).
  - `bereidingstijd` : Gemiddelde bereidingstijd in minuten.
  - `bezoeken` : Hoe vaak een klant het restaurant heeft bezocht.
  - `beoordeling` : Hoog (1) of Laag (0).

Stappen voor het Labo

1. **Dataset creëren**:
    - Genereer een dataset in Python met willekeurige gegevens.
    - Gebruik `pandas` om de gegevens te structureren.
2. **Gegevens verkennen**:
    - Bereken gemiddelden en visuele trends in de data.
    - Maak een histogram van beoordelingen.

3. **Model bouwen**:
    - Gebruik een eenvoudig algoritme zoals een beslissingsboom (`DecisionTreeClassifier` uit `scikit-learn`).
    - Train het model om hoge en lage beoordelingen te voorspellen.
4. **Voorspellingen doen**:
    - Test het model op nieuwe gegevens en beoordeel de nauwkeurigheid.
    - Geef de resultaten grafisch weer met `matplotlib`.