A logo of a company

Description automatically generated

Advanced Machine Learning

AI-Onderzoeksvraag

Bevindingen Samenvatting

27-10-2025

Bevindingen

Bevindingen Advanced ML

Bevindingen Advanced ML

27-10-2025

**Formaliteiten**

**Auteur:** Esmee Kraan

**Bedrijf:** AppBakkers   
**Opdrachtgever:** AppBakkers

Table of Contents

[1. Wat is er onderzocht 4](#_Toc212465851)

[2. Leerpunten 4](#_Toc212465852)

[2.1. Resultaten Eenvoudige Modellen 4](#_Toc212465853)

[2.2. Resultaten Complexere Modelen 5](#_Toc212465854)

[2.3. Resultaten Neurale Netwerken 5](#_Toc212465855)

[2.4. Resultaten Curse of Dimensionality en PCA 5](#_Toc212465856)

[2.5. Conclusie 5](#_Toc212465857)

[3. Toegevoegde waarde 6](#_Toc212465858)

[3.1. Weight Decay en Dropout 6](#_Toc212465859)

[3.2. Early Stopping 7](#_Toc212465860)

[3.3. PCA en Autoencoders 8](#_Toc212465861)

[3.4. Eenvoudige Architecturen 10](#_Toc212465862)

[4. Belangrijkste Inzichten 11](#_Toc212465863)

# Wat is er onderzocht

We onderzochten hoe verschillende machine learning modellen omgaan met de balans tussen eenvoud en complexiteit. Hiervoor gebruikten we de [Ames Housing dataset](https://www.kaggle.com/datasets/prevek18/ames-housing-dataset), een uitgebreide dataset met gegevens over woningen in Iowa. Deze dataset bevat kenmerken zoals woonoppervlakte, bouwjaar, aantal badkamers, garagegrootte en kwaliteitsbeoordelingen, maar ook meer specifieke informatie zoals de aanwezigheid van zwembaden of open haarden. De data bood een goede basis om verschillende modeltypen te vergelijken, omdat de variatie in type features ons toeliet te onderzoeken hoe goed modellen omgaan met zowel numerieke als categorische informatie.

Het doel van dit onderzoek was niet om huizenprijzen zo precies mogelijk te voorspellen, maar om te begrijpen hoe modellen overfitten en welke technieken helpen om beter te generaliseren naar nieuwe data. We wilden zien waar de grens ligt tussen een model dat leert van de onderliggende patronen en een model dat te veel leert van de ruis of uitzonderingen in de trainingsdata.

Het onderzoek bestond uit drie thema’s:

* **Bias–Variance Trade-off:** “*Hoe veranderd de prestatie van een model naarmate het complexer wordt?*”
* **Regularisatie en Early Stopping:** “*Hoe kunnen we overfitting beperken met extra beperkingen te leggen op het model of de training tijdig te stoppen?*”
* **Curse of Dimensionality en PCA:** “*Wat gebeurt er als we te veel irrevelante features toevoegen, en hoe kan PCA dit effect verminderen door ze te comprimeren?*”

Door deze drie onderdelen samen te testen konden we de invloed van modelkeuze, datacomplexiteit en regularisatie beter begrijpen. Dit levert kennis op die toepasbaar is bij elk project waarin we te maken hebben met complexe data of beperkte hoeveelheden trainingsmateriaal.

# Leerpunten

## Resultaten Eenvoudige Modellen

De experimenten laten zien dat eenvoudige modellen zoals Ridge en Lasso stabieler blijven bij beperkte of ruisachtige data. Ze leggen de belangrijkste trends vast zonder te veel te leren van kleine fluctuaties. Ridge, dat gebruikmaakt van L2-regularisatie, bleek het meest betrouwbaar. De regularisatieterm hield de modelgewichten klein, waardoor het model minder gevoelig werd voor ruis en beter generaliseerde naar nieuwe voorbeelden.

## Resultaten Complexere Modelen

Complexe modellen zoals Decision Trees en Neural Networks presteren goed op de trainingsdata, maar hun nauwkeurigheid zakt zodra ze nieuwe data zien. Ze hebben een lage bias, maar een hoge variantie. In ons experiment zagen we dat een Decision Tree met een te hoge diepte perfect scoorde op de trainingset maar slechter presteerde op de validatieset. Dit gedrag illustreert de bias–variance trade-off: meer complexiteit betekent vaak minder generalisatie.

## Resultaten Neurale Netwerken

Bij het neurale netwerk met twee verborgen lagen (MLP) zagen we dat early stopping hielp om overfitting te voorkomen. Zonder early stopping daalde de RMSE tot 0.21, maar steeg daarna weer zodra het model verder leerde op ruis. Door vroegtijdig te stoppen verbeterde de fout tot 0.19 RMSE. Dit bewijst dat een eenvoudig mechanisme zoals early stopping net zo belangrijk is als de architectuur zelf.

## Resultaten Curse of Dimensionality en PCA

Toen we extra ruisfeatures toevoegden aan de dataset, daalde de prestatie van alle modellen. Dit bevestigt de curse of dimensionality: meer dimensies maken de dataruimte complexer en moeilijker te leren. Modellen raken dan verdwaald in irrelevante patronen. PCA bood hier een oplossing. Door de features te reduceren tot een kleiner aantal hoofdcomponenten, verwijderden we redundante informatie zonder dat de modellen belangrijke signalen verloren.

## Conclusie

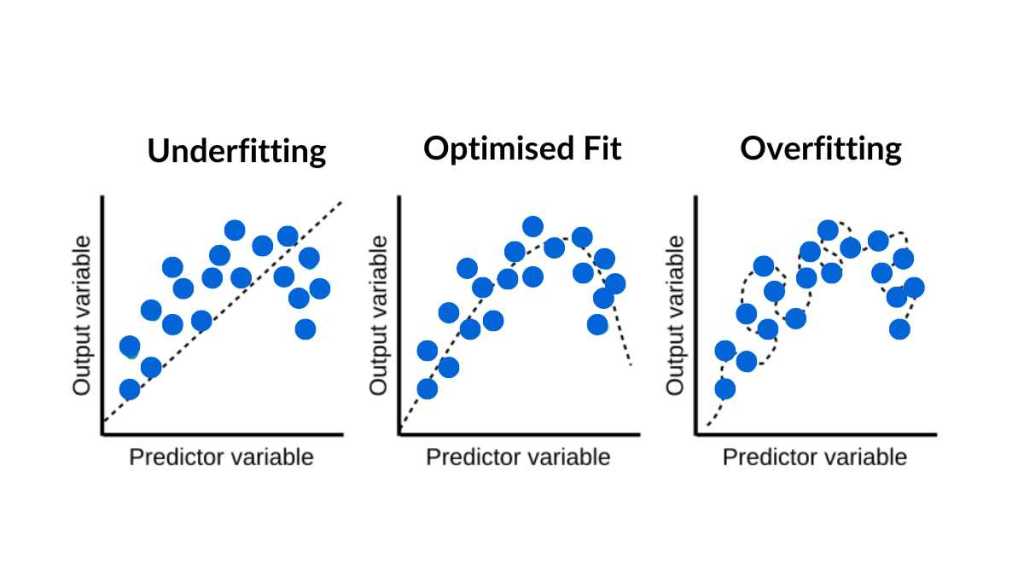
De belangrijkste les is dat meer features of complexere modellen niet automatisch leiden tot betere resultaten. Alleen informatieve, goed geschaalde en relevante features dragen bij aan generalisatie. Een gecontroleerde aanpak, met aandacht voor regularisatie en modelkeuze, levert uiteindelijk stabielere en betrouwbaardere prestaties op.

# Toegevoegde waarde

## Weight Decay en Dropout

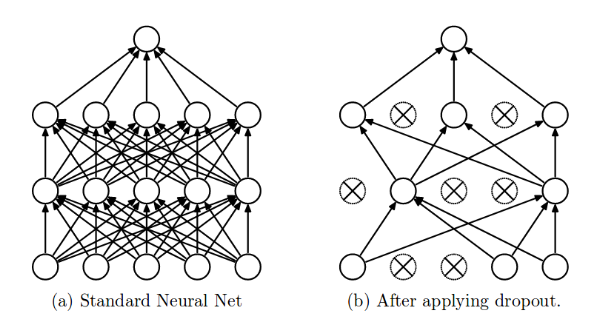
Een model leert patronen uit de data, maar als het te complex wordt, gaat het ook de ruis in de data onthouden. Dit noemen we overfitting. De eerste afbeelding hierboven laat dit duidelijk zien. Links zie je underfitting: het model is te simpel en mist belangrijke patronen. In het midden staat optimised fit: het model leert precies genoeg om de onderliggende structuur te vangen. Rechts zie je overfitting: het model volgt de ruis en uitzonderingen, waardoor het slecht presteert op nieuwe data.

Weight Decay en Dropout helpen om dat te voorkomen. Weight Decay (ook bekend als L2-regularisatie) zorgt ervoor dat de gewichten in een neuraal netwerk kleiner blijven. Het model wordt dan minder gevoelig voor kleine schommelingen in de data. Kleine gewichten dwingen het netwerk om generalere, stabielere patronen te leren in plaats van elk detail.



Figuur 1. Visualisatie van Weight Decay [1]

De tweede afbeelding laat zien hoe **Dropout** dit op een andere manier aanpakt. In het linkerdiagram zie je een standaard neuraal netwerk waar elke neuron met alle anderen is verbonden. In het rechterdiagram is dropout toegepast: willekeurige neuronen zijn tijdelijk “uitgeschakeld” tijdens de training. Hierdoor kan het model niet te afhankelijk worden van één specifieke route in het netwerk. Elke training leert het model met iets andere verbindingen, wat leidt tot robuustere representaties.

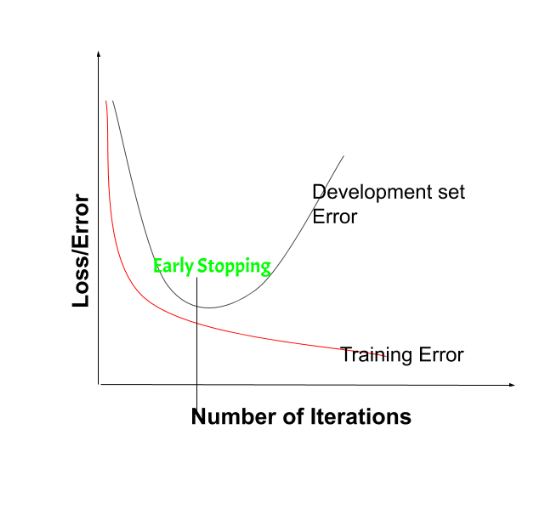


Figuur 2. Visualisatie van DropOut in een Standaard Neural Network [2]

Samen zorgen Weight Decay en Dropout ervoor dat het netwerk zich richt op de belangrijkste signalen in plaats van op ruis.

## Early Stopping

Figuur x hieronder laat zien waarom early stopping een belangrijk onderdeel kan zijn van het trainen van neurale netwerken. Op de horizontale as zie je het aantal trainingsiteraties, op de verticale as de fout (loss of error). De rode lijn toont de fout op de trainingsdata: die blijft dalen naarmate het model langer leert. De grijze lijn toont de fout op de validatieset, die eerst ook daalt, maar daarna weer stijgt. Dat stijgende stuk betekent dat het model de ruis uit de trainingsdata begint te leren in plaats van de echte patronen.



Figuur 3. Visualisatie van Early Stopping bij een Training Cyclus [3]

Early stopping voorkomt dat. Tijdens de training houden we de fout op de validatieset bij. Zodra die fout niet meer kleiner wordt, stoppen we het proces. In de grafiek is dat punt aangegeven in groen. Daar is het model het best afgestemd, het heeft genoeg geleerd om de data te begrijpen, maar nog niet zoveel dat het overfit.

In ons eigen experiment met het MLP-model zagen we dat dit principe werkt. Zonder early stopping zakte de fout (RMSE) naar 0.21, maar na langer trainen begon die weer te stijgen. Met early stopping eindigde de training op 0.19 RMSE.

## PCA en Autoencoders

Wanneer je werkt met veel variabelen, raak je snel informatie kwijt in de ruis. Niet elke kolom draagt iets unieks bij. Sommige meten hetzelfde of voegen variatie toe die geen betekenis heeft. Dimensionality reduction lost dat op door data samen te vatten tot een kleinere, informatierijke vorm.

De eerste afbeelding laat zien hoe PCA werkt. PCA zoekt de richting in de data met de meeste variatie. In dit voorbeeld zijn dat de assen voor Area en Radius. De eerste hoofdrichting, PC1, bevat de belangrijkste informatie. De tweede richting, PC2, voegt weinig toe. Door alleen PC1 te houden, wordt de data eenvoudiger zonder veel informatie te verliezen. PCA is wiskundig en direct te interpreteren.

Afbeelding met tekst, lijn, diagram, Lettertype

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 4. Een 2D Dataset Transformeren naar een 1D Representatie [5]

De tweede afbeelding toont een autoencoder. Dit is een neuraal netwerk dat zelf leert welke informatie belangrijk is. Het netwerk reconstrueert de invoer via een smalle laag in het midden, de bottleneck. Daar wordt de data samengeperst. Wat overblijft is een compacte representatie met de belangrijkste patronen. Afbeelding met cirkel, diagram, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 5. Voorbeeld Architectuur van een Autoencoder [4]

PCA werkt goed bij lineaire structuren. Autoencoders passen zich beter aan bij niet-lineaire data, zoals audio of medische signalen. Bij long- of hartgeluiden kan dit bijvoorbeeld helpen dit om ruis of herhalende frequenties te verwijderen. Zo leert het model stabieler en maakt het minder fouten bij nieuwe opnames.

## Eenvoudige Architecturen

In veel van onze projecten is het niet genoeg dat een model nauwkeurig voorspelt. We moeten ook begrijpen waarom het dat doet. Bij medische of audio toepassingen, zoals het HeartGuard en LungInsight project, is transparantie belangrijk. Artsen en onderzoekers moeten kunnen volgen hoe een model tot zijn conclusie komt.

Met een eenvoudige architectuur bedoelen we een model met beperkte diepte of minder parameters, zoals een klein CNN of een lineair regressiemodel. Zulke modellen zijn makkelijker te interpreteren omdat de relatie tussen invoer en uitvoer duidelijk blijft. Elke feature of filter heeft een zichtbaar effect op de voorspelling.

Bij medische signalen helpt dit om te begrijpen welke delen van het signaal bepalend zijn. Bijvoorbeeld welke frequentieband of golfvorm bijdraagt aan een diagnose. Dat inzicht maakt de resultaten bruikbaar in de praktijk. Complexe modellen zoals diepe neurale netwerken leveren vaak hogere nauwkeurigheid, maar verbergen hun beslissingen in duizenden parameters.

Een eenvoudiger model dwingt je om betere features te ontwerpen en biedt meer controle over de beslissingen. Dat vergroot het vertrouwen van domeinexperts in de output en maakt samenwerking tussen data scientists en specialisten effectiever.

# Belangrijkste Inzichten

Een te complex model leert te veel van de details in de trainingsdata en verliest zijn vermogen om nieuwe voorbeelden goed te voorspellen. Dit overfittingprobleem komt voor bij alle datatypes, van tabulaire data tot beeld en geluid. Regelmatige toepassing van technieken zoals regularisatie, early stopping en reductie van dimensies houdt modellen stabiel en voorkomt dat ze overleren.

Het verkleinen van de dataruimte met PCA of autoencoders helpt om informatieve patronen te isoleren en ruis te verwijderen. Hierdoor worden modellen efficiënter en presteren ze beter op nieuwe data. Regularisatie dwingt eenvoud af, wat zorgt voor betere generalisatie en minder afhankelijkheid van toeval in de data.

Eenvoudige modellen zijn niet per definitie zwakker. Ze hebben vaak minder variatie in hun voorspellingen en zijn beter uitlegbaar. In toepassingen waar betrouwbaarheid belangrijk is, zoals medische signalen of kritische geluidsclassificatie, biedt een eenvoudiger model meer controle en inzicht. Complexere modellen zijn nuttig bij grotere en rijkere datasets, maar vragen om striktere monitoring van hun generalisatie.

De belangrijkste les is dat generaliseerbaarheid altijd belangrijker is dan trainingsprestatie. Een model dat minder leert maar stabiel blijft op onbekende data is waardevoller dan een model dat alleen goed is op het trainingsmateriaal.

#### **Referentielijst**

[1] N. V. Otten and N. V. Otten, “Weight Decay In Machine Learning And Deep Learning Explained & How To Tutorial,” *Spot Intelligence*, May 02, 2024. https://spotintelligence.com/2024/05/02/weight-decay/

[2] H. Yadav, “Dropout in Neural Networks | Towards Data Science,” *Towards Data Science*, Jul. 05, 2022. https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9/

[3] GeeksforGeeks, “Regularization by Early Stopping,” GeeksforGeeks, Oct. 21, 2020. https://www.geeksforgeeks.org/machine-learning/regularization-by-early-stopping/

‌[4] GeeksforGeeks, “Autoencoders in Machine Learning,” GeeksforGeeks, Jun. 21, 2019. https://www.geeksforgeeks.org/machine-learning/auto-encoders/

[5] GeeksforGeeks, “Principal Component Analysis(PCA),” GeeksforGeeks, Jul. 07, 2018. https://www.geeksforgeeks.org/data-analysis/principal-component-analysis-pca/