



UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Studying the impact of new feature engineering methods

Batuhan Özdemir

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Koenraad VANHOOF



UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be

Universiteit Hasselt

Campus Hasselt:

Martelarenlaan 42 | 3500 Hasselt

Campus Diepenbeek:

Agoralaan Gebouw D | 3590 Diepenbeek

2024
2025



Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Studying the impact of new feature engineering methods

Batuhan Özdemir

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Koenraad VANHOOF

Woord vooraf

Deze masterthesis vormt het sluitstuk van mijn opleiding Handelsingenieur in de Beleidsinformatica aan de Universiteit Hasselt. Tijdens het schrijven van deze thesis was het voor mij niet enkel een opdracht, maar ook een leertraject waarin ik mijn kennis op vlak van data-analyse, machine learning en kritisch denken verder heb ontwikkeld.

Gaandeweg heb ik geleerd om zelfstandig problemen aan te pakken, beslissingen te verantwoorden en mijn aanpak voortdurend te verbeteren om aan de verwachtingen van mijn promotor te voldoen. Het proces bracht dus enkele uitdagingen met zich mee, van het communiceren tot het begrijpen en toepassen van de algoritmen. Uit die leerervaring heb ik ontdekt dat ik in staat ben om zelf naar mogelijke verklaringen te zoeken.

Bij deze gaat mijn dank uit naar mijn promotor, Prof. Dr. Koenraad Vanhoof, voor zijn begeleiding, inhoudelijke expertise en constructieve feedback. Hij heeft mij ondersteund en in de juiste richting gestuurd tijdens mijn onderzoek om het resultaat te behalen.

Batuhan Özdemir
Genk, juni 2025

Inhoudsopgave

Woord vooraf	1
1 Introductie	3
2 Onderzoeksvragen	4
3 Gerelateerd werk	5
3.1 Machine learning	5
3.2 Traditionele methoden	5
3.3 Feature engineering	7
3.4 Toepassingen	8
4 Methodologie	9
4.1 Datasets	9
4.2 Programma's en programmeertalen	10
4.3 Methoden	10
5 Resultaten experimenten	14
5.1 Iris	15
5.1.1 Unsupervised K-Means	15
5.1.2 Supervised K-Means	20
5.1.3 KNN	25
5.1.4 KNN toegepast op de supervised K-Means data	26
5.1.5 Beslissingsbomen	28
5.1.6 Beslissingsbomen toegepast op de supervised K-Means data	31
5.2 USArrests	34
5.2.1 Unsupervised K-Means	34
5.2.2 Supervised K-Means	39
5.2.3 KNN	43
5.2.4 KNN toegepast op supervised K-Means	44
5.2.5 Beslissingsbomen	45
5.2.6 Beslissingsbomen toegepast op supervised K-Means	47
5.3 Wine	49
5.3.1 Unsupervised K-Means	49
5.3.2 Supervised K-Means	55
5.3.3 KNN	59
5.3.4 KNN toegepast op supervised K-Means	61
5.3.5 Beslissingsbomen	62
5.3.6 Beslissingsbomen toegepast op supervised K-Means	64
5.4 Titanic	67
5.4.1 Unsupervised K-Means	67
5.4.2 Supervised K-Means	72
5.4.3 KNN	77
5.4.4 KNN toegepast op de supervised K-Means	79
5.4.5 Beslissingsbomen	80
5.4.6 Beslissingsbomen toegepast op de supervised K-Means	83
6 Discussie	86

1 Introductie

In machine learning speelt *feature engineering* een belangrijke rol in het verbeteren van de prestaties van verschillende machine learningmodellen (Lee, 2019; Zhou, 2021; Jo, 2021; Bonilla, 2023; Trost et al., 2025). In Zhou (2021) worden belangrijke features uit datasets getransformeerd, geselecteerd of gecreëerd en toegepast in zowel supervised als unsupervised modellen om beter te generaliseren of nauwkeurigere voorspellingen te maken van de klassen. De wijze waarop data wordt voorgesteld, speelt zowel een rol binnen de supervised als unsupervised learning (Jo, 2021).

Er is reeds literatuur beschikbaar over onderzoek dat is gedaan naar *feature engineering* methoden waarbij de prestaties van de machine learning-algoritmen worden geoptimaliseerd (Jolly, 2018; Lee, 2019; Zhang, 2021; Jo, 2021; Zhou, 2021; Bonilla, 2023; Paghdal, 2025; Trost et al., 2025). De meest gehanteerde technieken zijn normalisatie en standaardisatie (Jolly, 2018). Beide technieken zorgen ervoor dat de numerieke variabelen op een gelijke schaal liggen en dat er vervolgens een correcte afstand berekend wordt (Jolly, 2018; Zhou, 2021). Daarnaast worden methoden gebruikt om continue of numerieke variabelen om te vormen naar categorische variabelen (Paghdal, 2025). Die methoden zijn onder meer discretisatie of binning, en worden voornamelijk toegepast bij beslissingsbomen (Paghdal, 2025). Vervolgens bestaat er een methode, namelijk featureselectie, waarbij de variabelen die de meeste informatie bevatten worden geselecteerd aan de hand van ingebouwde pythonfuncties, zoals *SelectKBest* (Zhang, 2021; Bonilla, 2023; Trost et al., 2025). Het doel van die methode is om overfitting te reduceren en de modellen beter interpreteerbaar te maken (Trost et al., 2025). Er is sprake van overfitting als het model te sterk op de trainingsdata afstemt, waardoor het slecht op nieuwe data, namelijk de testdata, presteert (Han & Kamber, 2006). Ten slotte worden nieuwe variabelen aangemaakt door combinaties uit bestaande data of afleidingen te maken (Bonilla, 2023; Paghdal, 2025; Trost et al., 2025). Dat principe wordt featurecreatie genoemd.

Hoewel er een breed overzicht aan literatuur is van verschillende *feature engineering* methoden, blijft er een kloof tussen nieuwe en originele methoden die de positie van de bestaande datapunten transformeren en vervolgens in machine learning-algoritmen opnemen. In deze masterproef wordt onderzocht hoe de preprocessingtechniek, namelijk het verplaatsen van datapunten naar hun dichtstbijzijnde centroiden, de prestatie van de diverse machine learning-algoritmen beïnvloedt. De centroiden van deze techniek geven het midden van de clusters weer. De preprocessingtechniek is een nieuw en origineel idee waar geen literatuuronderzoek voor bestaat. De positie van de datapunten wordt aangepast, terwijl bij de oude methoden enkel de structuur werd aangepast.

De focus van dit onderzoek ligt op het evalueren van die verplaatsing van de datapunten naar hun dichtstbijzijnde centroiden, de preprocessingstap, binnen de drie meest gebruikte, traditionele algoritmen, namelijk K-Means, K-Nearest Neighbors (KNN) en beslissingsbomen (Rokach and Maimon, 2005; Everitt et al., 2011; Aggarwal and Reddy, 2013; Deng et al., 2016; Xing and Bei, 2020). Een verdere toelichting voor de keuze van deze machine learning-algoritmen wordt in Sectie 3 en Sectie 4.3 beschreven. In dit onderzoek worden de methoden op een gekende alsook een supervised manier toegepast. Zo wordt in de gekende aanpak de K-Means unsupervised clustering toegepast, waarbij een vooraf bepaald aantal clusters willekeurig clusters gaat maken van de volledige dataset. Vervolgens worden de datapunten verplaatst naar de centroiden binnen de clusters en worden de andere twee classificatiealgoritmen op die verplaatsingen toegepast en geëvalueerd. Na elke verplaatsing worden de prestaties van de modellen berekend en vergeleken met de voorafgaande stappen en de beginsituatie, waar geen verplaatsing plaatsvindt.

Bij de supervised aanpak wordt er supervised K-Means uitgevoerd. Dit houdt in dat er eerst clusters worden gevormd op basis van de klassewaarden, gebruikmakend van de traindata die verkregen wordt nadat de train-test-split met een 80/20-verhouding, respectievelijk train en test, wordt uitgevoerd. Nadat er per klasse wordt geclusterd, wordt de preprocessingtechniek daarop toegepast. Op basis van de supervised K-Means clusteringmethode worden zowel de KNN- als beslissingsbomenclassificatiemethoden daarop toegepast. Om de accuraatheid van de classificatie-algoritmen na te gaan, worden de train- en testaccuracies bekeken.

Deze thesis is als volgt opgebouwd: Sectie 2 geeft een overzicht van de onderzoeksvragen weer. In Sectie 3 wordt er een korte literatuurstudie gegeven over gerelateerde werken. Die literatuurstudie zorgt ervoor dat enkele begrippen worden verduidelijkt, de keuzes van de machine learning-algoritmen worden verantwoord en ten slotte alternatieve *feature engineering* methoden worden beschreven. In Sectie 4 wordt de toegepaste methode beschreven. Hierbij worden de datasets, de tools en de stappen van de experimenten uitgelegd. In Sectie 5 worden de resultaten besproken, door enkele waarden en visualisaties te interpreteren. Ten slotte wordt er in Sectie 6 een discussie geschreven, waarin de resultaten worden geanalyseerd en vergeleken met alternatieve methoden. Bijkomend wordt kritiek gegeven op de methode en wordt toekomstig onderzoek aangeraden.

2 Onderzoeksvragen

Om het onderzoek en de analyse te begeleiden, zijn er verschillende onderzoeksvragen opgesteld. Gegeven is een overzicht van de onderzoeksvragen die voldoen aan de eindvereisten:

- ***Wat is het effect van het verschuiven van datapunten naar hun dichtstbijzijnde centroiden?***
 - Door gebruik te maken van supervised en unsupervised methoden worden datapunten door de preprocessingtechniek verplaatst naar hun centroiden. Na elke verplaatsing worden de prestaties van de modellen berekend of gevisualiseerd.
- ***Is het effect verschillend tussen supervised en unsupervised methoden?***
 - De K-Means, KNN- en beslissingsboommethoden worden zowel op een supervised als op een unsupervised manier uitgevoerd. Voor de supervised wordt er verder gewerkt met de 80% traindata en voor de unsupervised wordt de gehele dataset gebruikt. Beide aanpakken worden vervolgens met elkaar vergeleken, om de gelijkenissen of verschillen te achterhalen.
- ***Hoe verschilt deze nieuwe methode met de bestaande feature engineering methoden die beschreven zijn in de literatuur?***
 - De resultaten van deze methoden worden vergeleken met de andere *feature engineering* methoden, namelijk discretisatie of binning, normalisatie en standaardisatie, feature selection en feature creation. Bijkomend worden ook andere machine learning algoritmen vergeleken met de klassieke methoden die in dit onderzoek worden toegepast, zoals Support Vector Machines (SVM).
- ***Indien er geen significante effecten optreden: Wat kan er beter en waar ging het eventueel mis?***
 - Bij deze vraag wordt uitgelegd waarom de experimenten geen resultaten opleverden of wat er beter gedaan kan worden.

3 Gerelateerd werk

In deze sectie worden enkele begrippen verduidelijkt aan de hand van bestaande literatuur. Ook wordt de keuze van de traditionele machine learning-methoden beschreven, met telkens de voor- en nadelen van de methoden. Ten slotte worden enkele *feature engineering*-technieken uit studies beschreven met de geleverde prestaties.

3.1 Machine learning

Machine learning is een tak binnen artificial intelligence (AI) waarbij modellen worden gebouwd die automatisch gaan leren uit de data die beschikbaar is, zonder vooraf geprogrammeerd te worden (Deng et al., 2016; Singh, 2019). Het bouwen van die algoritmen dient om patronen te detecteren of om voorspellingen te optimaliseren (Lee, 2019). Volgens Singh (2019) zijn zowel ongestructureerde als gestructureerde datasets waardevol om machine learning-modellen op te bouwen, voornamelijk voor clustering, classificatie en regressie.

Om de modellen te trainen, bestaan er verschillende aanpakken. De meest toegepaste zijn supervised en unsupervised learning (Zhou, 2021; Jo, 2021). Bij supervised learning leert het model door de traintdata, een subset van de dataset waar de features en de labels, die de klassenwaarden voorstellen, aanwezig zijn, te evalueren op een aparte subset, namelijk de testdata, om na te gaan hoe nauwkeurig de voorspellingen worden gemaakt op ongeziene data (Lee, 2019; Singh, 2019; Zhou, 2021).

Een alternatieve aanpak is de unsupervised learning. Hier worden de labels, de klassenwaarden, achterwege gelaten (Jo, 2021; El Morr et al., 2022). In plaats van een voorspelling te maken en vervolgens deze af te toetsen om de nauwkeurigheden te bepalen, worden hier de inputdata gebruikt om structuren, patronen of groeperingen te ontdekken op basis van gelijkenissen (Zhou, 2021; El Morr et al., 2022). Deze aanpak wordt vaak gebruikt in situaties waarbij de datalabels niet beschikbaar of te verkrijgen zijn (Aggarwal & Reddy, 2013). De meest toegepaste vorm van unsupervised learning is clustering (Everitt et al., 2011; Aggarwal and Reddy, 2013). Hieronder worden de methoden K-Means en Density-Based Spatial Clustering of Applications with Noise (DBSCAN) geplaatst. In de volgende sectie, Sectie 3.2, wordt hierop ingegaan.

Naast de supervised en unsupervised learning, bestaat ook de reinforcement learning (El Morr et al., 2022). Dit is een leermethode waarbij interactie vereist is met de omgeving om via *trial-and-error* het gedrag te optimaliseren (Zhou, 2021; El Morr et al., 2022). Na elke interactie wordt een beloning of straf gegeven, afhankelijk van een positief of negatief resultaat. Die beloningen of straffen zorgen voor verdere optimalisaties (Zhou, 2021). Deze aanpak wordt voor dit onderzoek buiten beschouwing gelaten, en ligt de focus dus eerder op de supervised en unsupervised methoden.

3.2 Traditionele methoden

Binnen machine learning zijn heel wat verschillende algoritmen ontwikkeld met elk hun eigen voor- en nadelen. In dit onderzoek wordt de nadruk gelegd op de traditionele methoden die representatief zijn voor zowel de supervised als unsupervised learning aanpak. De vijf meest gebruikte algoritmen zijn: K-Means, DBSCAN, KNN, beslissingsbomen en Support Vector Machines (SVM) (Everitt et al., 2011; Aggarwal and Reddy, 2013; Rodriguez et al., 2019; Xu and Tian, 2015; Zhou, 2021).

Unsupervised methoden

Startend met de unsupervised algoritmen worden K-Means en DBSCAN het vaakst gebruikt (Aggarwal and Reddy, 2013; Rodriguez et al., 2019; Chakraborty et al., 2020; Jo, 2021). Ten eerste is K-Means een methode die gebruikt wordt om data in k -groepen, ook wel clusters genoemd, te verdelen (Everitt et al., 2011; Jo, 2021). Om K-Means toe te passen is een vooraf bepaald clusteraantal vereist. Het doel van dit algoritme is om clusters te vormen op basis van onderlinge gelijkenissen in afstandsmetingen, zoals de Euclidische afstand (Everitt et al., 2011). De Euclidische afstand geeft de afstand tussen twee punten, namelijk een datapunt en de centroïde, in de ruimte weer (Everitt et al., 2011). Het K-Means-algoritme brengt voordelen met zich mee, namelijk dat het eenvoudig te implementeren is (Xu & Tian, 2015). Ook is de methode zeer transparant (Xu and Tian, 2015; Jolly, 2018). Verder is het schaalbaar naar grote datasets en is het algoritme efficiënter en sneller dan complexere algoritmen (Everitt et al., 2011; Jolly, 2018; Jo, 2021). Naast de voordelen brengt het algoritme ook nadelen met zich mee. Allereerst moet, zoals eerder vermeld, het aantal clusters (k) vooraf bepaald worden, wat de uiteindelijke resultaten kan beïnvloeden (Everitt et al., 2011; Jo, 2021; El Morr et al., 2022). Daarnaast is de methode gevoelig voor outliers, waardoor onnauwkeurige clustercentra worden berekend (Xu & Tian, 2015).

Een alternatieve methode die de beperkingen van de K-Means aanvult, is de DBSCAN-methode (Schubert et al., 2017). De DBSCAN-methode behoort tot de dichtheidsgebaseerde clusteringmethoden, waarbij de datapunten gegroepeerd worden op basis van hun lokale dichtheid (Everitt et al., 2011; Xu and Tian, 2015; Rodriguez et al., 2019). Anders dan de K-Means-methode vereist DBSCAN geen vooraf bepaald aantal clusters (k). Hoewel de prestaties sterk afhankelijk zijn van de parameterskeuze: epsilon (ϵ) en MinPts. Epsilon (ϵ) geeft de maximale afstand weer waarbinnen de punten als burenen worden beschouwd. De MinPts geeft het minimum aantal burenen dat een punt moet hebben om als kernpunt gezien te worden weer (Everitt et al., 2011; Xu and Tian, 2015; Schubert et al., 2017). Verder maakt het algoritme onderscheid tussen kernpunten, randpunten en ruis. Bijkomend is dit algoritme geschikter voor onregelmatige datasets en datasets die outliers bevatten (Everitt et al., 2011; Aggarwal and Reddy, 2013; Xu and Tian, 2015).

Supervised methoden

Voor de supervised methoden behoren KNN, beslissingsbomen en SVM tot de meest gebruikte technieken. Die methoden worden gebruikt voor classificatie- of regressietaken en zijn zeer eenvoudig toepasbaar (Zhou, 2021; El Morr et al., 2022). Ten eerste wordt het classificatie-algoritme KNN gebruikt om de klassen te bepalen van een nieuw datapunt op basis van de klassenwaarden van de k dichtstbijzijnde burenen (Jolly, 2018; Xing and Bei, 2020; Zhou, 2021). Het KNN-algoritme is eenvoudig en toepasbaar op verschillende datasets. Hoewel een te grote dataset ervoor kan zorgen dat het algoritme traag is. Vervolgens is het algoritme gevoelig voor outliers, want de klassenwaarden worden bepaald op basis van de burenen en als die afstand dus groter wordt, kunnen de resultaten beïnvloed worden (Deng et al., 2016; Xing and Bei, 2020; Zhou, 2021; El Morr et al., 2022). Ten slotte voegt Zhou (2021) er nog aan toe dat de keuze van het aantal burenen belangrijk is. Enerzijds kan een te lage k leiden tot overfitting. Anderzijds leidt een te hoge k tot underfitting (Zhou, 2021; El Morr et al., 2022).

Ten tweede worden beslissingsbomen opgesteld die dienen als een classificatie-algoritme. Hier gaat het model de data opdelen via een reeks beslissingen die worden genomen op basis van enkele condities (Rokach and Maimon, 2005; Vasconcellos et al., 2011). De condities worden visueel weergegeven in de bomen door de knooppunten in de splitsingen. De beslissing wordt gemaakt op basis van één attribuut (Rokach and Maimon, 2005; Lu and Braunstein, 2014). Een boom kan meerdere splitsingen

bevatten. Hoewel het aantal splitsingen vaak kan leiden tot slechte interpretatie door de instabiele structuren (Lu and Braunstein, 2014; Kuo et al., 2021; Zhou, 2021). De bomen worden volgens een hiërarchie opgebouwd, waarin data stapsgewijs wordt gesplitst tot een eindklasse ontstaat (Kuo et al., 2021). Er bestaan verschillende soorten boomalgoritmen, namelijk CART (*Classification and Regression Trees*), ID3 en C4.5 (Rokach and Maimon, 2005; Zhou, 2021; Jo, 2021). In deze studie wordt enkel met de bomen voor classificatie rekening gehouden. Het toepassen van de beslissingsbomen brengt voordelen met zich mee (Lu and Braunstein, 2014; Jo, 2021; Kuo et al., 2021; Rokach and Maimon, 2005). Ten eerste is het algoritme eenvoudig interpreteerbaar (Lu & Braunstein, 2014). Vervolgens vereist het algoritme weinig datapreparatie, zoals het schalen of normaliseren van data (Lu and Braunstein, 2014; Rokach and Maimon, 2005). Ten slotte beschikt het algoritme over het vermogen om zowel numerieke (continue) als categorische variabelen te verwerken in de bomen (Lu and Braunstein, 2014; Jo, 2021; Kuo et al., 2021). Naast de voordelen heeft het algoritme ook nadelen. Eerst en vooral zijn de bomen gevoelig voor kleine schommelingen in de data, wat tot instabiele boomstructuren kan leiden (Rokach & Maimon, 2005). Daarnaast kunnen de bomen snel overfitten bij het gebruik van irrelevante attributen of bij bomen met grote splitsingsaantallen (Rokach and Maimon, 2005; Lu and Braunstein, 2014; Zhou, 2021).

Ten slotte wordt het Support Vector Machine (SVM) algoritme vaak gebruikt in supervised learning context (Zhou, 2021). Het SVM-algoritme maakt gebruik van support vectors op een hypervlak om vervolgens een optimale scheidingsgrens te vinden die de diverse klassen in de dataset scheiden. De support vectors die worden gebruikt, zijn datapunten die het dichtst bij het scheidingsvlak liggen en de beslissing bepalen (Carlsen and Bruggemann, 2020; Jo, 2021; Zhou, 2021). Het SVM-algoritme werkt goed in hoge dimensies en resulteert in hoge accuraatheid bij goed gestructureerde data. Verder is het algoritme robuust tegen overfitting (Carlsen and Bruggemann, 2020; Zhou, 2021). Hoewel het algoritme veel moeite vereist bij het kiezen van de juiste parameters, wat op zich ook de interpretatie van het algoritme bemoeilijkt (Jo, 2021).

3.3 Feature engineering

Nadat de meest voorkomende methoden van zowel de supervised als unsupervised learning-aanpak aan bod zijn gekomen, wordt in deze sectie *feature engineering* uitgelegd. Vooraleer er dieper wordt ingegaan in deze sectie, wordt kort even uitgelegd wat een *feature* is. Een *feature* is een inputvariabele die een kenmerk vertegenwoordigt van een observatie, zoals de lengte van een kroonblad in de Iris dataset. Die inputvariabelen worden gebruikt in verschillende machine learning-modellen (Han and Kamber, 2006; Zhang, 2021; Bonilla, 2023).

Feature engineering is het proces waarbij ruwe data wordt omgezet naar relevante features die de prestaties van de machine learning-algoritmen verbeteren, door middel van transformaties, feature-creatie of featureselectie (Lee, 2019; Zhou, 2021). Het doel van *feature engineering* is, zoals eerder aangehaald, om de prestaties van de modellen te verbeteren door de structuur van de data beter te maken (Paghdal, 2025). Een betere structuur zorgt ervoor dat het model minder fouten gaat maken, beter gaat generaliseren en dus een hogere accuraatheid gaat behalen (Bonilla, 2023; Nandan et al., 2023; Trost et al., 2025). De meest voorkomende *feature engineering*-methoden in de literatuur zijn: normalisatie en standaardisatie, featureselectie, featurecreatie en discretisatie of binning (Lee, 2019; Zhang, 2021; Bonilla, 2023; Paghdal, 2025; Trost et al., 2025).

Een eerste methode is de normalisatie en standaardisatie methode die wordt toegepast op numerieke (continue) variabelen. Het doel van deze methode is om de data op een gelijke schaal te brengen,

zodat verschillen in schaalgrootte voorkomen kunnen worden en een correcte afstand berekend kan worden (Paghdal, 2025). De schaling kan gebeuren op twee manieren, namelijk (MinMax) normalisatie of standaardisatie, om vervolgens een correcte afstand te berekenen (Han and Kamber, 2006; Lee, 2019; Zhou, 2021; Jo, 2021):

- Normalisatie transformeert de data naar de waarden in het interval $[0,1]$.
- Standaardisatie schaalt de numerieke variabelen zodat de gemiddelde gelijk is aan nul en de standaardafwijking gelijk is aan één.

Ten tweede wordt de featureselectiemethode gebruikt om de meest relevante attributen (features) van de dataset te selecteren en vervolgens daarmee de algoritmes op te bouwen (Trost et al., 2025). Door enkel met de relevante attributen te werken, versnelt het trainingproces, verlaagt de complexiteit en verhoogt de prestaties van de modellen. Ook helpt het bij het reduceren van overfitting (Lee, 2019; Zhang, 2021; Trost et al., 2025). Er zijn verschillende manieren om de beste attributen te selecteren, maar de meest voorkomende zijn, onder meer univariate statistische chi-kwadraat of ANOVA-tests, *Recursive Feature Elimination (RFE)*, *Information Gain* of *SelectKBest* (Lee, 2019; Zhang, 2021; Trost et al., 2025). De focus in deze studie wordt voornamelijk op de *SelectKBest*-methode gelegd. De *SelectKBest*-methode geeft op een snelle, eenvoudige en efficiënte manier de meest relevante attributen van de datasets weer (Zhang, 2021; Jo, 2021).

Een alternatieve methode die vaak wordt gebruikt binnen machine learningcontext, is de *feature creation* methode (Bonilla, 2023). Hierbij worden nieuwe attributen samengevoegd, getransformeerd of afgeleid uit bestaande data (Jo, 2021). Een veelvoorkomend voorbeeld binnen de literatuur is het combineren van de variabelen lengte en gewicht om de variabele *Body Mass Index (BMI)* af te leiden (Zhou, 2021; Trost et al., 2025). Het doel van die nieuwe variabelen is om complexe patronen verder te verduidelijken, met het oog op verbetering van de modelprestaties (Jo, 2021; Zhou, 2021; Bonilla, 2023).

Ten slotte wordt de discretisatie- of binningmethode vaak gebruikt binnen machine learningcontext (Paghdal, 2025). Hierbij worden numerieke (continue) variabelen omgevormd tot een aantal intervallen die bepaalde categorieën voorstellen (Bonilla, 2023; Paghdal, 2025). Door categorieën aan te maken, wordt de data eenvoudiger om te interpreteren. Bovendien is deze methode beter toepasbaar op algoritmen zoals beslissingsbomen, waarbij de splitsingen efficiënter gebeuren op de categorische variabelen (Bonilla, 2023; Paghdal, 2025).

3.4 Toepassingen

In deze sectie worden enkele toepassingen van de *feature engineering* methoden, benoemd in Sectie 3.3, uit de literatuur samen met hun resultaten op de machine learning-algoritmen beschreven.

In de studie van Paghdal (2025) wordt er op het KNN-algoritme de Z-score standaardisatiemethode gebruikt. Vooraleer de schaling wordt toegepast, wordt KNN uitgevoerd op de educatieve dataset die attributen zoals *testscore*, *leeftijd* en *studieuren* bevat. Uit de resultaten bleek dat het model slechts een accuraatheid van 71% behaalde. Ook was er duidelijk te zien dat het attribuut *testscore* domineerde over de andere kenmerken. Nadat de standaardisatie werd toegepast, was onmiddellijk een stijging in de accuraatheid te zien tot 84%. Dit komt doordat alle andere attributen nu een gelijke bijdrage leveren (Paghdal, 2025). In de studie van Lee (2019) wordt de Min-Max normalisatie toegepast op klantendata. De klantendata bevat variabelen zoals *jaarinkomen*, *leeftijd* en *bestedingsscore*. Lee (2019) paste eerst de K-Means methode toe op de originele dataset. Hierbij werden onnatuurlijke clusters gevormd. Na het toepassen van de normalisatie bleek de clusterstructuur onmiddellijk te

veranderen (Lee, 2019). Daarnaast steeg de matching met de verwachte klantenclusters van 62% naar 89%. Er is dus een verbetering in de prestatie van het algoritme (Lee, 2019).

Zhang (2021) testte in zijn studie de featureselectiemethode, namelijk *SelectKBest*, op de klassieke Iris dataset. De klassieke Iris dataset bestaat uit 4 attributen, namelijk *Sepal_length*, *Sepal_width*, *Petal_length* en *Petal_width* (Varoquaux & Grobler, 2025). Door de featureselectiemethode toe te passen, daalde het aantal attributen van 4 naar 2, de attributen *Petal_length* en *Petal_width* bleven over. Dit resulteerde vervolgens ook in een stijging van de accuraatheid van het KNN-algoritme van 91% naar 96% (Zhang, 2021).

Naast featureselectie past Bonilla (2023) featurecreatie toe in zijn studie. Hiervoor gebruikt Bonilla (2023) de Titanic dataset om passagiersoverleving te voorspellen door middel van classificatie. Bonilla (2023) voegt 3 nieuwe attributen toe aan de Titanic dataset. Eerst en vooral wordt het attribuut *FamilySize* afgeleid door de som van de attributen *SibSp* en *Parch* te nemen en te verhogen met 1 (Bonilla, 2023). Vervolgens wordt op basis van het nieuwe attribuut *FamilySize* afgeleid of de passagier alleen is. Dit wordt gedaan door te kijken naar de waarden van het attribuut *FamilySize*. Als de waarde van *FamilySize* gelijk is aan 1, dan is het nieuwe attribuut *IsAlone*, een binaire variabele, ook gelijk aan 1, elders 0. Ten slotte wordt op basis van de passagiersnaam de titel afgeleid en opgeslagen onder het attribuut *Title* (Bonilla, 2023). Door die attributen aan te maken en te gebruiken, steeg de prestatie van het machine learning-algoritme van 81% naar 86% (Bonilla, 2023).

Ten slotte wordt discretisatie of binning toegepast door Paghdal (2025) en Zaidi (2017). Beide passen discretisatie (binning) toe op de Titanic dataset. Zo worden de variabelen *Age* en *Fare* in gelijke *bins* gedeeld. Die *bins* stellen intervallen voor van gelijke grootte (Zaidi et al., 2017; Paghdal, 2025). Het discretiseren van numerieke (continue) variabelen zorgt ervoor dat de interpretatie van het beslissingsboomalgoritme vergemakkelijkt. Dankzij de categorische intervallen die aangemaakt zijn, worden de splitsingen effectiever genomen en is de data beter gedefinieerd (Zaidi et al., 2017; Paghdal, 2025). Wat betreft de accuraatheid van de modellen is er noch een stijging noch een daling te zien in de waarden (Paghdal, 2025). Zaidi (2017) voegt er bijkomend aan toe dat discretisatie de invloed van outliers beperkt, waardoor het model vereenvoudigt.

In Sectie 6 worden de resultaten van de toepassingen vergeleken met de resultaten van de nieuwe methode, die in Sectie 5 worden beschreven.

4 Methodologie

In deze sectie wordt de aanpak van de experimenten besproken. Hierbij wordt dieper ingegaan op de gebruikte datasets, de gebruikte programma's en programmeertalen, en tenslotte de stappen die tijdens de experimenten per machine learning-algoritme zijn uitgevoerd. Bijkomend wordt ook vermeld waarom er specifiek voor die algoritmen wordt gekozen.

4.1 Datasets

De datasets die voor de experimenten gebruikt zullen worden, zijn voornamelijk built-in datasets in R of Python. In tabel 1 wordt een overzicht gegeven van alle built-in datasets. Voor elke dataset wordt ook het aantal observaties, aantal features alsook de types van de features en de bibliotheken vanwaar de datasets afkomstig zijn.

Dataset	Aantal Observaties	Aantal Features	Type Features	Bibliotheek
Iris	150	4	Numeriek	<code>from sklearn.datasets import load_iris</code>
USArrests	50	4	Numeriek	<code>from sklearn.datasets import fetch_openml</code>
Wine	178	13	Numeriek	<code>from sklearn.datasets import load_wine</code>
Titanic	891	10	Num. & Categorical	<code>import seaborn as sns</code>

Table 1: Overzicht van gebruikte datasets en hun eigenschappen

De datasets, terug te vinden in tabel 1, doorlopen allemaal dezelfde stappen die beschreven worden in sectie 4.3.

4.2 Programma's en programmeertalen

De opstelling van de experimenten gebeurt voornamelijk in Google Collaboratory, waar de programmeertaal Python gebruikt wordt. Naast Python wordt er ook gebruikgemaakt van R Studio, waar de programmeertaal R van toepassing is. R is een open-source programmeertaal en omgeving, speciaal ontwikkeld voor statistische berekeningen en visualisaties. In Python en R worden verschillende bibliotheken en pakketten gebruikt om toegang tot de datasets te krijgen en om de machine learning-algoritmen uit te voeren.

4.3 Methoden

In deze subsectie worden de stappen en methoden uitgelegd die worden toegepast tijdens de experimenten in de vorm van een stappenplan. Ook wordt de keuze van die methoden verantwoord door middel van bestaande literatuur. Een algemeen overzicht van de stappen ziet er als volgt uit:

- Unsupervised K-Means
- Supervised K-Means
- KNN
- KNN toegepast op de supervised K-Means data
- Beslissingsbomen
- Beslissingsbomen toegepast op de supervised K-Means data

De keuze voor de klassieke algoritmen, namelijk K-Means, K-Nearest Neighbors en beslissingsbomen, komt voornamelijk door het gebruik van kleine en gestructureerde datasets onder meer Iris, Wine, USArrests en Titanic (Carlsen and Bruggemann, 2020; Bonilla, 2023). Die datasets bevatten weinig tot geen ontbrekende waarden, waardoor het ideaal is voor de traditionele methoden (Zhang, 2021; Paghdal, 2025). Daarnaast zijn de methoden eenvoudig, transparant en geschikt voor kleine, goed gestructureerde datasets (Carlsen & Bruggemann, 2020). Ook zijn die traditionele methoden eenvoudig te implementeren en leveren resultaten met veel inzicht op (Rokach and Maimon, 2005; Zhou, 2021). K-Means is geschikt om patronen te identificeren op ongelabelde data, terwijl KNN en beslissingsbomen eenvoudig interpreteerbare classificatiemethoden zijn die geen complexe parameters vereisen (Jolly, 2018; Jo, 2021; Xing and Bei, 2020).

De DBSCAN- en SVM-methode daarentegen vereisen meer parameters en zijn complexer. Beide methoden zijn gevoelig voor de parameterkeuze, wat tot foute resultaten kan leiden (Rodriguez et al., 2019;

Zhou, 2021). De SVM-methode biedt een hoge accuraatheid, maar de interpretatie van de classificatie is veel te complex (Carlsen and Bruggemann, 2020; Zhou, 2021). Verder staat in dit onderzoek transparantie centraal, wat ervoor zorgt dat K-Means, KNN en bomen aantrekkelijkere modellen worden (Everitt et al., 2011; Jolly, 2018; Jo, 2021). Verder richt dit onderzoek zich meer op de centroiden van de clusters, wat ervoor zorgt dat de dichtheden en outlierdetectie er niet bij aansluiten (Aggarwal and Reddy, 2013; Xu and Tian, 2015; Schubert et al., 2017). Hierdoor valt de DBSCAN-methode buiten beschouwing.

Zoals eerder vermeld gebeurt de uitwerking van de experimenten voornamelijk in Python, in Google Collaboratory. Elke dataset die in het overzicht voorkomt in tabel 1 zal hetzelfde stappenplan volgen voor dit onderzoek. Om de stappen van de experimenten te verduidelijken, worden die stappen uitgelegd door naar de Iris-dataset te verwijzen.

Stap 1: Unsupervised K-Means

Een initieel algoritme dat wordt toegepast, is het unsupervised K-Means-algoritme. Hierbij wordt er eerst en vooral gekeken naar het optimale aantal clusters. Die worden op zich achterhaald door de elbow- en silhouette-methode. Op beide grafieken wordt er gekeken bij welke k (aantal clusters) er een knik of een felle verandering in de waarden voorkomt (Rodriguez et al., 2019; Jo, 2021; Zhou, 2021). Bij de Iris-dataset is het optimale clusteraantal gelijk aan 3. Nadat het aantal clusters bepaald is, worden de clusters gevisualiseerd. Bij de Iris-dataset resulteert de visualisatie in 3 willekeurige clusters. Om aan een clusteranalyse te doen, moet de data als volgt voorbereid worden (Everitt et al., 2011, Jolly, 2018, Jo, 2021, Zhou, 2021, Paghdal, 2025):

- Rijen komen overeen met observaties (individuen of instanties) en kolommen zijn variabelen.
- Ontbrekende waarden moeten geschat of verwijderd worden.
- Alle variabelen moeten NUMERIEK zijn. Nominale variabelen moeten dus *dummy coded* worden. Als de nominale variabele binair is, is het niet nodig.
- Soms is het nodig om variabelen te schalen om ze vergelijkbaar te maken en de correcte afstand te berekenen. Dit kan op 2 manieren: normalisatie en standaardisatie.
 - Standaardisatie schaalt de numerieke variabelen zodat de gemiddelde gelijk is aan nul en de standaardafwijking gelijk is aan één.
 - Normalisatie is het transformeren van variabelen in het interval $[0,1]$.

Voor dit onderzoek wordt er verder gewerkt met normalisatie. De datasets die in dit onderzoek worden gebruikt, zijn goed gestructureerd en kennen dus niet veel ontbrekende waarden. Nadat de clusters gevisualiseerd zijn, worden er per cluster centroiden toegevoegd in de vorm van een ster of asterisk. De centroiden geven het midden van de clusters weer. Eens de centroiden aanwezig zijn, vindt de preprocessingtechniek plaats. Hierbij worden de datapunten van de dataset in vijf stappen naar de dichtstbijzijnde centroiden verplaatst. De verplaatsing heeft telkens een grootte van 0,1. Na elke verplaatsing (na elke stap) worden de clusterpunten weer gevisualiseerd om het effect van de verplaatsing visueel voor te stellen. Hoewel de visualisaties nuttig kunnen zijn voor een eerste indruk, zijn ze niet voldoende om de kwaliteit van de clustering te beoordelen en daaruit iets te concluderen (Han and Kamber, 2006; Aggarwal and Reddy, 2013; Xu and Tian, 2015). Daarom worden ook de prestatie-indicatoren, namelijk de silhouette scores, de standaardafwijking en de Within Sum of Squares (WSS) scores, berekend en vergeleken per stap (Everitt et al., 2011; Aggarwal and Reddy, 2013; Rodriguez et al.,

2019).

De silhouette scores geven aan hoe goed een punt in zijn cluster is gedefinieerd (Rodriguez et al., 2019). Verder wordt de standaardafwijking berekend om aan te geven hoe de punten binnenin een cluster verspreid zijn. Hoe groter de standaardafwijking, hoe meer variatie binnen een cluster (Aggarwal & Reddy, 2013). Ten slotte geeft de WSS-score aan hoe compact een cluster is, waarbij een lagere score aangeeft dat de clusterpunten dichter bij de centroiden liggen en dus een goede clustering aangeeft (Han and Kamber, 2006; Everitt et al., 2011). Nadat de prestatie-indicatoren voor het optimale cluster aantal, voor de Iris-dataset is dat 3, zijn berekend, wordt het cluster aantal verhoogd. De verhoging van de cluster aantallen is afhankelijk van de optimale cluster aantallen. De verhoging van het cluster aantal wordt doorgevoerd om later na te gaan of er een patroon te zien is bij de verplaatsing van de datapunten naar hun centroiden. Bij de Iris-dataset stijgt het cluster aantal van 3 naar 6 clusters. Voor het verhoogde cluster aantal worden alweer de clusters samen met hun centroiden gevisualiseerd. Nadat die clusters gevisualiseerd worden, worden de datapunten hier ook naar de dichtstbijzijnde centroiden verplaatst. Die verplaatsingen op zich worden ook gevisualiseerd. Ten slotte worden hier ook de prestatie-indicatoren berekend en vergeleken met het vorige cluster aantal, voor de Iris-dataset is dat een cluster aantal van 3. Om de unsupervised K-Means clusteringmethode af te ronden, wordt het cluster aantal voor een laatste keer verhoogd. Voor de Iris-dataset wordt het cluster aantal verhoogd tot 9 clusters. Vervolgens worden er op eenzelfde manier als tevoren de visualisaties van zowel de clusters als de verplaatsingen weergegeven. Ten slotte worden hier ook de prestaties van de clusters berekend en vergeleken met de vorige twee cluster aantallen, namelijk cluster aantallen 3 en 6 voor de Iris-dataset. Bij een optimaal cluster aantal van 2 worden de cluster combinaties 2, 4 en 6 gebruikt om de analyses op eenzelfde manier dat hier wordt uitgelegd uit te voeren.

Stap 2: Supervised K-Means

Nadat de unsupervised K-Means stap is afgerond, wordt er overgeschakeld naar de supervised K-Means clusteringmethode. In deze stap worden de datasets eerst opgesplitst in een train- en test-set (Pedregosa et al., 2011; Jolly, 2018). Dit wordt gedaan door middel van een train-test-split met de verhouding 0,8-0,2, respectievelijk train en test, uit te voeren. Vervolgens wordt er verder gewerkt met de traintdata, waarop de K-Means clusteringmethode plaatsvindt. Additioneel worden de klas-labels gebruikt om de clusters te bepalen, terwijl bij de unsupervised K-Means clusteringmethode de clusters willekeurig bepaald worden. Eerst en vooral worden er dus per klasse één cluster gevisualiseerd. Dit resulteert voor de Iris-dataset in 3 clusters, want er zijn drie klassen waarvoor telkens één cluster wordt gemaakt. Vervolgens worden, net als bij de unsupervised methode, de clusters samen met hun centroiden gevisualiseerd. Vooraleer de preprocessingtechniek wordt uitgevoerd op de clusters, worden eerst de visualisaties vergeleken met de unsupervised methode. Dit wordt gedaan om na te gaan of zowel de centroiden op een andere locatie liggen als nieuwe clustervormen worden aangemaakt. Na die vergelijking worden de verplaatsingen van de datapunten weergegeven per stap. Ten slotte worden de prestaties van de clusters per stap berekend en vergeleken. Ook worden bij de supervised K-Means clusteringmethode de cluster aantallen verhoogd. De verhoging van het aantal clusters gaat van één cluster per klasse naar twee clusters per klasse. Voor de Iris-dataset resulteert dit totaal in 6 clusters. Voor het nieuwe cluster aantal worden ook de verplaatsingen gevisualiseerd en de prestaties berekend. Die prestaties worden vergeleken met het vorige cluster aantal, maar ook met de unsupervised K-Means clusteringmethode om te kijken wat het effect van supervised clustering daadwerkelijk is. Het cluster aantal wordt een laatste keer verhoogd naar drie clusters per klasse, wat voor de Iris-dataset resulteert in een totaal aantal van 9 clusters. Dezelfde stappen worden uitgevoerd voor dit cluster aantal en worden de resultaten vergeleken met de voorgaande stappen.

Stap 3: KNN

In deze stap worden eerst de k beste features (*SelectKBest*) van de datasets bepaald, afhankelijk van de grootte van de dataset (Lee, 2019; Zhou, 2021; Trost et al., 2025). De features die voor de Iris-dataset worden gebruikt, zijn de volgende: *Petal_length*, *Petal_width*, *Sepal_length* en *Sepal_width*. Nadat de beste features bepaald zijn, wordt voor die features het beste aantal burens (k) bepaald om verder te werken met KNN (Deng et al., 2016; Xing and Bei, 2020; Zhou, 2021). Hiervoor wordt er een bereik gebruikt van 1 tot en met 20 of 1 tot en met 50, afhankelijk van de grootte van de dataset. Voor de Iris-dataset wordt het bereik van 1 tot en met 20 gebruikt, want de dataset bevat enkel 150 observaties. Voor elke k wordt er dan gekeken naar de hoogste accuracy score; vervolgens wordt die beste k genomen voor het vervolg van het experiment (Deng et al., 2016; Xing and Bei, 2020; Zhou, 2021). De hoogste accuracy score voor de Iris-dataset werd behaald door het aantal burens gelijk aan 3. Het bepalen van het optimale aantal burens (k) zorgt ervoor dat zowel een te klein als een te groot aantal vermeden wordt. Een te klein aantal kan voor overfitting zorgen, waarbij het model zich te veel aan de traindata gaat aanpassen. Terwijl bij een te grote k het model te veel vereenvoudigingen gaat maken en voor underfitting zorgt (Deng et al., 2016; Xing and Bei, 2020; Zhou, 2021). Eens het beste aantal burens (k) bepaald is, wordt er visueel een KNN voorgesteld met de beslissingsgrenzen per klasse. Het KNN-algoritme dat hier wordt toegepast, is eigenlijk een classificatie-algoritme dat de datapunten van de Iris-dataset gaat classificeren op basis van de labels van de burens. KNN wordt in dit experiment dus eerst en vooral toegepast op de unsupervised K-Means clusteringmethode. Vervolgens worden er per klasse centroiden toegevoegd, net als bij de K-Means clustering. Daarna worden de datapunten verplaatst naar hun dichtstbijzijnde centroiden, op eenzelfde manier als de K-Means clustering. Bij elke stap wordt er gekeken naar de accuraatheid van het model, voor de vijf stappen, en hoe die verschilt met het oorspronkelijke model waar geen verplaatsing aanwezig is (Xing & Bei, 2020). Ook worden de stappen toegenomen tot 10 om het effect van een verdere verplaatsing te bekijken. Ook worden de visualisaties weergegeven van de vijf stappen van verplaatsing. Dit wordt gedaan om te kijken wat voor effect dit gaat hebben op de beslissingsgrenzen van de KNN's (Jolly, 2018; Zhou, 2021).

Stap 4: KNN toegepast op de supervised K-Means data

Voor de volgende stap wordt het KNN-algoritme toegepast op de supervised K-Means clusteringmethode. Hier werd er dus initieel een train-test-split uitgevoerd om 80% over te houden als traindata en 20% als testdata. Ook werden de clusters per klasse gevormd in plaats van willekeurige clusters te vormen, net als bij de unsupervised methode. Het aantal burens en attributen dat gebruikt wordt voor dit KNN-algoritme komt overeen met die in Stap 3. Voor de Iris-dataset was dat aantal gelijk aan 3 en waren de attributen *Petal_length*, *Petal_width*, *Sepal_length* en *Sepal_width* gebruikt. De KNN's worden per stap gevisualiseerd met de beslissingsgrenzen, om het effect van de verplaatsing te achterhalen. Ook worden de accuracies van de vijf stappen weergegeven, maar verschillend van de gewone KNN worden zowel de train- als testaccuracies per stap weergegeven. Het weergegeven van beide accuracies zorgt ervoor dat er gekeken kan worden naar de kloof tussen de train- en testaccuracies. Hoe groter de kloof wordt naarmate de stappen vorderen, hoe meer het model gaat overfitten (Han and Kamber, 2006; Pedregosa et al., 2011; Zhou, 2021). Bijkomend worden de stappen toegenomen tot tien om het effect van een verdere verplaatsing weer te geven.

Stap 5: Beslissingsbomen

Een finale methode die tijdens de experimenten wordt uitgevoerd, zijn de beslissingsbomen. Ook worden tijdens deze methode de beste features, via de *SelectKBest*-methode, achterhaald om verder

mee te werken. Nadat de features achterhaald zijn, worden er telkens drievoudige crossvalidaties uitgevoerd per stap. Dat zorgt ervoor dat er drie bomen per stap worden gemaakt, waarvan twee met traindata worden gemaakt en één boom met de testdata. De beslissingsbomen kunnen snel overfitten door de traindata, hierdoor wordt er verder gewerkt met drievoudige crossvalidatie om meer betrouwbare gemiddelden op te leveren dan enkel met één testset Rokach and Maimon, 2005. Verder is crossvalidatie een robuuste techniek voor generalisatie (Han & Kamber, 2006). Initieel wordt dit op de originele dataset getest. Vervolgens worden de bomen toegepast op de K-Means clusteringmethode, waarbij de datapunten naar hun dichtstbijzijnde centroiden verplaatsen. Er wordt enkel gewerkt met het optimale clusteraantal. Het optimale clusteraantal voor de Iris-dataset is gelijk aan 3 clusters. Vervolgens wordt er telkens na verplaatsing van de datapunten naar hun dichtstbijzijnde centroiden de drievoudige crossvalidatie toegepast. Het doel van deze stap is om na te gaan wat voor effect de verplaatsingen hebben op de structuur van de bomen, alsook de accuraatheid ervan (Rokach and Maimon, 2005; Kuo et al., 2021). Voor de structuur worden de bomen van de vijf stappen gevisualiseerd. De accuracies daarentegen worden weergegeven van 10 stappen om het effect op een verdere verplaatsing te analyseren. Naast de structuur en accuraatheid wordt er ook gekeken naar de feature importance per stap. De feature importances geven aan hoeveel invloed een bepaald attribuut op de splitsingsbeslissing heeft (Jolly, 2018; Zhou, 2021; Kuo et al., 2021). Bij de Iris-dataset wordt er dus gekeken naar de invloed die de vier variabelen, *Petal_length*, *Petal_width*, *Sepal_length* en *Sepal_width*, op de splitsbeslissing hebben. Een extra toevoeging aan de beslissingsbomen kan de parameter *max_depth* zijn. Deze toevoeging is afhankelijk van het aantal splitsingen dat zich in de bomen bevindt. Als er te veel splitsingen zijn, worden die beperkt tot een maximale diepte van 4. Een maximale diepte toekennen aan de bomen zorgt ervoor dat overfitting wordt beperkt en dat de complexiteit van de bomen wordt beheerst (Rokach and Maimon, 2005; Han and Kamber, 2006). Er zijn veel minder splitsingen te zien, terwijl de bomen veel dieper kunnen gaan. Dit komt niet van toepassing bij de Iris-dataset.

Stap 6: Beslissingsbomen toegepast op de supervised K-Means data

Om de experimenten af te ronden, worden als finale stap nog de beslissingsbomen toegepast op de supervised K-Means clusteringmethode. Er wordt dus gewerkt met de traindata van 80% verkregen uit de train-test-split. Ook vinden de clusters en verplaatsingen per klasse plaats. Per stap worden er hier ook drie beslissingsbomen gemaakt, aan de hand van drievoudige crossvalidatie. Vervolgens wordt er per stap gekeken naar de structuur van de bomen, de accuracies en de feature importances. Bij de structuur worden voornamelijk het aantal splitsingen en de voorwaarden van de attributen waarop de splitsbeslissingen gebeuren geanalyseerd. Daarnaast worden de train- en testaccuracies per stap weergegeven om te kijken hoe de verplaatsingen die accuracies beïnvloeden. De accuracies worden weergegeven voor 10 stappen om een beter beeld te krijgen van een verdere verplaatsing. Ten slotte worden ook de belangen van de attributen per stap geanalyseerd. De attributen die hier worden gebruikt, zijn hetzelfde als de attributen die in Stap 5 worden gebruikt, voor de gewone beslissingsbomen. Voor de Iris-dataset zijn dat de volgende: *Petal_length*, *Petal_width*, *Sepal_length* en *Sepal_width*. Die importances worden op zich ook vergeleken met die van de beslissingsbomen toegepast op de unsupervised K-Means.

5 Resultaten experimenten

In deze sectie worden de resultaten van de experimenten per dataset besproken. Om een overzicht te behouden, worden de datasets in de volgorde van tabel 1 behouden. Ook worden de stappen van Sectie 4.3 gevolgd.

5.1 Iris

De Iris-dataset is een van de bekendste en meest gebruikte datasets in machine learning. Die dataset is een built-in dataset die gebruikt kan worden in R of in Python. De meeste toepassingen van de dataset zijn voor classificatie-algoritmen zoals K-Means, KNN en beslissingsbomen. De dataset bevat informatie over 150 irisbloemen, opgedeeld in 3 klassen, namelijk Setosa, Versicolor en Virginica. Elk bloem is beschreven aan de hand van vier numerieke variabelen die verschillende metingen van de bloem beschrijven. *Sepal_length* geeft de lengte van het kelkblad weer en wordt uitgedrukt in cm. *Sepal_width* geeft de breedte van het kelkblad weer in cm. Naast de kelkbladeren is er ook informatie beschikbaar over de kroonbladeren. *Petal_length* geeft de lengte van het kroonblad weer in cm en *Petal_width* geeft de breedte van het kroonblad weer in cm (Varoquaux & Grobler, 2025).

5.1.1 Unsupervised K-Means

Startend met de unsupervised K-Means clustering, wordt er voor de Iris-dataset gekozen om te werken met 3, 6 en 9 clusters. De keuze die hiervoor wordt gemaakt, komt door de elbow- en silhouettemethode weergegeven in figuur 1. Beide methoden geven aan dat de optimale clusterkeuze gelijk is aan 3. Een groter clusteraantal zorgt ervoor dat er veel meer clusters zijn, waardoor het model veel rekening gaat houden met kleine details. Hierdoor gaat het model beginnen overfitten. Terwijl een te laag clusteraantal voor underfit zorgt.

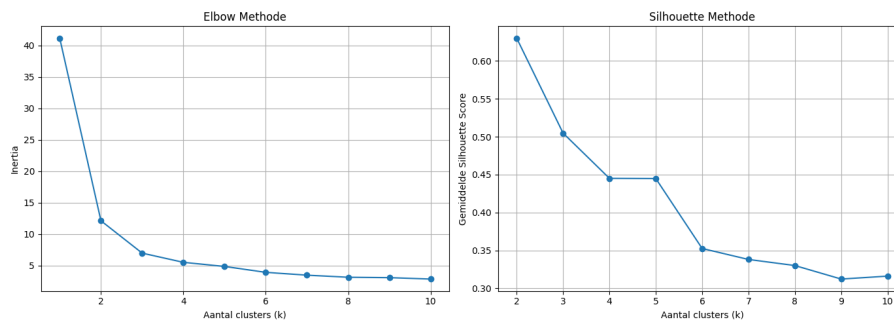


Figure 1: Elbow- en Silhouettemethode - Iris-dataset

3 clusters

Initieel wordt de clustering met 3 clusters gevisualiseerd en weergegeven in figuur 2. Nadat de initiële clustering is uitgevoerd, worden de centroiden toegevoegd aan de clusters en weergegeven in figuur 3. Hiervoor wordt er gebruik gemaakt van de Principal Component Analysis. Na het toevoegen van de centroiden worden de datapunten in de richting van hun dichtstbijzijnde centroiden verplaatst. Er worden 5 stappen genomen met telkens een grootte van 0,1.

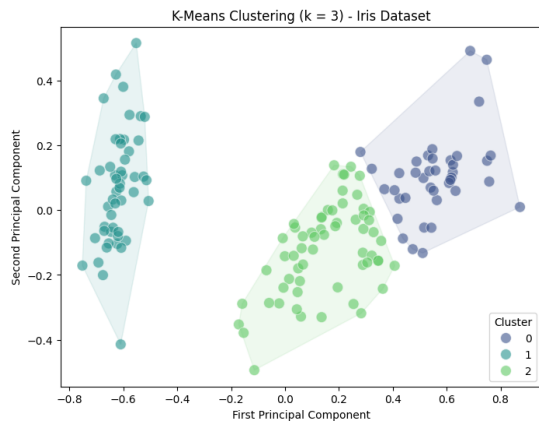


Figure 2: Initiële clustering Iris-dataset ($k = 3$)

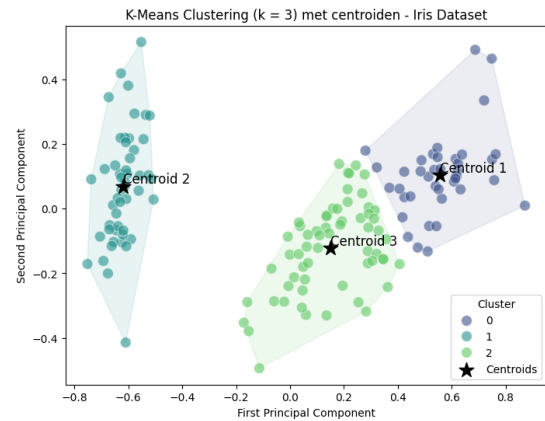


Figure 3: Initiële clustering Iris-dataset ($k = 3$) met centroiden

In figuur 4 worden alle stappen van de verplaatsing gevisualiseerd. Uit die grafieken is duidelijk af te leiden dat de datapunten dichter bij de centroiden komen en dus de clusters veel compacter en kleiner worden. Het is moeilijk te achterhalen wat voor effect dit werkelijk heeft op de clusters. Hiervoor worden verschillende prestatie-indicatoren berekend en vergeleken met de initiële clustering.

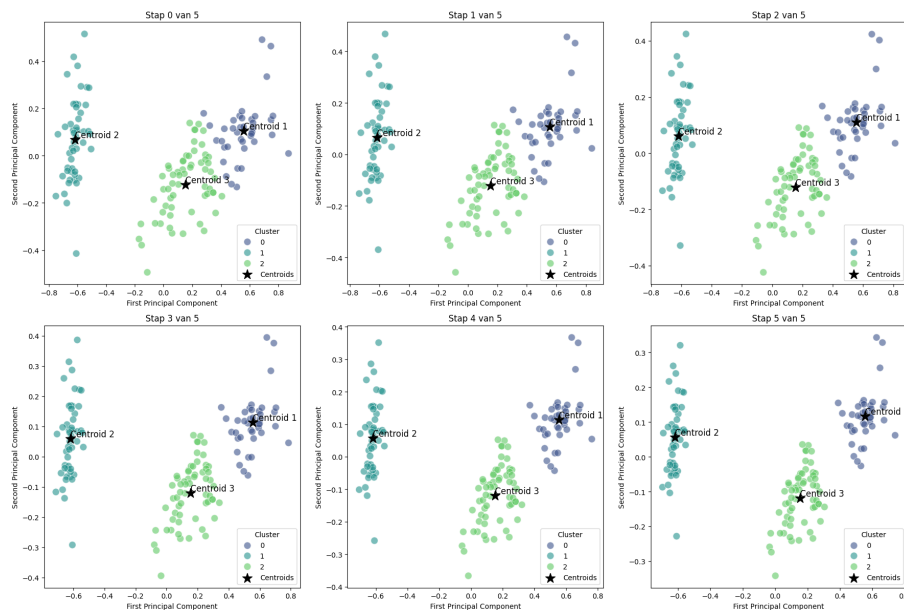


Figure 4: Clusters na verplaatsing van 0,1 grootte - Iris-dataset ($k = 3$)

Eerst en vooral worden de silhouette scores voor alle stappen berekend. De silhouette scores geven aan hoe goed een punt in zijn cluster is gedefinieerd. Uit onderstaande boxplot, weergegeven in figuur 5, is af te leiden dat de silhouette scores toenemen, naarmate de punten dichter bij de centroiden terechtkomen.

Vervolgens wordt de standaardafwijking berekend, die aangeeft hoe de punten verspreid zijn binnen een cluster. Hoe groter de standaardafwijking, hoe meer variatie binnen een cluster. Uit de boxplots in figuur 6 is af te lezen dat er minder variatie binnen een cluster is, naarmate de punten dichter bij de centroiden komen.

Ten slotte wordt er naar de Within-cluster Sum of Squares (WSS) gekeken. De WSS geeft aan hoe compact een cluster is. Een lagere WSS-score geeft aan dat de punten dicht bij de centroiden liggen, wat een goede clustering aangeeft. Terwijl een hogere score aangeeft dat de punten verspreid zijn en slecht geclusterd worden. In figuur 7 is er af te leiden dat de WSS-scores afnemen naarmate de punten dichterbij de centroiden verplaatsen. Dit wil zeggen dat er betere clusters gedefinieerd worden.

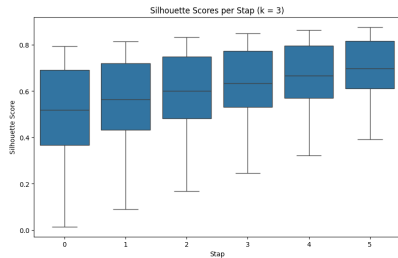


Figure 5:
Gemiddelde Silhouette (k = 3)

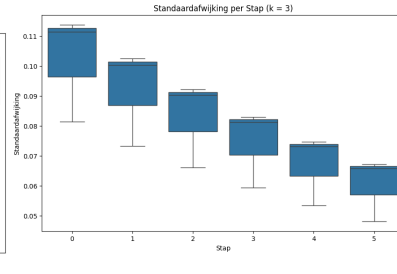


Figure 6:
Gemiddelde Standaardafwijking
(k = 3)

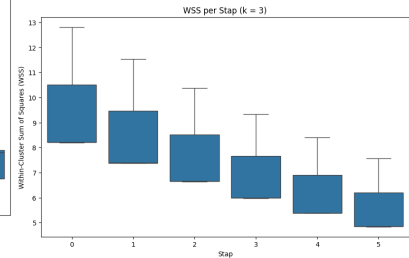


Figure 7:
Gemiddelde WSS-scores (k = 3)

6 clusters

Door het clusteraantal te laten toenemen, verschillen de locaties en de aantallen van de centroiden. Hierbij wordt er dus gekeken wat het effect is als de clusters gelijk zijn aan 6. In figuur 8 worden de 6 clusters met hun centroiden gevisualiseerd.

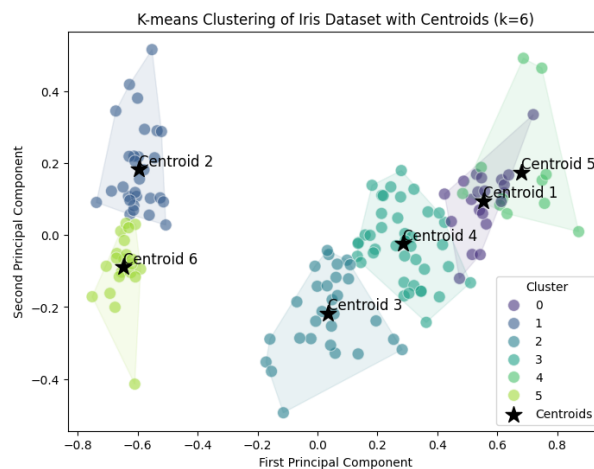


Figure 8: Initiële clustering Iris-dataset (k = 6)

Eens de clusters gevisualiseerd zijn, worden de datapunten verplaatst naar de centroiden. De verplaatsing wordt gevisualiseerd in figuur 9. Uit deze figuur is af te leiden dat de clusters minder datapunten bevatten dan tevoren, vanwege een groter clusteraantal.

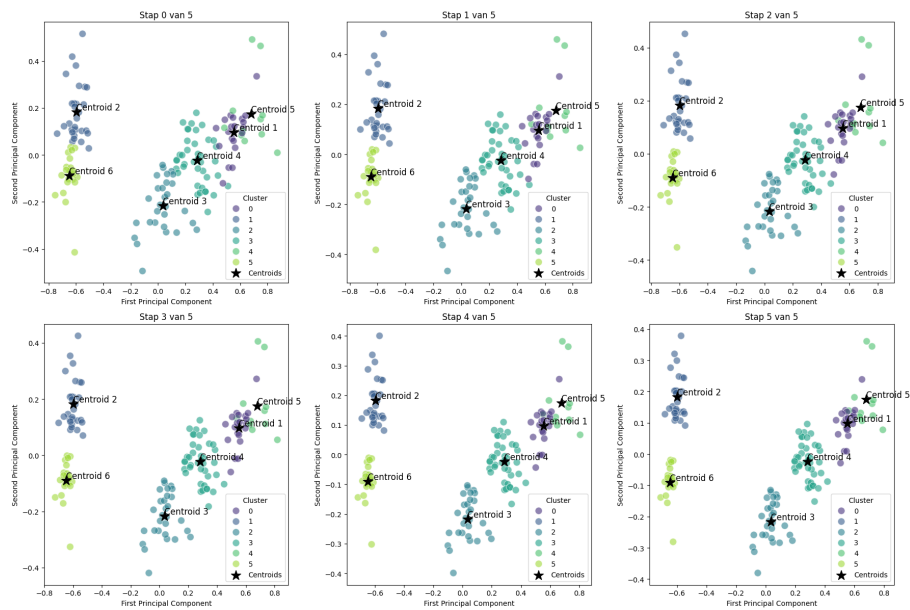


Figure 9: Clusters met verplaatsing van 0,1 grootte - Iris-dataset ($k = 6$)

Om meer inzichten te verkrijgen, wordt er gekeken naar de prestatie-indicatoren. De silhouette scores geven bij een clusteraantal van 6 ook aan dat er een stijging is naarmate de datapunten dichter bij hun centroiden terechtkomen. Hoewel de scores lager liggen dan die van figuur 5.

Verder vertonen de standaardafwijkingen, in figuur 11, een gelijkaardig patroon als bij een clusteraantal gelijk aan 3. Hoewel de waarden hier lager liggen, vanwege de aanwezigheid van het aantal datapunten per cluster. Ieder cluster omvat bij een clusteraantal gelijk aan 6 veel minder datapunten, waardoor de spreiding van de punten ook verschilt.

Ten slotte zijn de WSS-scores, weergegeven in figuur 12, ook lager, maar vertonen een gelijkaardige daling als bij 3 clusters. Dit geeft aan dat de clusters veel compacter zijn bij 6 clusters dan bij 3.

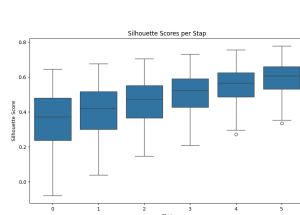


Figure 10:
Gemiddelde Silhouette ($k = 6$)

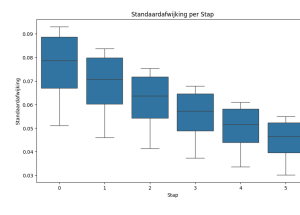


Figure 11:
Gemiddelde Standaardafwijking
($k = 6$)

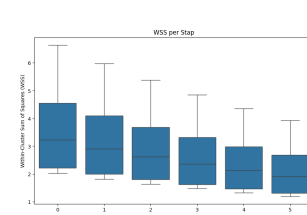


Figure 12:
Gemiddelde WSS-scores ($k = 6$)

9 clusters

Ten slotte wordt er voor de unsupervised K-Means clusteringmethode bij de Iris-dataset gewerkt met 9 clusters. De initiële clustering met 9 clusters wordt weergegeven in figuur 13. Op de figuur is er duidelijk te zien dat de centroiden op een andere plaats gaan liggen dan bij een ander clusteraantal.

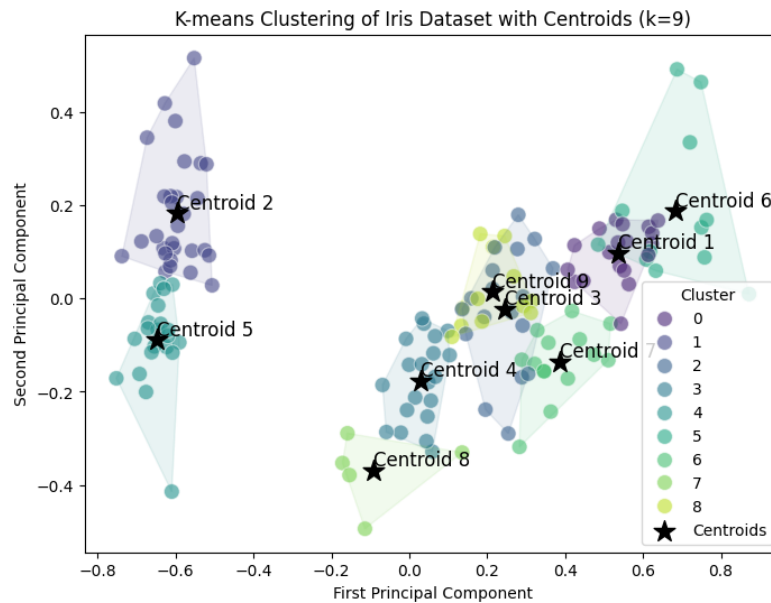


Figure 13: Initiële clustering Iris-dataset (k = 9)

De verplaatsing van de datapunten naar hun centroiden wordt weergegeven in figuur 14. Daar is ook af te leiden dat de clusters veel minder datapunten bevatten.

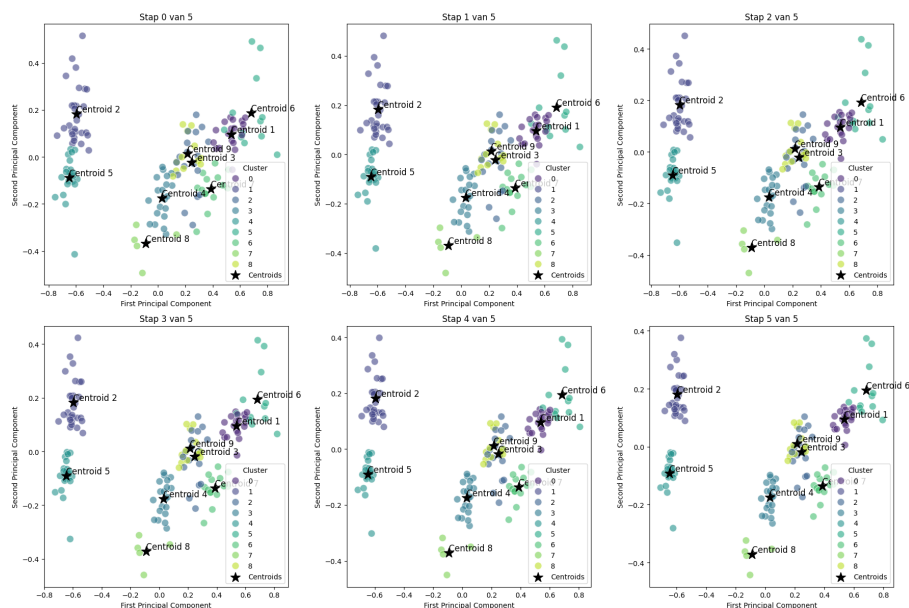


Figure 14: Clusters met verplaatsing van 0,1 grootte - Iris-dataset (k = 9)

Door de prestatie-indicatoren te berekenen voor alle stappen met 9 clusters, worden de volgende boxplots voorgesteld. De silhouette scores geven hetzelfde weer als bij 6 clusters. In figuur 15 is er een stijging van de silhouette scores af te lezen, naarmate de punten dichterbij hun centroiden liggen. Dit geeft aan dat de clusters beter gedefinieerd zijn.

Uit de boxplot van de standaardafwijkingen, weergegeven in figuur 16, is er een daling af te lezen van de standaardafwijking van de clusters. Dit is eenzelfde patroon als bij de andere twee cluster aantallen. Hoewel de waarden hier nog lager liggen dan die van de cluster aantallen 3 en 6. Een lagere

standaardafwijking duidt erop dat er veel minder spreiding is binnen de clusters.

Ten slotte geldt hetzelfde voor de WSS-scores, weergegeven in figuur 17. Naarmate de datapunten verplaatsen in de richting van hun dichtstbijzijnde centroiden, neemt de WSS-score af. Ook liggen de waarden van de WSS-scores bij 9 clusters lager dan die van de cluster aantallen 3 en 6. Dit geeft aan dat er veel beter geclusterd wordt bij 9 clusters.

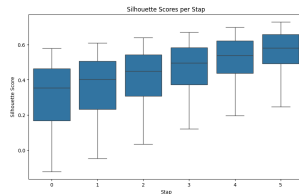


Figure 15:
Gemiddelde Silhouette (k = 9)

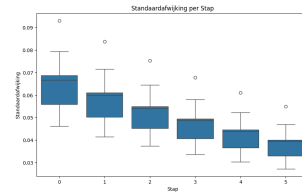


Figure 16:
Gemiddelde Standaardafwijking
(k = 9)

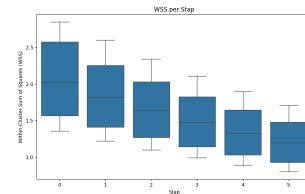


Figure 17:
Gemiddelde WSS-scores (k = 9)

5.1.2 Supervised K-Means

Na het afronden van de unsupervised K-Means clustering, wordt de Iris-dataset opgesplitst in traindata en testdata. De verhouding die hiervoor wordt gebruikt is 80% traindata en 20% testdata. Nadat de data is opgesplitst, wordt de traindata gebruikt om clusters te vormen per klasse. Initieel wordt er gewerkt met één cluster per klasse. Vervolgens wordt het aantal verhoogd naar twee clusters per klasse. Ten slotte worden er drie clusters per klasse gevisualiseerd. De cluster aantallen komen hierbij dus overeen met de cluster aantallen van de unsupervised K-Means, namelijk 3, 6 en 9. Hetzelfde cluster aantal zorgt ervoor dat beide methoden met elkaar vergeleken kunnen worden.

3 clusters

Startend met één cluster per klasse, worden op figuur 18 de initiële clusters weergegeven, samen met hun centroiden. Vergeleken met de figuur van de unsupervised clustering is er een lichte wijziging in de clusters te zien, alsook een wijziging in de plaats van de centroiden. Er wordt dus verder naar de verplaatsing gekeken, alsook de prestatie-indicatoren na verplaatsing bij dit cluster aantal.

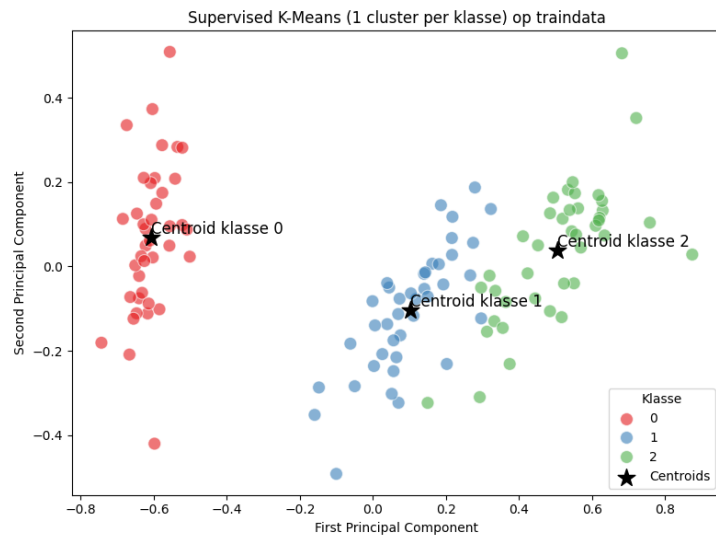


Figure 18: Supervised clustering Iris-dataset ($k = 3$)

Vervolgens worden de datapunten naar hun centroiden verplaatst en weergegeven in figuur 19. Uit de figuur blijkt dat de clusters na verplaatsing kleiner worden. De punten liggen minder verspreid. Echter blijkt dat de clusters bij de unsupervised methode compacter zijn.

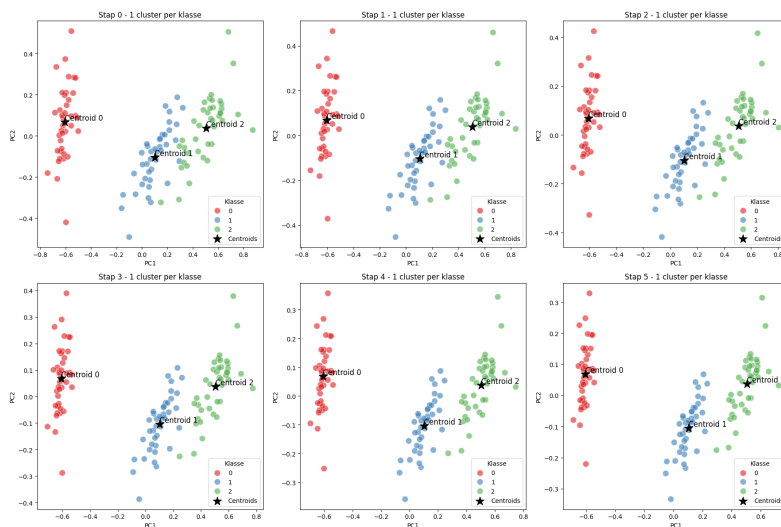


Figure 19: Supervised clusters na verplaatsing van 0,1 grootte - Iris-dataset ($k = 3$)

Ten slotte worden de prestatie-indicatoren van de clusters berekend. In figuur 20 worden de box-plots van de silhouette scores weergegeven. Uit die figuur blijkt dat de datapunten beter gedefinieerd worden naarmate de punten zich verplaatsen. Dit is te zien door de stijging per stap, net als bij de unsupervised clusteringmethode. De gemiddelde waarden van de silhouette scores liggen iets hoger dan bij de unsupervised methode, wat aangeeft dat de supervised clusteringmethode beter de datapunten definieert in de clusters.

De standaardafwijkingen, weergegeven in figuur 21, vertonen eenzelfde gedrag als bij de unsupervised methode, namelijk een daling na elke stap. De gemiddelde waarden van de standaardafwijkingen liggen bij de supervised clusteringmethode een klein beetje lager. Dit geeft aan dat de punten iets

minder verspreid zijn binnen de clusters. De WSS-scores liggen ook lager bij de supervised methode. In figuur 22 worden de WSS-scores weergegeven en daaruit blijkt dat de scores eenzelfde dalingsgedrag vertonen als bij de unsupervised methode.

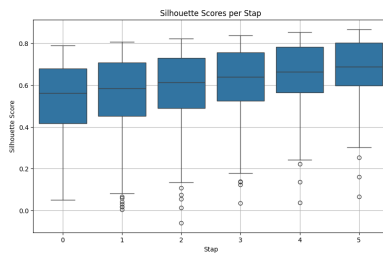


Figure 20:
Gemiddelde Silhouette ($k = 3$)

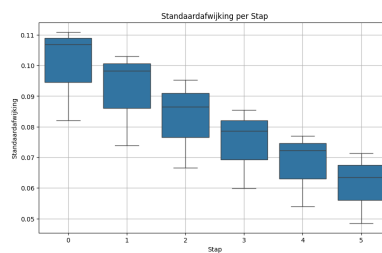


Figure 21:
Gemiddelde Standaardafwijking
($k = 3$)

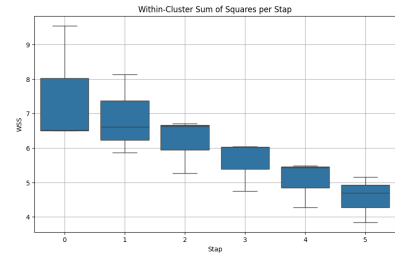


Figure 22:
Gemiddelde WSS-scores ($k = 3$)

6 clusters

Voor de visualisatie van de zes clusters, dus twee clusters per klasse, met hun centroiden is er duidelijk in figuur 23 te zien dat de centroiden op een andere plaats liggen dan bij de unsupervised methode. Hiermee wordt er dus verder gewerkt om te kijken wat het resultaat gaat zijn van de verplaatsing van de datapunten naar hun dichtstbijzijnde centroiden.

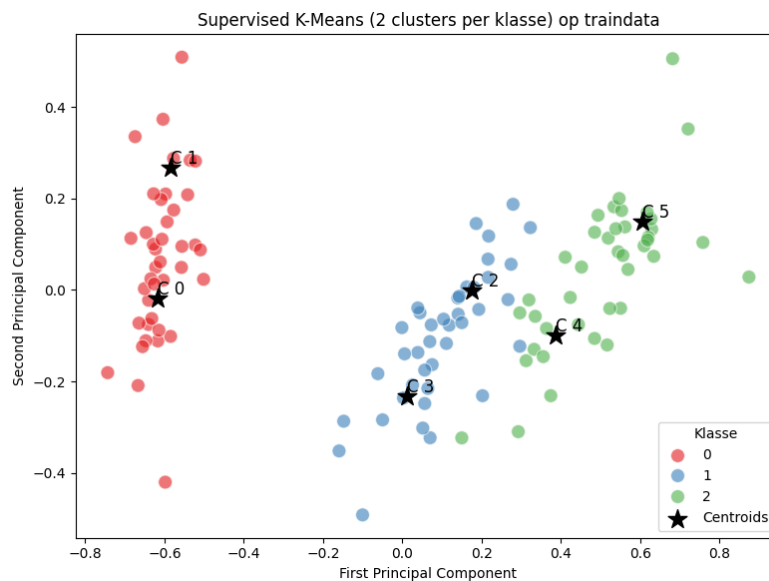


Figure 23: Initiële supervised clustering Iris-dataset ($k = 6$)

Vervolgens worden de datapunten verplaatst naar hun dichtstbijzijnde centroiden, weergegeven in figuur 24. Uit de figuur is er duidelijk te zien dat de centroiden op een andere plaats liggen en dat er ook andere clusters gevormd zijn, in vergelijking met de unsupervised methode. Ook zijn de clusters veel kleiner.

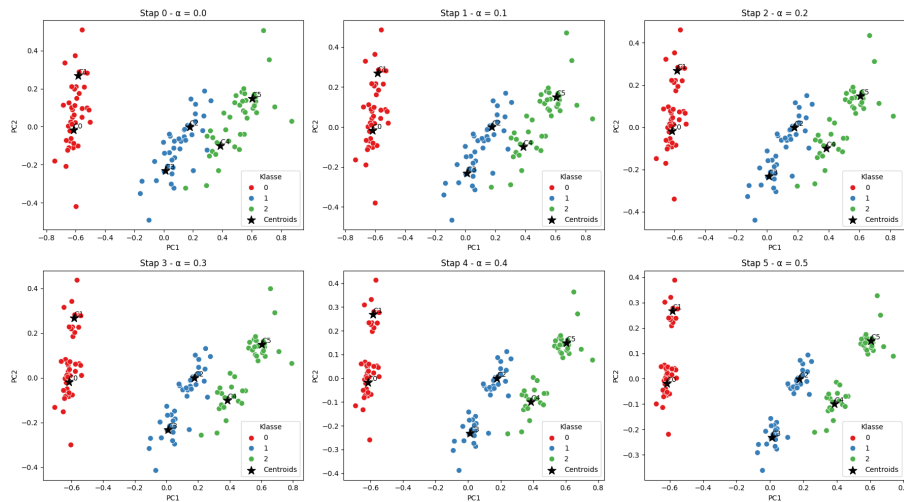


Figure 24: Supervised clusters na verplaatsing van 0,1 grootte - Iris-dataset (k = 6)

De prestatie-indicatoren voor het cluster aantal 6 blijven ongeveer hetzelfde bij zowel supervised als unsupervised K-Means clustering. De silhouette scores, weergegeven in figuur 25, vertonen eenzelfde patroon en liggen de gemiddelde waarden ongeveer op hetzelfde niveau. Daarnaast is er af te leiden in figuur 26 dat de standaardafwijkingen dalen naarmate de punten dichterbij hun centroiden liggen. Hoewel de waarden van de standaardafwijkingen ongeveer hetzelfde blijven als die van de unsupervised methode. Ten slotte worden de WSS-scores van de supervised methode met zes clusters weergegeven in figuur 27. Uit die figuur is er af te leiden dat de WSS-scores dalen. Dit is een identiek gedrag als bij de unsupervised methode. De waarden van de WSS-scores liggen hier wel lager dan bij de unsupervised methode, dat wil zeggen dat er bij de supervised methode de clusterpunten dichterbij de centroiden liggen en dus compactere clusters vormen.

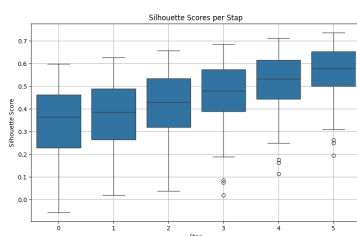


Figure 25:
Gemiddelde Silhouette (k = 6)

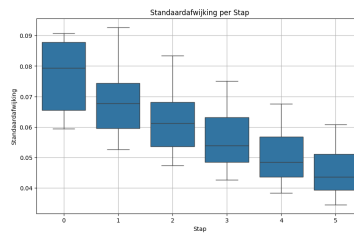


Figure 26:
Gemiddelde Standaardafwijking
(k = 6)

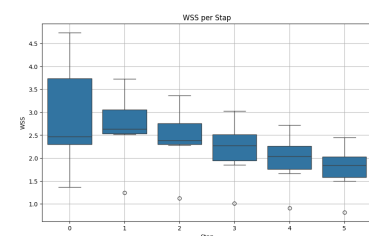


Figure 27:
Gemiddelde WSS-scores (k = 6)

9 clusters

Ten slotte wordt er voor de Iris-dataset met 9 clusters gewerkt, drie clusters per klasse. In figuur 28 worden de 9 clusters met hun centroiden weergegeven. Er is duidelijk te zien dat de centroiden op een andere plaats liggen dan bij de unsupervised methode.

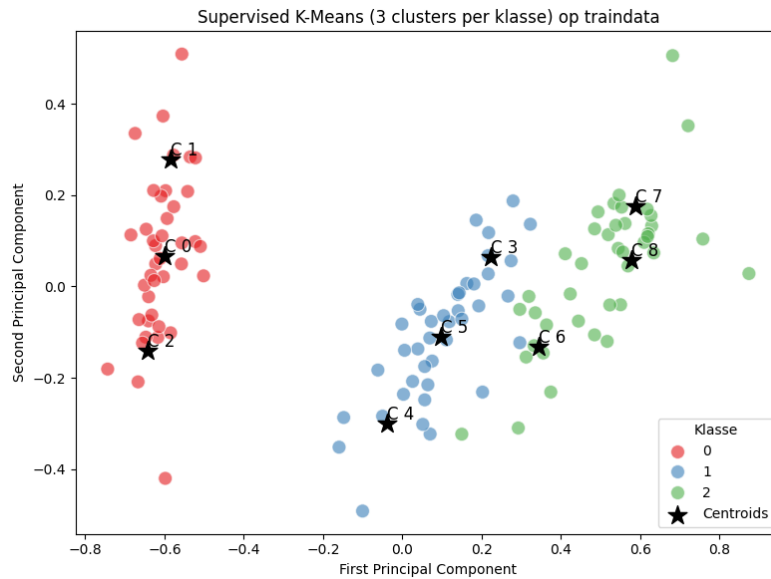


Figure 28: Initiële clustering Iris-dataset ($k = 9$) met centroiden

Na de initiële clustering met 9 clusters, worden de datapunten in de richting van hun dichtstbijzijnde centroiden verplaatst en weergegeven in figuur 29. Uit die figuur is af te leiden dat de clusters veel kleiner beginnen te worden.

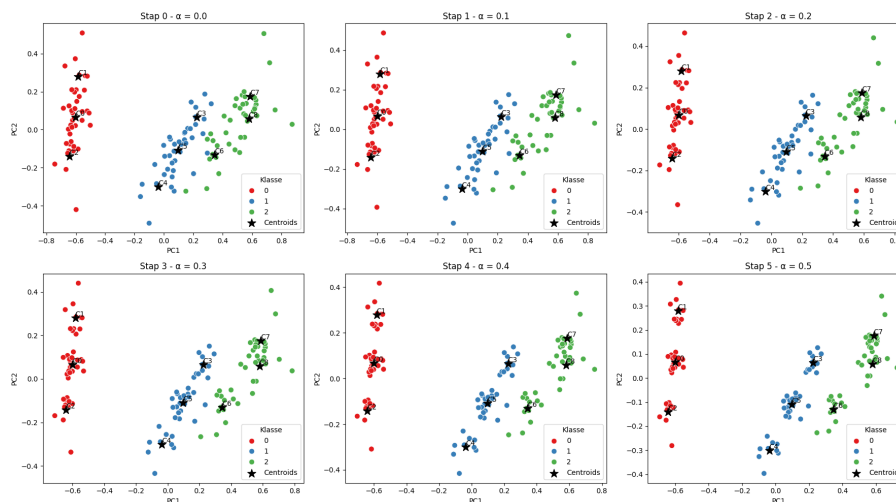


Figure 29: Supervised clusters na verplaatsing van 0,1 grootte - Iris-dataset ($k = 9$)

Ten slotte wordt er gekeken naar de prestatie-indicatoren bij 9 clusters. Op het eerste vlak vertonen de silhouette scores, weergegeven in figuur 30, eenzelfde patroon als bij de unsupervised methode. De scores beginnen te stijgen naarmate de punten verplaatsen. De waarden van die scores komen ongeveer overeen met die van de unsupervised methode. In figuur 31 worden de standaardafwijkingen weergegeven. De standaardafwijkingen dalen net als bij de unsupervised methode. Hoewel de gemiddelde waarden van de standaardafwijkingen bij de supervised methode lager liggen. Ten slotte kennen de WSS-scores ook een daling bij de supervised methode. De waarden van die scores, weergegeven in figuur 32, liggen bij de supervised methode ook lager dan bij de unsupervised methode.

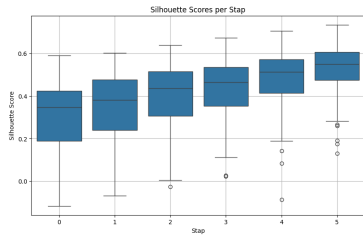


Figure 30:
Gemiddelde Silhouette (k = 9)

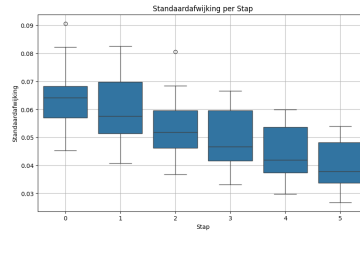


Figure 31:
Gemiddelde
Standaardafwijking (k = 9)

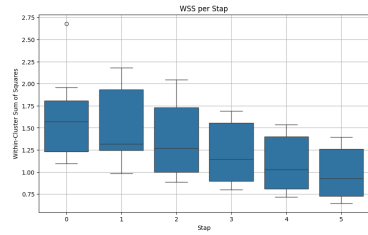


Figure 32:
Gemiddelde WSS-scores (k = 9)

5.1.3 KNN

In deze sectie wordt het KNN-algoritme uitgelegd dat werd gebruikt voor de Iris-dataset. KNN werd uitgevoerd voor een aantal k-waarden om het optimale aantal burens te kiezen. De accuracy van die KNN's ligt het hoogst bij k = 3. Bij k = 3 bedraagt de accuracy 76,67%. Voor k = 1 wordt meestal nooit gekozen, want dit zorgt voor overfitten. k = 1 zorgt ervoor dat elk datapunt geclassificeerd wordt op basis van één buur. Als dat ene buurpunt een andere klasse is, kan het leiden tot een foute classificatie. Hierdoor wordt er gekozen voor een grotere, maar niet al te grote k. Een veel te grote k kan leiden tot underfitten, waardoor er geen patronen worden herkend. In figuur 33 worden de KNN-algoritmen visueel voorgesteld voor verschillende aantallen burens. Uit de figuur is te zien dat de scheidelijnen veranderen bij een ander aantal burens. Verder worden de centroiden toegevoegd en worden vervolgens de datapunten naar de dichtstbijzijnde centroide verplaatst.

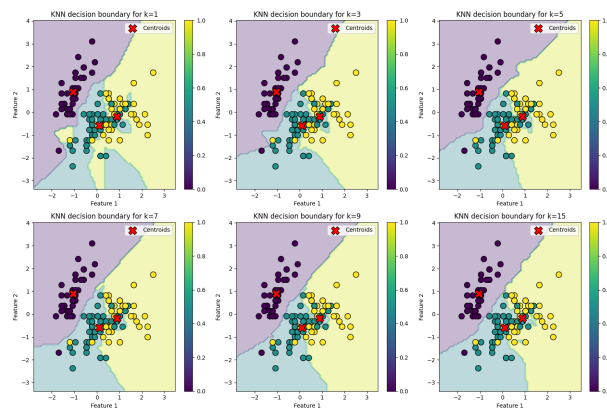


Figure 33: KNN met k = [1, 3, 5, 7, 9, 15] en centroiden

Om de datapunten te verplaatsen, wordt er verder gewerkt met k = 3, omdat deze de hoogste accuracy heeft. De datapunten worden naar de dichtstbijzijnde centroide verplaatst. In totaal worden er 5 stappen gezet met telkens een grootte van 0,1. In figuur 34 worden telkens de verplaatsingen weergegeven, samen met de accuracy van de stappen.

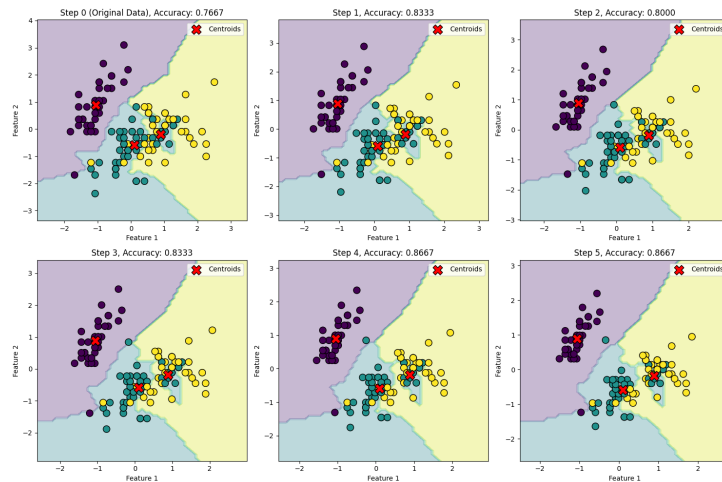


Figure 34: KNN's van alle stappen by verplaatsing van 0,1 - Iris-dataset ($k = 3$)

Om een beter beeld te krijgen van de accuracies per stap, wordt er in figuur 35 een lijngrafiek weergegeven die de accuraatheid van het model voor de 5 stappen toont. In die grafiek is te zien dat de accuracy stijgt en een piek bereikt, maar na 3 stappen zal de accuracy terugsdalen. In figuur 36 worden de stappen toegenomen van 5 naar 10 met telkens een grootte van 0,1. Uit de figuur is af te leiden dat de accuraatheid van het model een constante waarde bereikt, namelijk 86,67%.

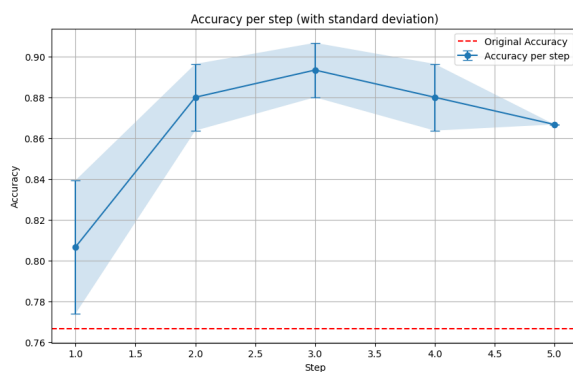


Figure 35: Accuracy per stap (5 stappen)

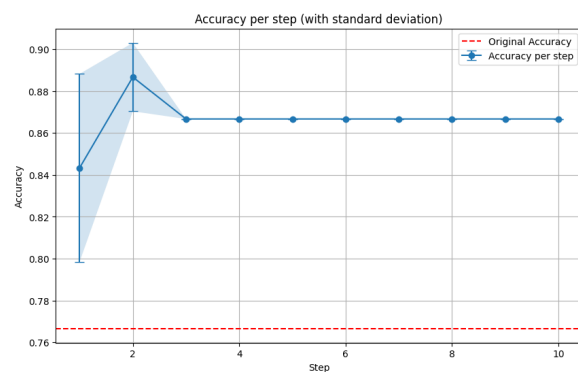


Figure 36: Accuracy na 10 stappen

5.1.4 KNN toegepast op de supervised K-Means data

Na het opsplitsen van de Iris-dataset in train- en testdata, wordt de supervised KNN-stap uitgevoerd. Hierbij wordt het KNN-classificatie-algoritme toegepast op de verplaatsingen van de datapunten die gebeurd zijn bij de supervised K-Means. Initieel wordt de visualisatie met $k = 3$ samen met de centroiden voorgesteld in figuur 37. De accuracy van deze methode is hoger dan die van de KNN-methode op de volledige dataset.

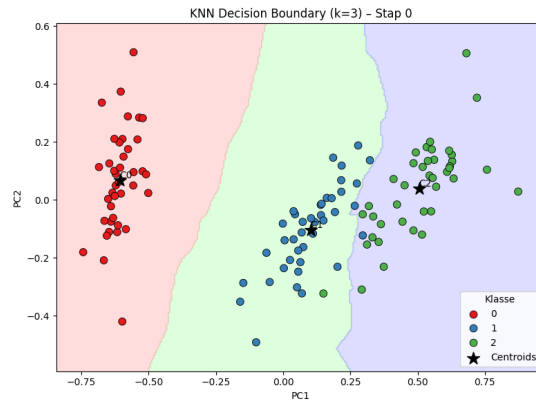


Figure 37: KNN-visualisatie toegepast op supervised K-Means - Iris-dataset ($k = 3$)

Vervolgens worden de datapunten naar hun dichtstbijzijnde centroiden verplaatst, in 5 stappen met telkens een grootte van 0,1. De verplaatsing wordt weergegeven in figuur 38. Uit de figuur blijkt dat oorspronkelijk de datapunten relatief verspreid zijn in de klassen. Naarmate de stappen vorderen, is er duidelijk te zien dat de beslissingsgrenzen van KNN rechter en stabielere worden. Dit is te verklaren door de compactere clusters per klasse.

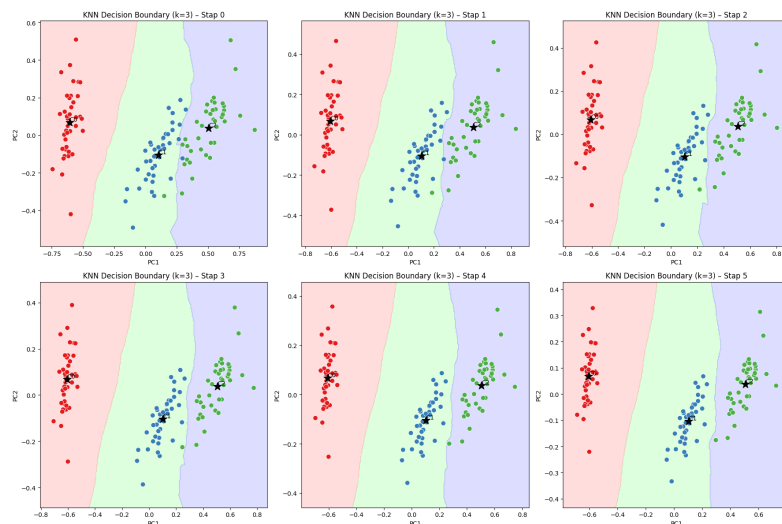


Figure 38: KNN's van alle stappen toegepast op supervised K-Means by verplaatsing van 0,1 - Iris-dataset ($k = 3$)

De evolutie van de accuraatheid van het model wordt weergegeven in figuur 39. De grafiek toont, net zoals de gewone KNN die reeds werd uitgevoerd, een stijging in zowel de train- als testaccruacy. De trainaccruacy bereikt na stap 2 een accruacy van 100%, wat aangeeft dat het model perfect de traintdata leert. De testaccruacy stijgt van 93% naar 97%. Voor de overige stappen wordt die waarde constant behouden. Het verplaatsen van de datapunten naar de centroiden maakt dus de klassen duidelijker gescheiden, waardoor de prestaties verbeteren. Om meer te weten te komen over de stijging, worden de stappen van de verplaatsing verhoogd tot 10. Indien de stappen worden toegenomen tot 10, dan is er duidelijk in figuur 40 te zien dat de accuraatheid een constante waarde blijft behouden tot en met stap 10. Die constante accruacy bedraagt ongeveer 97%.

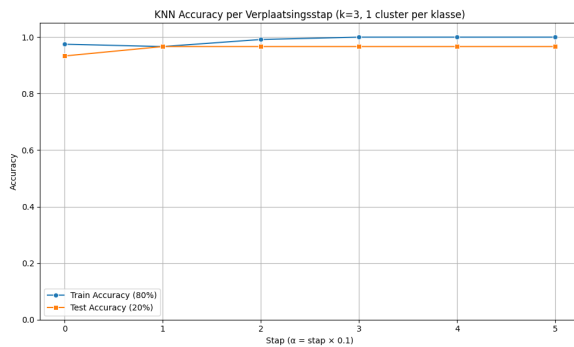


Figure 39: KNN toegepast op supervised K-Means accuracy (5 stappen)

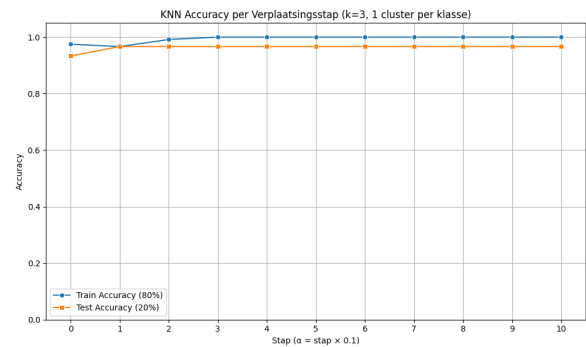


Figure 40: KNN toegepast op supervised K-Means accuracy (10 stappen)

5.1.5 Beslissingsbomen

Ten slotte worden beslissingsbomen gemaakt om hier ook naar de accuracy te kijken. Naast de accuracy wordt er ook gekeken naar de structuur van de bomen. Eerst en vooral wordt er aan de hand van drievoudige crossvalidatie drie initiële bomen gecreëerd van de Iris-dataset, toegepast op de K-Means clusteringmethode. Die bomen worden weergegeven in figuur 41. Wat af te leiden is uit de drie bomen, is dat de structuur sterk verschilt. Hiermee wordt er bedoeld dat de splitsingen niet hetzelfde zijn, alsook de attributen waarop gesplitst wordt. Zo is er een duidelijk verschil bij de derde fold. De tweede splitsing daar gebeurt niet op hetzelfde attribuut als de eerste twee folds.

In een volgende stap worden de datapunten naar hun dichtstbijzijnde centroiden verplaatst, zoals eerder gedaan bij de K-Means en KNN. Na elke verplaatsing wordt er gekeken wat voor effect dit heeft op de bomen.

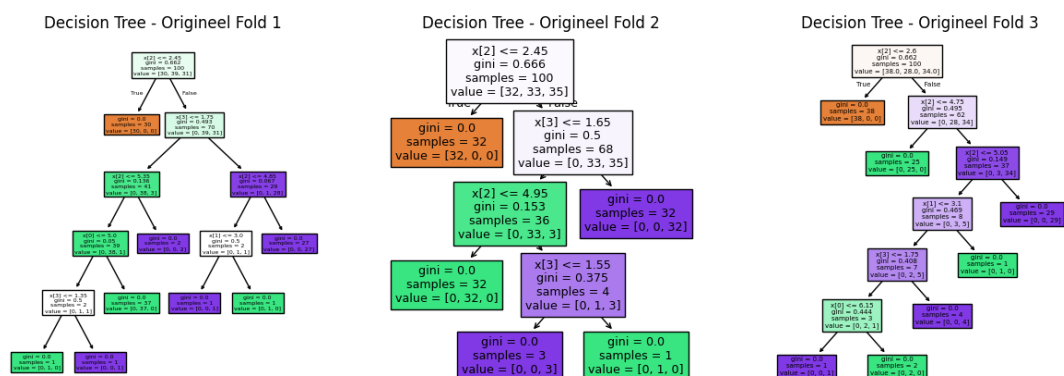


Figure 41: Initiële beslissingsbomen - Iris-dataset

In figuur 42 worden de beslissingsbomen per stap van verplaatsing weergegeven. Elke stap omvat een verplaatsing van de datapunten met telkens een drievoudige crossvalidatie, waardoor er drie bomen weergegeven worden. Naarmate de punten dichterbij de centroiden geraken, verminderen de splitsingen van de boom en gaat de structuur ook minder variëren. Dit geeft aan dat de boom de data beter gaat uitleggen. Naast de structuur is er ook te zien dat de eerste splitsingen ongeveer hetzelfde zijn voor alle 3 de bomen. Vanaf stap 2 is er te zien dat de tweede splitsing gebeurt op basis van hetzelfde attribuut, namelijk *Petal_width*. Ook is er vanaf stap 4 te zien dat de bomen een identieke structuur, dezelfde vertakkingen en aantallen, hebben.

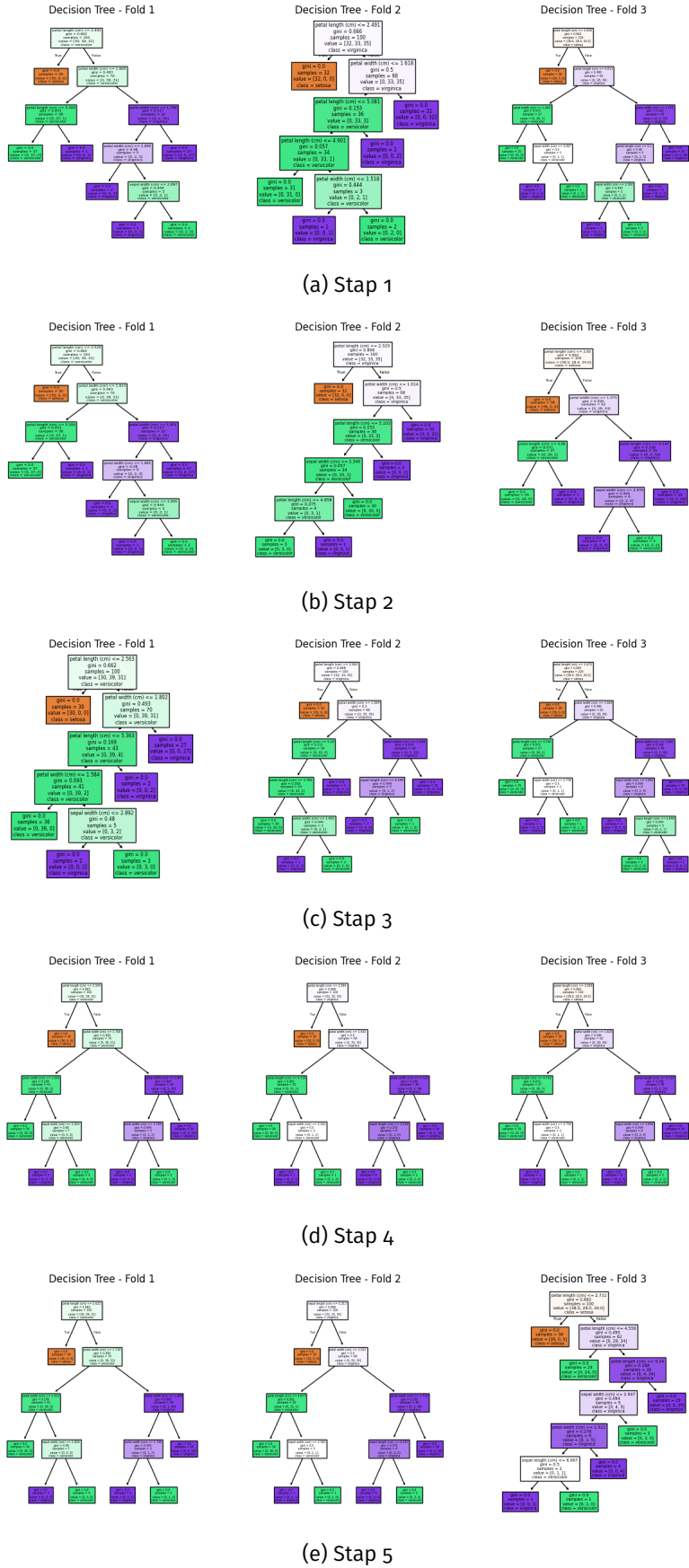


Figure 42: Beslissingsbomen per stap van verplaatsing - Iris-dataset

Verder wordt er gekeken naar de accuracies van de bomen. De accuracy van de eerste vijf stappen na verplaatsing wordt in figuur 43 voorgesteld. Naarmate de punten dicht bij hun centroiden komen, neemt de accuraatheid van de bomen af. Hoewel er een daling is, is die daling zeer klein. Vervolgens wordt er gekeken wat er na de vijfde stap gebeurt door de stappen te laten toenemen naar 10. In figuur 44 worden de accuracies van de 10 stappen weergegeven. Uit die grafiek blijkt dat de accuracy na stap 6 terug zal stijgen en dichtbij de originele accuracy zal geraken.

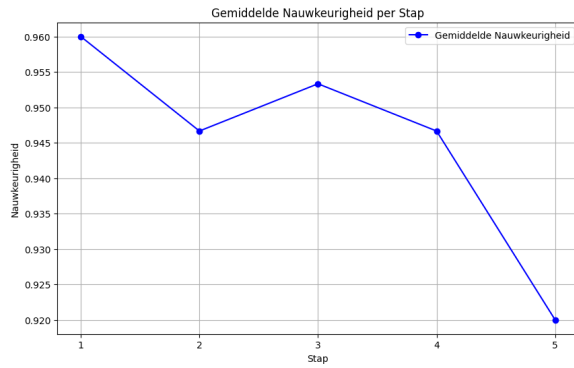


Figure 43: Gemiddelde accuracy beslissingsbomen (5 stappen)

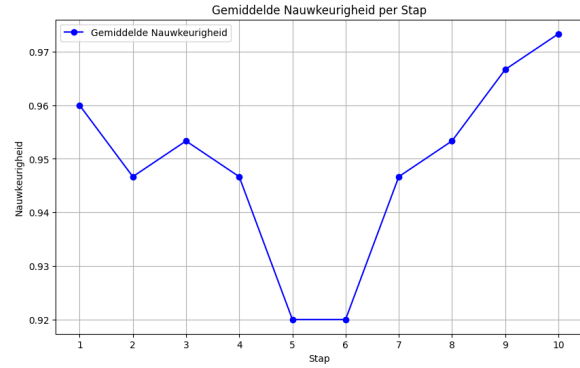


Figure 44: Gemiddelde accuracy beslissingsbomen (10 stappen)

Ten slotte wordt er gekeken naar de feature importance per stap. Voor de beslissingsbomen worden de 4 features van de Iris dataset gebruikt, namelijk *Sepal_length*, *Sepal_width*, *Petal_length* en *Petal_width*. Op figuur 45 en 46 is te zien dat de belangrijkheid van de feature *Petal_length* het grootst is. De feature importance geeft aan welke attributen van de dataset het meest bijdragen aan de beslissingen en splitsingen van de data. Wat ook opvalt, is dat die belangrijkheid afneemt naarmate de datapunten verplaatsen naar hun centroiden.

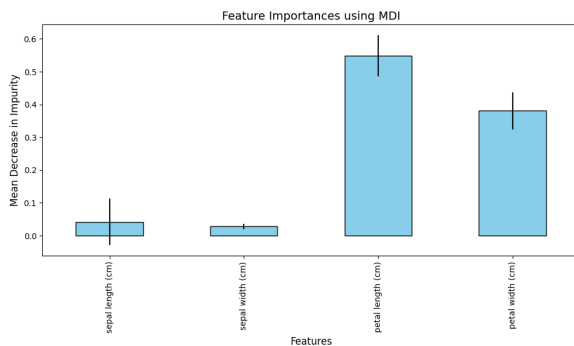


Figure 45: Feature Importance (boxplot)

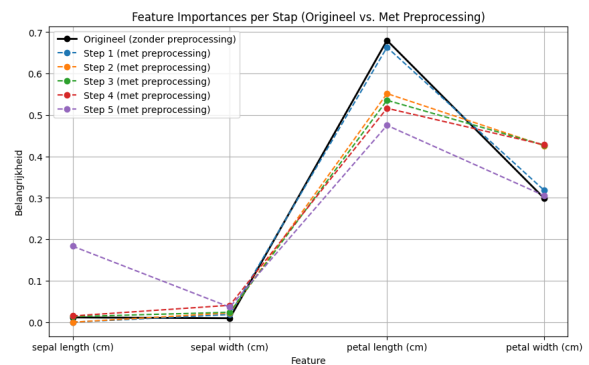


Figure 46: Feature Importance per stap (5 stappen)

5.1.6 Beslissingsbomen toegepast op de supervised K-Means data

Door de data verkregen uit de supervised K-Means clusteringmethode te gebruiken om beslissingsbomen daarop toe te passen als classificatiemethode, worden de bomen in figuur 47 weergegeven. De bomen van de supervised aanpak hebben minder vertakkingen dan de gewone aanpak. Verder is er ook te zien dat de structuur van de bomen verschilt, net als bij de gewone aanpak. De derde vertakking gebeurt bij de tweede fold op basis van de *Petal_width* attribuut, terwijl de laatste fold de splitsing doet op basis van het *Sepal_length* attribuut. Ook is het aantal vertakkingen bij fold twee meer dan bij de andere twee folds.

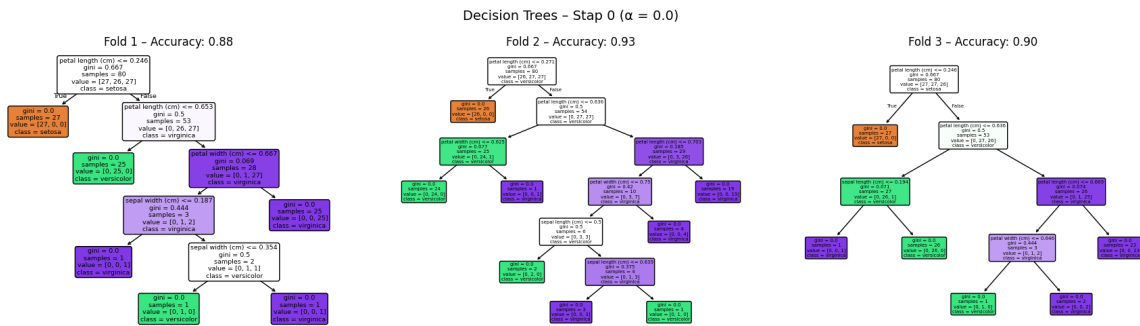


Figure 47: Initiële beslissingsbomen toegepast op supervised K-Means - Iris-dataset

Door de datapunten richting hun dichtstbijzijnde centroiden te verplaatsen, worden de drievoudige crossvalidaties uitgevoerd en weergegeven in figuur 48. De figuren van de supervised beslissingsbomen vertonen een gelijkaardig gedrag als de gewone beslissingsbomen. Naarmate de punten dichter bij de centroiden komen, krijgen de bomen een gelijkaardige structuur. Ook is er te zien dat de eerste splitsing op basis van dezelfde attributen gebeurt. Vanaf stap 3 gebeurt de tweede splitsing op basis van het attribuut *Petal_width* voor alle drie de bomen. Ook beginnen de beslissingswaarden waarop de splitsingen gebeuren te dalen naarmate de punten verplaatsen. Dit betekent dat er van een bredere splitsingen overgeschakeld wordt naar een effectievere, smallere splitsing.

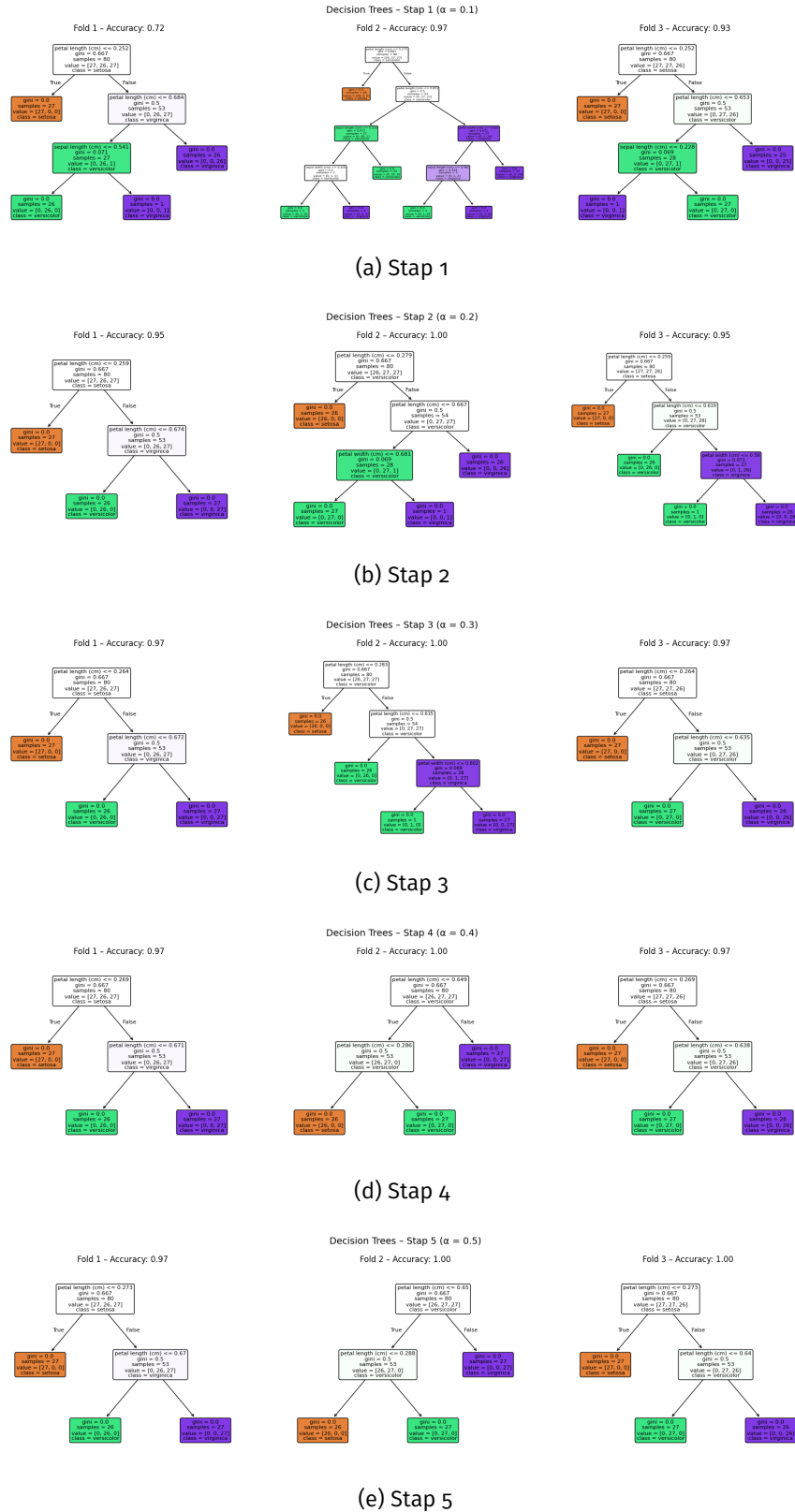


Figure 48: Beslissingsbomen toegepast op supervised K-Means per stap van verplaatsing - Iris-dataset

In figuur 49 worden de accuracies van de eerste 5 stappen weergegeven. Zowel de train- als testaccuracies worden per stap weergegeven. Uit de figuur blijkt dat de trainaccuracy 100% blijft voor de vijf stappen, terwijl de testaccuracy een stijging kent en vervolgens een daling. Door de stappen te laten

toenemen naar 10, is er duidelijk te zien dat de testaccuracy terug naar de oorspronkelijke waarde zakt. Hierbij is er duidelijk overfitting zichtbaar, want het model leert de traindata perfect, maar presteert niet beter op de testdata. In het algemeen heeft de preprocessingstechniek weinig tot geen effect op de accuraatheid van de modellen, maar wel op de structuur van de bomen.

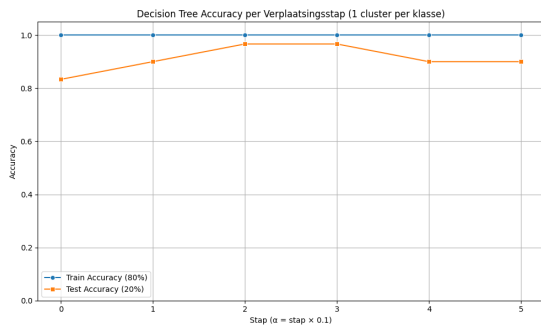


Figure 49: Gemiddelde accuracy beslissingsbomen (5 stappen)

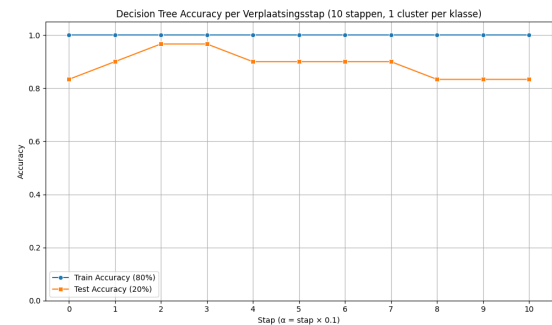


Figure 50: Gemiddelde accuracy beslissingsbomen (10 stappen)

Ten slotte wordt er gekeken naar de belangrijkheid van de features. Die worden weergegeven in figuur 51. De figuur vertoont een ander patroon dan bij de gewone beslissingsbomen. Voor de eerste vijf stappen neemt het belang van het attribuut *Petal_length* toe. Na stap vijf kent het belang een sterke daling en is vanaf stap 7 niet meer het meest dominante attribuut. Het attribuut *Sepal_length* overschrijdt het belang van *Petal_length* en heeft dus de meeste invloed op de splitsingen.

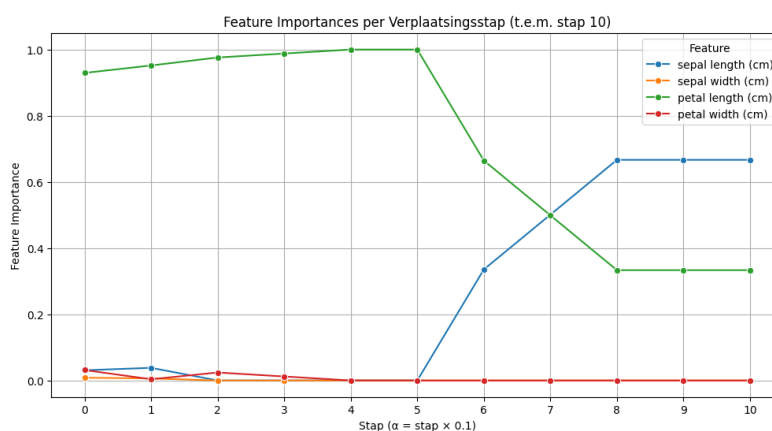


Figure 51: Feature Importance per stap (10 stappen)

5.2 USArrests

Een tweede dataset die voor de experimenten wordt gebruikt, is de USArrests. De USArrests dataset is een built-in dataset in R of Python, waarin informatie over verschillende arrestaties binnen de Verenigde Staten is terug te vinden. De dataset bevat 50 observaties, één per staat, over Amerikaanse staten en 4 variabelen die de criminaliteit per staat aangeven. De criminaliteit is gebaseerd op het aantal arrestaties per 100.000 inwoners. De eerste variabele is de variabele *Murder*. Die variabele geeft het aantal moorden per 100.000 mensen in een staat aan. Vervolgens wordt de variabele *Assault* gebruikt om het aantal aanvallen of andere geweldsmisdrijven per 100.000 mensen in een staat aan te geven. Verder wordt de *UrbanPop* variabele gebruikt om het percentage van de bevolking in een staat dat in stedelijke gebieden woont aan te geven. Ten slotte geeft de variabele *Rape* het aantal verkrachtingen per 100.000 mensen in een staat weer (McNeil, 2025).

5.2.1 Unsupervised K-Means

Uit de elbow- en silhouettemethode die op de USArrests dataset wordt toegepast, worden de twee grafieken in figuur 52 weergegeven. Uit die grafieken blijkt dat een clusteraantal gelijk aan 2 de beste keuze is. Om ook resultaten te verkrijgen van experimenten met grotere clusteraantallen, worden de clusteraantallen 4 en 6 in beschouwing genomen. Weliswaar bevat de USArrests dataset enkel 50 observaties, waardoor een clusteraantal van 4, maar vooral een clusteraantal van 6 zal resulteren in clusters met slechts één of twee datapunten.

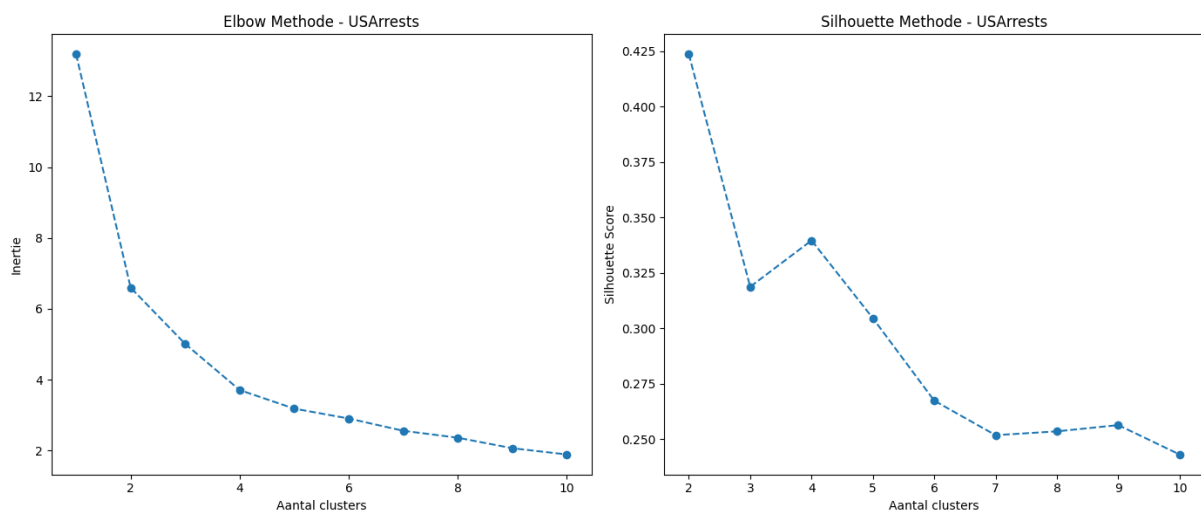


Figure 52: Elbow- en Silhouettemethode - USArrests dataset

2 clusters

Bij de unsupervised methode wordt er gestart met het clusteraantal 2. De visualisatie van de clusters met hun centroiden wordt weergegeven in figuur 53.

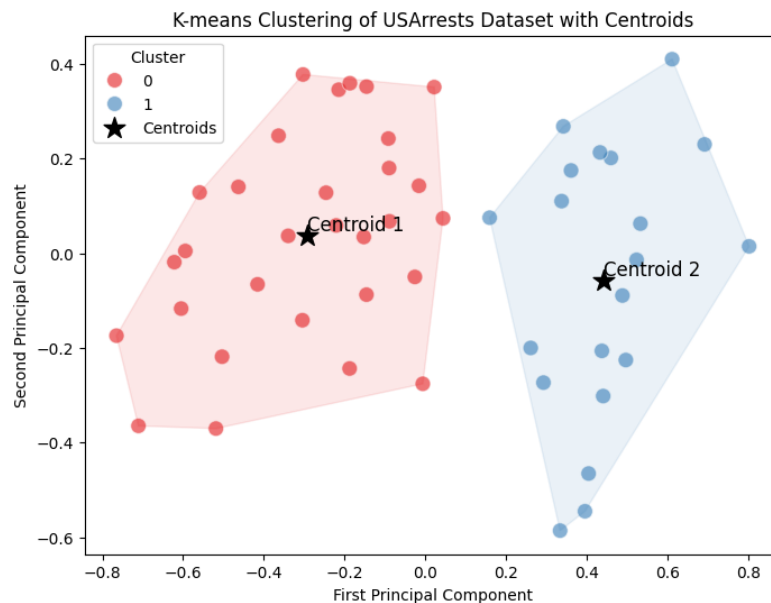


Figure 53: Initiële clustering USArrests dataset ($k = 2$) met centroiden

Vervolgens worden de datapunten van de clusters verplaatst naar hun dichtstbijzijnde centroiden. De verplaatsing wordt weergegeven in figuur 54. Elke stap van de verplaatsing is 0,1 groot en wordt herhaald voor 5 stappen. Uit de figuur blijkt dat de centroiden veel compacter worden en dat de punten minder verspreid zijn. Om dit te bevestigen, worden de prestatie-indicatoren berekend.

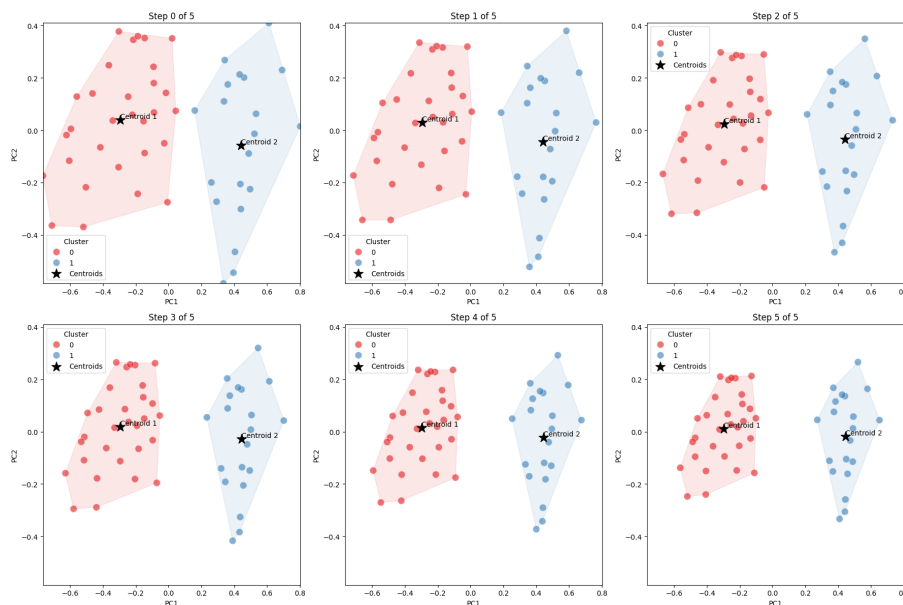


Figure 54: Clusters na verplaatsing van 0,1 grootte - USArrests dataset ($k = 2$)

In figuur 55 worden de silhouette scores gevisualiseerd per stap in de vorm van boxplots. Uit die grafiek

blijkt dat er een stijging van de silhouette scores is, naarmate de punten dichterbij hun centroiden liggen. De datapunten worden beter gedefinieerd in de clusters. Naast de silhouette scores is er in figuur 56 duidelijk te zien dat de standaardafwijkingen afnemen. Er is dus weinig spreiding tussen de datapunten binnenin een cluster. Ten slotte worden de WSS-scores weergegeven in figuur 57. Zoals ook te zien was op de clustervisualisatie in figuur 54, nemen de WSS-scores af wat aangeeft dat de clusters compacter worden. Kortom, zorgt het verplaatsen van de datapunten naar hun dichtstbijzijnde centroiden ervoor dat een goede clustering plaatsvindt.

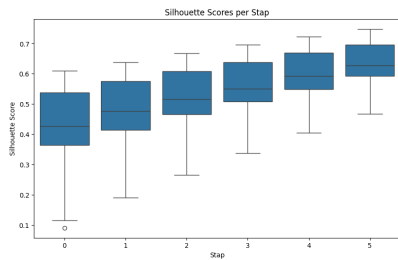


Figure 55:
Gemiddelde Silhouette (k = 2)

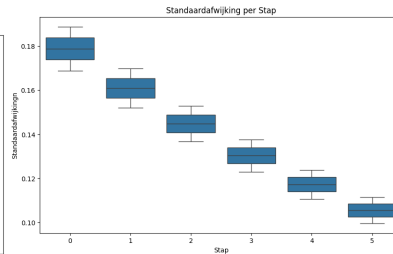


Figure 56:
Gemiddelde Standaardafwijking
(k = 2)

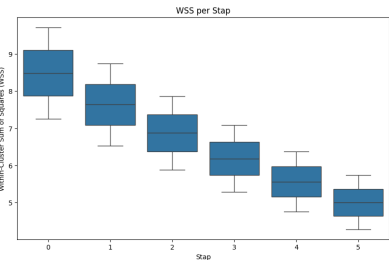


Figure 57:
Gemiddelde WSS-scores (k = 2)

4 clusters

De cluster aantallen worden vervolgens toegenomen van 2 naar 4 clusters. Die clusters worden met hun centroiden weergegeven in figuur 58. Uit die figuur is te zien dat de clusters minder datapunten bevatten.

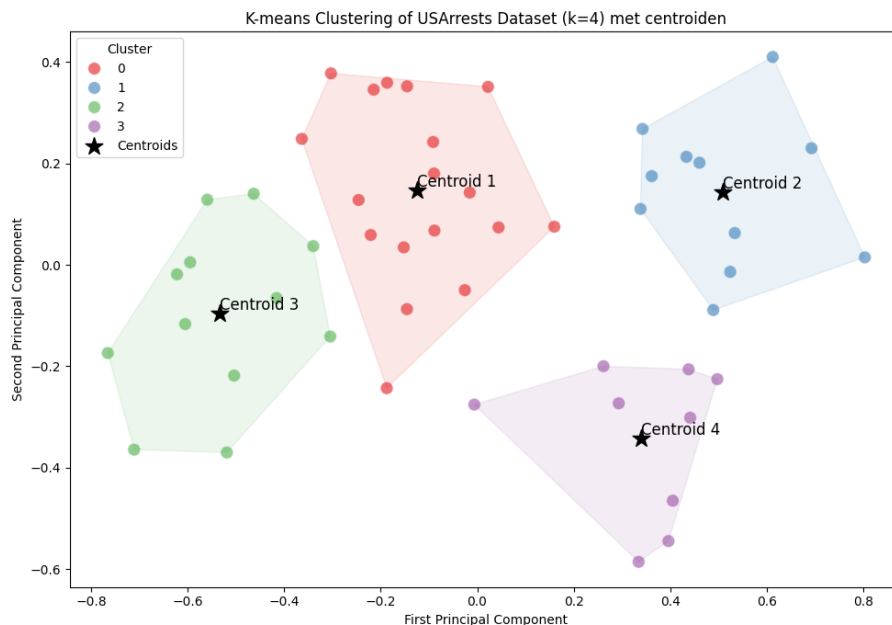


Figure 58: Initiële clustering USArrests dataset (k = 4) met centroiden

Vervolgens vinden de verplaatsingen plaats. Die verplaatsingen worden per stap gevisualiseerd in figuur 59. Uit die figuur blijkt dat de clusters nog compacter worden dan bij een cluster aantal van 2.

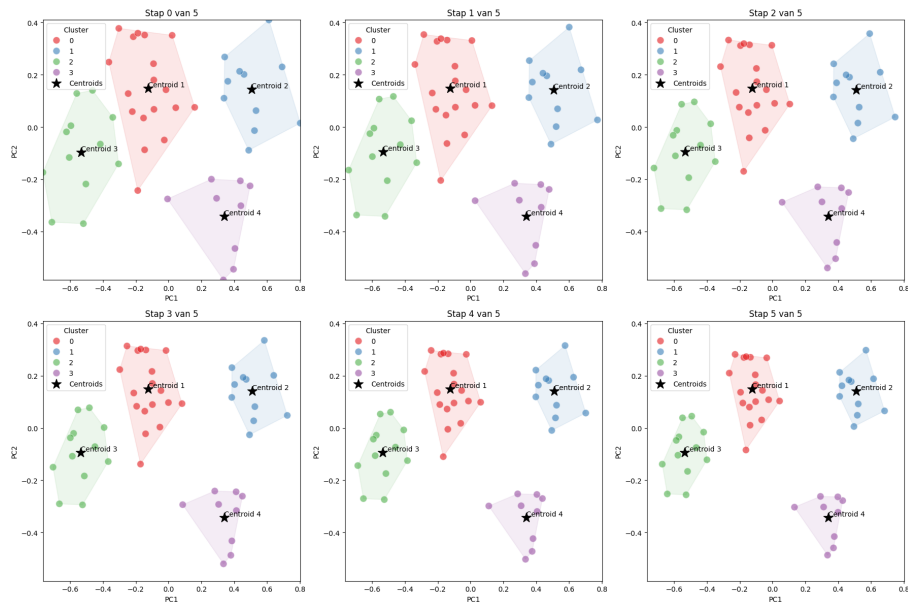


Figure 59: Clusters na verplaatsing van 0,1 grootte - USArrests dataset ($k = 4$)

Door de silhouette scores te berekenen, worden de volgende boxplots in figuur 60 gevisualiseerd. De waarden van de silhouette scores liggen lager dan bij een cluster-aantal van 2. Naast een lagere waarde vertonen de boxplots eenzelfde gedrag in de stijging. Uit de standaardafwijkingen in figuur 61 blijkt dat de clusters minder variatie hebben dan bij het cluster-aantal van 2. Verder is er ook een daling van de standaardafwijkingen af te lezen uit de figuur, naarmate de punten dichterbij hun centroiden komen. Ten slotte liggen de WSS-scores, weergegeven in figuur 62, lager dan bij het cluster-aantal van 2. Dit geeft aan dat bij cluster-aantal van 4 de clusters compacter zijn en er goed geclusterd is.

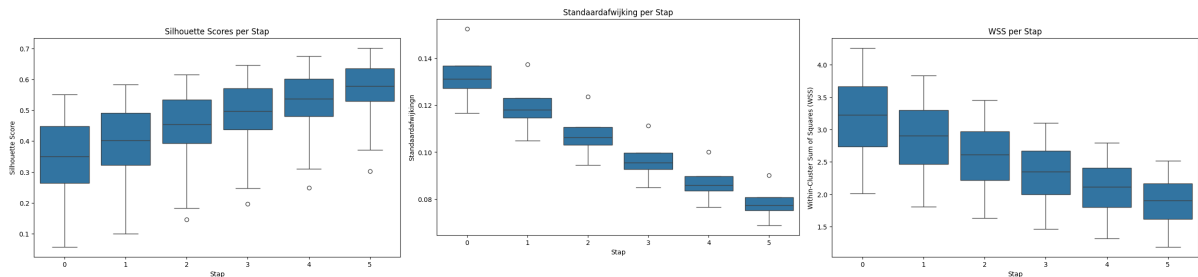


Figure 60:
Gemiddelde Silhouette ($k = 4$)

Figure 61:
Gemiddelde Standaardafwijking
($k = 4$)

Figure 62:
Gemiddelde WSS-scores ($k = 4$)

6 clusters

De cluster-aantallen worden voor een laatste keer toegenomen van 4 naar 6 clusters. In figuur 63 worden de 6 clusters weergegeven. Uit die figuur blijkt dat er clusters zijn die slechts één of twee datapunten bevatten in de clusters. Dit geeft aan dat de clusters kleiner en compacter zijn dan bij een lager cluster-aantal.

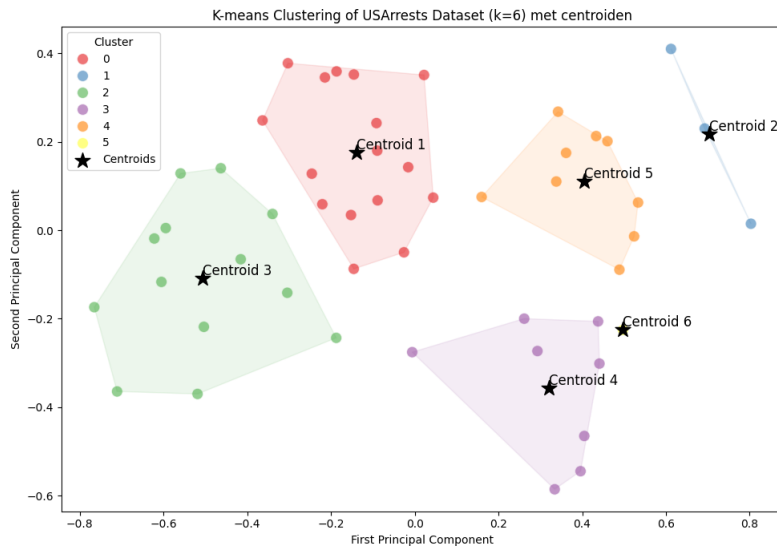


Figure 63: Initiële clustering USArrests dataset ($k = 6$) met centroiden

Vervolgens worden de clusterpunten in vijf stappen verplaatst naar hun dichtstbijzijnde centroiden en weergegeven in figuur 64. De verplaatsing geeft aan dat de clusters compacter worden.

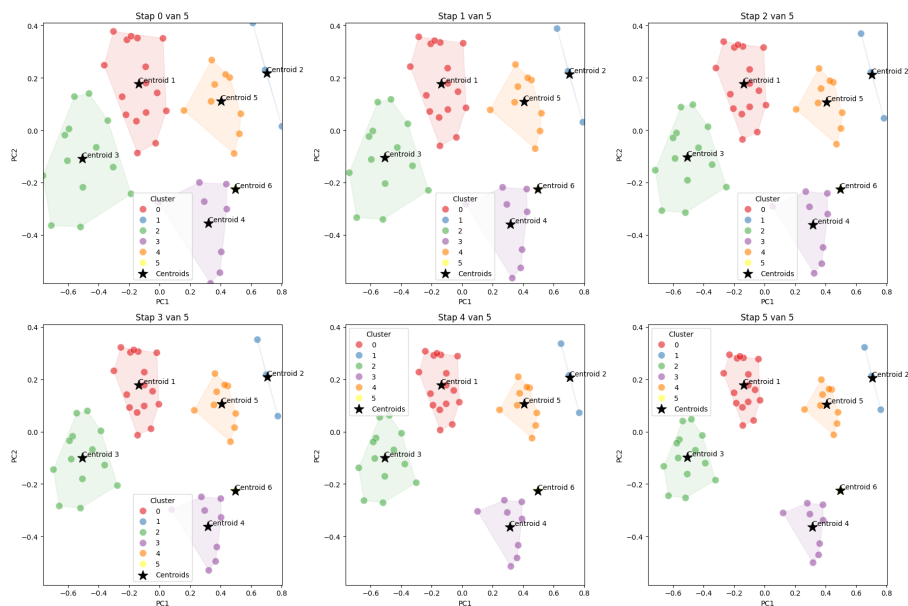


Figure 64: Clusters na verplaatsing van 0,1 grootte - USArrests dataset ($k = 6$)

Ten slotte worden de prestatie-indicatoren per stap berekend om meer informatie te verkrijgen over de clusters. Zo is er in figuur 65 te zien dat de gemiddelde silhouette scores per stap lager liggen dan bij een clusteraantal van 2, maar wel op eenzelfde niveau als bij een clusteraantal van 4. De datapunten worden bij een groter clusteraantal slechter gedefinieerd binnen een cluster. De boxplots gedragen zich wel op eenzelfde manier; er wordt een stijging van de scores gerealiseerd. Wat betreft de standaardafwijkingen: in figuur 66 worden de boxplots gevisualiseerd waarop een afnemende trend te herkennen is. Weliswaar liggen de waarden van de standaardafwijking lager dan tevoren, wat aangeeft dat er minder variatie binnen een cluster is. Voor de WSS-scores is er eenzelfde patroon te zien; de scores dalen naarmate de punten dichterbij hun centroiden komen. Bijkomend liggen de WSS-scores

lager dan bij de cluster aantallen 2 en 4 en is er dus sprake van een betere, compactere clustering.

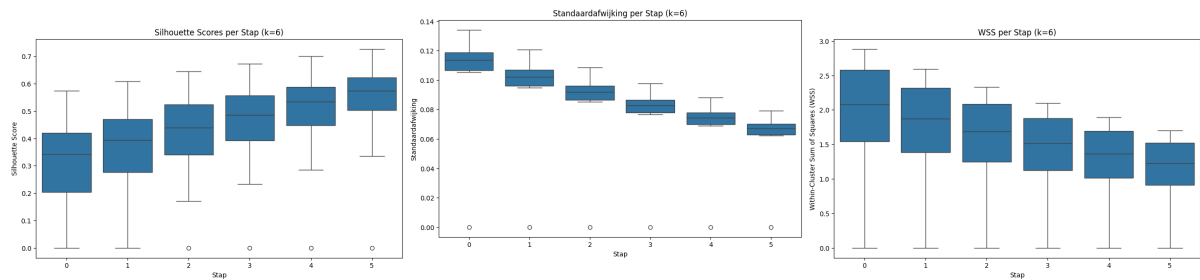


Figure 65:
Gemiddelde Silhouette (k = 6)

Figure 66:
Gemiddelde Standaardafwijking
(k = 6)

Figure 67:
Gemiddelde WSS-scores (k = 6)

5.2.2 Supervised K-Means

Een tweede methode die wordt toegepast op de USArrests dataset is de supervised K-Means clustering methode. Hierbij wordt de USArrests dataset opgesplitst in train- en testdata, respectievelijk 80% en 20%. Vervolgens wordt de traindata gebruikt om de stappen van unsupervised K-Means te herhalen. Het enige wat hier verschillend is, is dat de clustering gebeurt op basis van de klassen. Bij de unsupervised methode gebeurt de clustering willekeurig op de data. Voor deze methode wordt er ook gebruikgemaakt van de volgende clustercombinaties: 2, 4 en 6. Die hoeveelheden komen bij de supervised K-Means overeen met één, twee en drie clusters per klasse voor de USArrests dataset.

2 clusters

Na het opsplitsen van de data in train- en testset, wordt de clusteringmethode toegepast met een cluster aantal van 2. Hierbij wordt er dus per klasse één cluster aangemaakt. De clusters die uit die aanpak resulteren, worden in figuur 68 gevisualiseerd samen met hun centroiden. Uit de figuur blijkt dat de centroiden op ongeveer dezelfde plaats liggen als bij de unsupervised K-Means methode met een cluster aantal van 2. Hierdoor wordt er niet verder gegaan met dit cluster aantal.

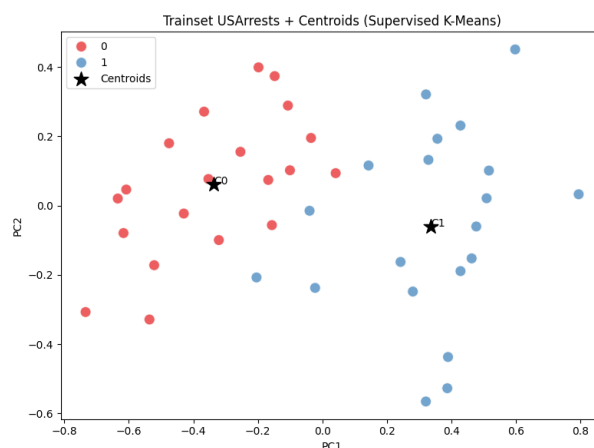


Figure 68: Initiële supervised clustering USArrests dataset (k = 2) met centroiden

4 clusters

Door het clusteraantal toe te nemen van 2 naar 4 is er in figuur 69 duidelijk te zien dat de centroiden op een andere plaats liggen dan bij de unsupervised methode. Er worden hier dus twee clusters per klasse gemaakt.

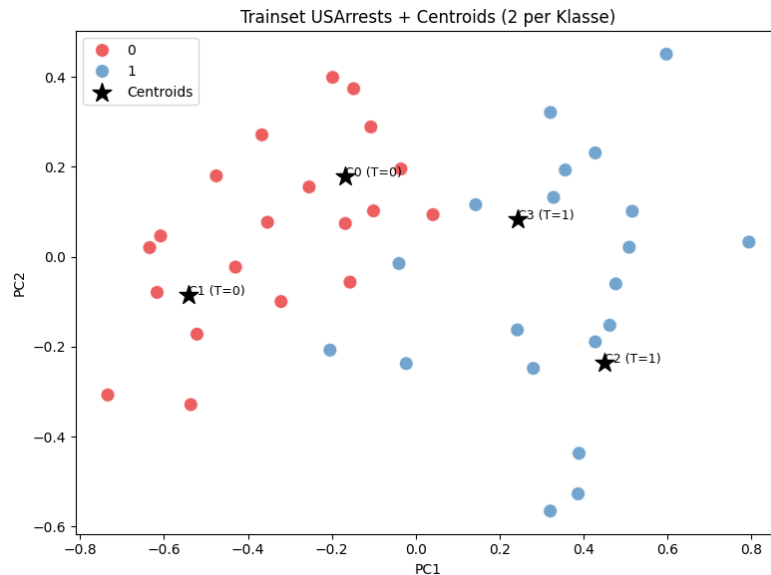


Figure 69: Initiële supervised clustering USArrests dataset ($k = 4$) met centroiden

Vervolgens worden de datapunten in vijf stappen met telkens 0,1 grootte verplaatst naar hun dichtstbijzijnde centroiden. Elke stap van die verplaatsing wordt gevisualiseerd in figuur 70. Vergeleken met de clusters van de unsupervised methode, zijn er kleinere clusters gevormd en liggen de punten dichter bij hun centroiden na de vijfde stap.

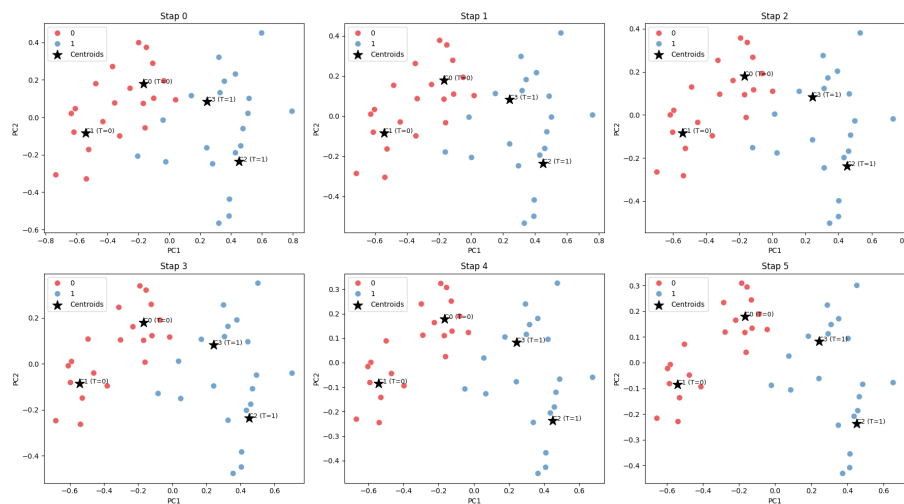


Figure 70: Supervised clusters na verplaatsing van 0,1 grootte - USArrests dataset ($k = 4$)

Ten slotte worden voor elke stap de prestatie-indicatoren berekend. De silhouette scores, weergegeven in figuur 71, kennen net als bij de unsupervised methode een stijging per stap. Weliswaar is de stijging bij de supervised K-Means veel lichter. Ook valt er op dat de gemiddelde waarden van de silhouette

scores hoger liggen dan bij de unsupervised methode. De boxplots van de standaardafwijkingen vertonen in figuur 72 ook een gelijkaardig patroon als bij de unsupervised methode. Echter liggen de waarden hier hoger, wat aangeeft dat er meer variatie is binnen een cluster. In figuur 73 worden de WSS-scores weergegeven. Ook die scores kennen een daling, net als bij de unsupervised methode. Ook liggen hier de gemiddelde waarden van de WSS-scores hoger dan bij de unsupervised methode.

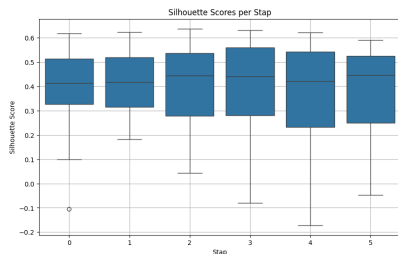


Figure 71:
Gemiddelde Silhouette ($k = 4$)

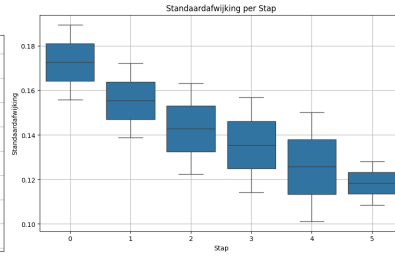


Figure 72:
Gemiddelde Standaardafwijking
($k = 4$)

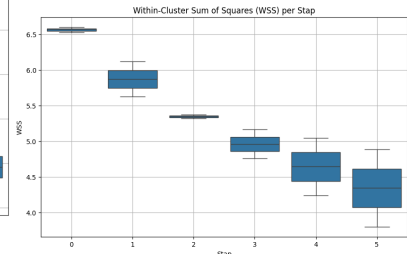


Figure 73:
Gemiddelde WSS-scores ($k = 4$)

6 clusters

Ten slotte wordt het clusteraantal een laatste keer toegenomen van 4 naar 74. De zes clusters worden in figuur gevisualiseerd. Uit de figuur blijkt dat de ligging van de centroiden een klein beetje verschilt van de unsupervised methode. Er wordt hier dus wel verder gewerkt met dit clusteraantal.

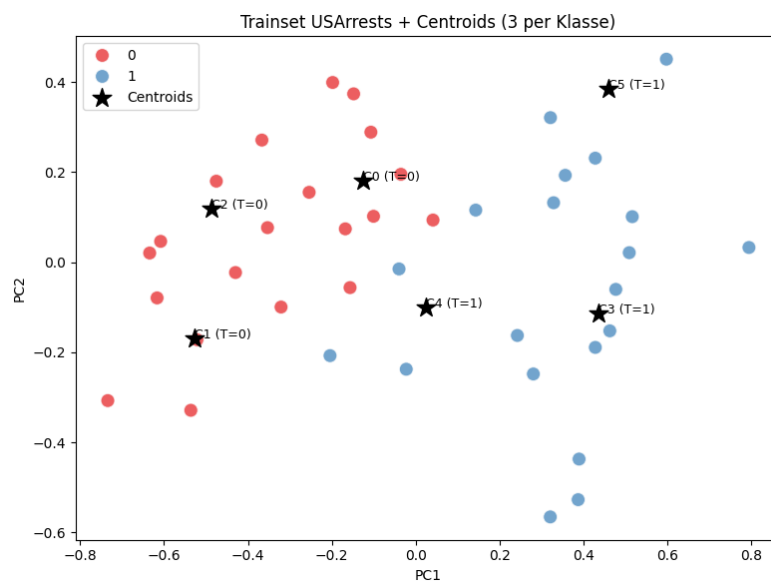


Figure 74: Initiële supervised clustering USArrests dataset ($k = 6$) met centroiden

Nadat de clusters gevisualiseerd zijn, worden de datapunten van de clusters in vijf even grote stappen verplaatst naar de dichtstbijzijnde centroiden. In figuur 75 is er niet veel informatie af te leiden, buiten het feit dat de clusters weinig datapunten bevatten.

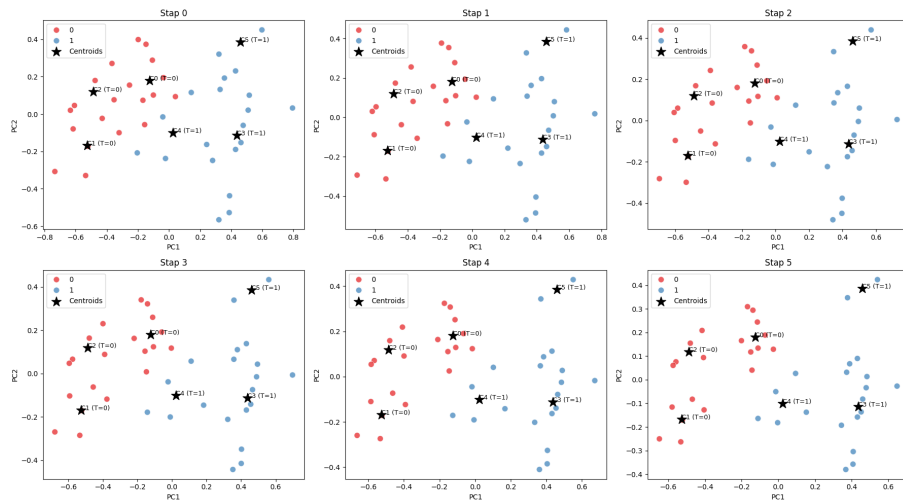


Figure 75: Supervised clusters na verplaatsing van 0,1 grootte - USArrests ($k = 6$)

Om meer over de clusters te kunnen zeggen, worden de prestatie-indicatoren per stap berekend en gevisualiseerd in de vorm van boxplots. In figuur 76 worden de silhouette scores getoond. Uit die figuur blijkt dat de silhouette scores ongeveer hetzelfde blijven voor de stappen. Ook ligt de silhouette score van stap 0 ongeveer op hetzelfde niveau als bij de unsupervised methode. Verder liggen de gemiddelde waarden van de standaardafwijkingen, weergegeven in figuur 77, op een iets hoger niveau. Ook is er hier een dalende trend in de standaardafwijking herkenbaar. Ten slotte worden de WSS-scores in figuur 78 gevisualiseerd. De gemiddelde WSS-scores liggen bij de unsupervised methode lager dan bij de supervised methode, wat aangeeft dat de clusters bij de unsupervised methode compacter zijn. Echter vertonen beide methoden een daling in de WSS naarmate de punten dichter bij hun centroiden liggen.

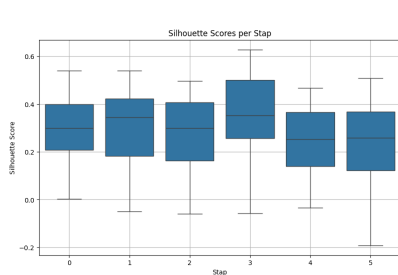


Figure 76:
Gemiddelde Silhouette ($k = 6$)

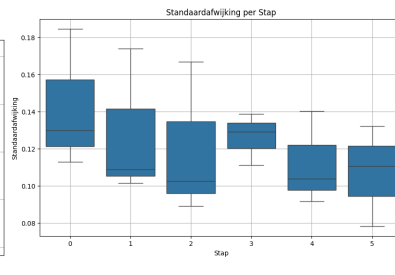


Figure 77:
Gemiddelde Standaardafwijking
($k = 6$)

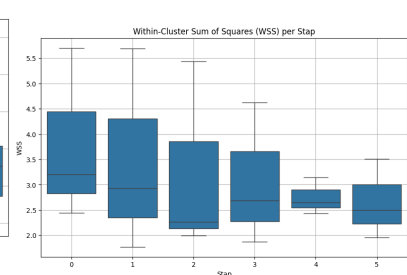


Figure 78:
Gemiddelde WSS-scores ($k = 6$)

5.2.3 KNN

Na het afronden van de supervised K-Means clusteringexperimenten, wordt er verder gegaan met de KNN-algoritmen. Eerst en vooral wordt er voor de gehele USArrests dataset het beste aantal burens (k) bepaald. In figuur 79 wordt een overzicht gegeven van de accuracies van verschillende aantallen burens binnen het bereik 1 tot en met 20. Uit de grafiek blijkt dat de eerste beste k gelijk is aan 4, met een accuracy van 80%. Een veel te grote k wordt vermeden, omwille van het underfitten van het model.

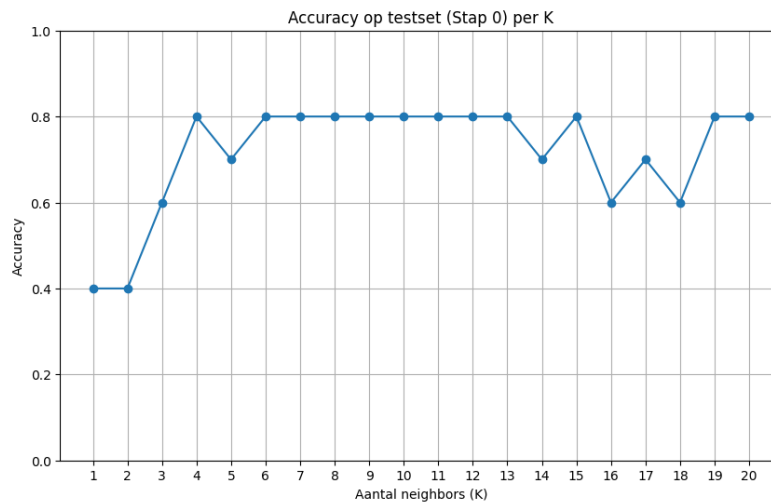


Figure 79: KNN accuracy per k

Nadat het beste aantal burens bepaald is, wordt de verplaatsing van de datapunten voor het KNN-algoritme weergegeven in figuur 80. Naarmate de datapunten naar hun dichtstbijzijnde centroiden verplaatsen, is er een kleine wijziging te zien aan de scheidslijn.

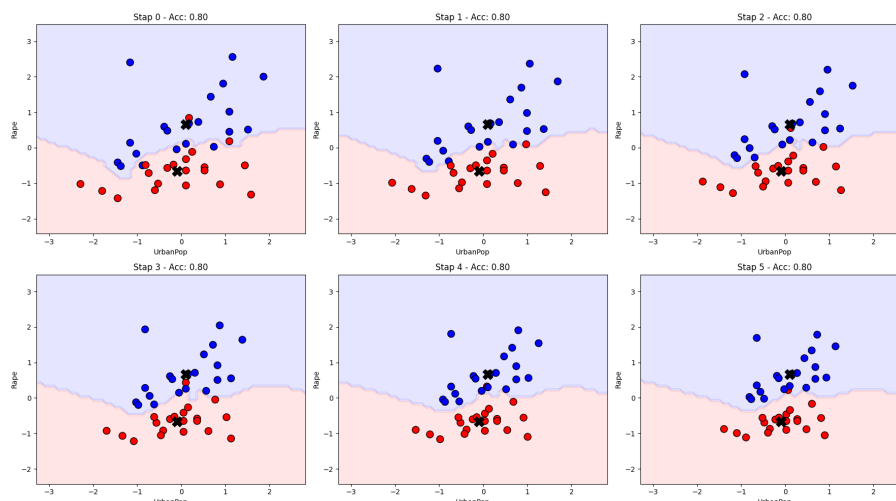


Figure 80: KNN's van alle stappen by verplaatsing van 0,1 - USArrests dataset ($k = 4$)

Ten slotte wordt er gekeken naar de accuraatheid van het model. Door de verplaatsing van de datapunten is er noch een stijging noch een daling in de accuracies. In figuur 81 worden de accuracies van 10 stappen weergegeven, omdat er na 5 stappen geen echt effect te zien is. De accuracy van het model bedraagt voor alle stappen 80%.

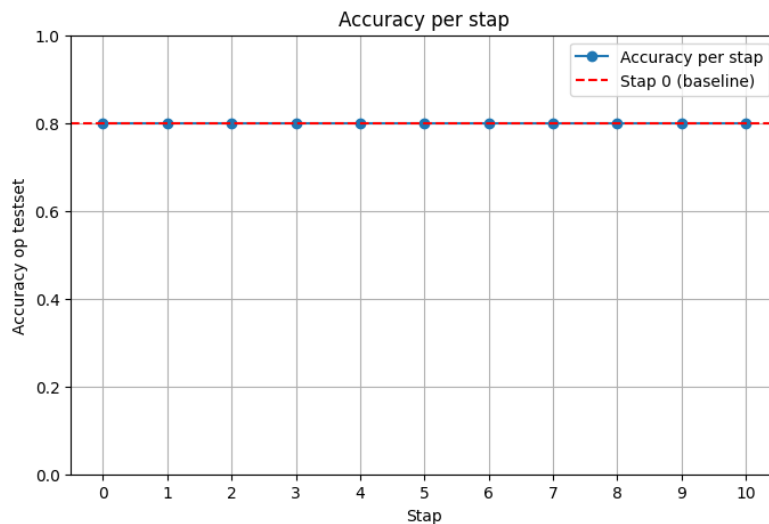


Figure 81: KNN accuracy (10 stappen)

5.2.4 KNN toegepast op supervised K-Means

Het KNN-algoritme wordt herhaald en toegepast op de supervised K-Means clusteringmethode, waarbij de traindata die verkregen wordt na de train-test-split wordt gebruikt. Ook hier wordt hetzelfde aantal burens gebruikt, namelijk 4, om later de resultaten met de gewone KNN te vergelijken.

In figuur 82 worden de stappen van verplaatsing voor het KNN-algoritme toegepast op de supervised K-Means methode weergegeven. In die figuur is er duidelijker te zien dat de scheidingslijn verandert en dat de afstand tussen beide klassen groter wordt. De scheidingslijn wordt namelijk lineair.

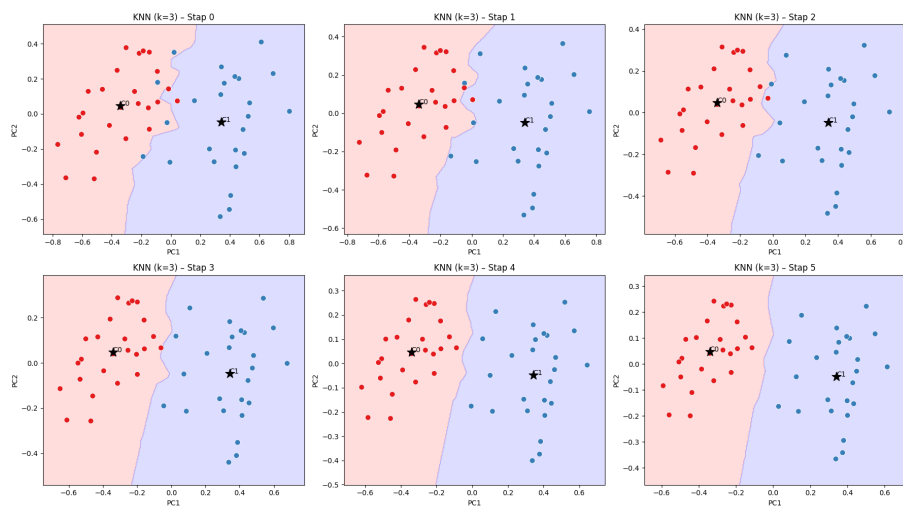


Figure 82: KNN's van alle stappen toegepast op de supervised K-Means bij verplaatsing van 0,1 - USArrests dataset ($k = 4$)

Om daadwerkelijk het effect van de verplaatsing te zien, worden de accuracies van de vijf stappen weergegeven in figuur 83. Voor de vijf stappen kent de testaccuracy noch een stijging noch een daling. De waarde van die testaccuracy blijft constant behouden op 80%. De trainaccuracy daarentegen kent een stijging vanaf stap 2 en bereikt een waarde van 100%. Dit geeft aan dat het model de traindata perfect leert. Door de stappen van verplaatsing toe te nemen naar 10 worden de accuracies weergegeven

in figuur 84. Uit die figuur blijkt dat de testaccuracy op 80% blijft en de trainaccuracy op 100%.

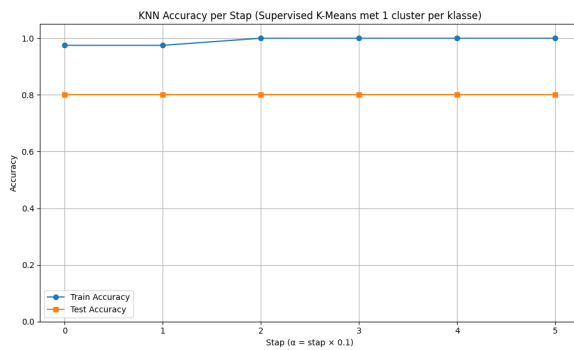


Figure 83: Supervised KNN accuracy (5 stappen)

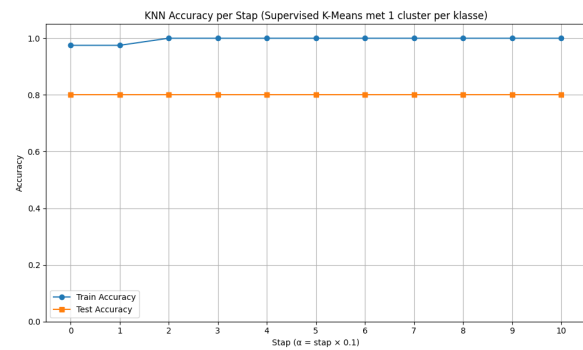


Figure 84: Supervised KNN accuracy (10 stappen)

5.2.5 Beslissingsbomen

Een laatste methode die op de USArrests dataset wordt toegepast, zijn de beslissingsbomen. In figuur 85 worden de drie beslissingsbomen weergegeven voor de originele dataset zonder verplaatsing. De structuur en het aantal splitsingen van de 3 bomen zijn hetzelfde. Het enige wat verschilt, is dat bij de eerste boom de splitsing gebeurt op basis van de attribuutwaarde *Assault* kleiner of gelijk aan 0,443. Terwijl de andere twee bomen op basis van dezelfde attribuut splitsen, maar dan voor een hogere waarde, namelijk 0,473.

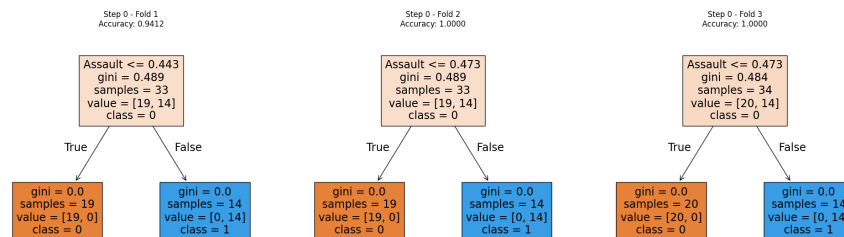


Figure 85: Initiële beslissingsbomen - USArrests dataset

Door de verplaatsingen van de datapunten naar hun dichtstbijzijnde centroiden door te voeren, zijn er kleine wijzigingen op de bomen, weergegeven in figuur 86, te achterhalen. Zo is er te zien dat de waarden dalen waarop de beslissingen gebeuren. Ook daalt het verschil tussen boom 1 en 2; in de originele stap bedroeg het verschil ongeveer 0,030. Nu bedraagt het verschil 0,020 en wordt steeds kleiner naarmate de punten dichterbij hun centroiden liggen. Vanaf stap 4 komt er voor zowel de tweede als de derde boom twee extra splitsingen bij. De tweede splitsing gebeurt op basis van hetzelfde attribuut en waarde, maar de derde splitsing gebeurt enkel op basis van hetzelfde attribuut. Hoewel de waarden van de derde splitsing ook afnemen naarmate de stappen vorderen.

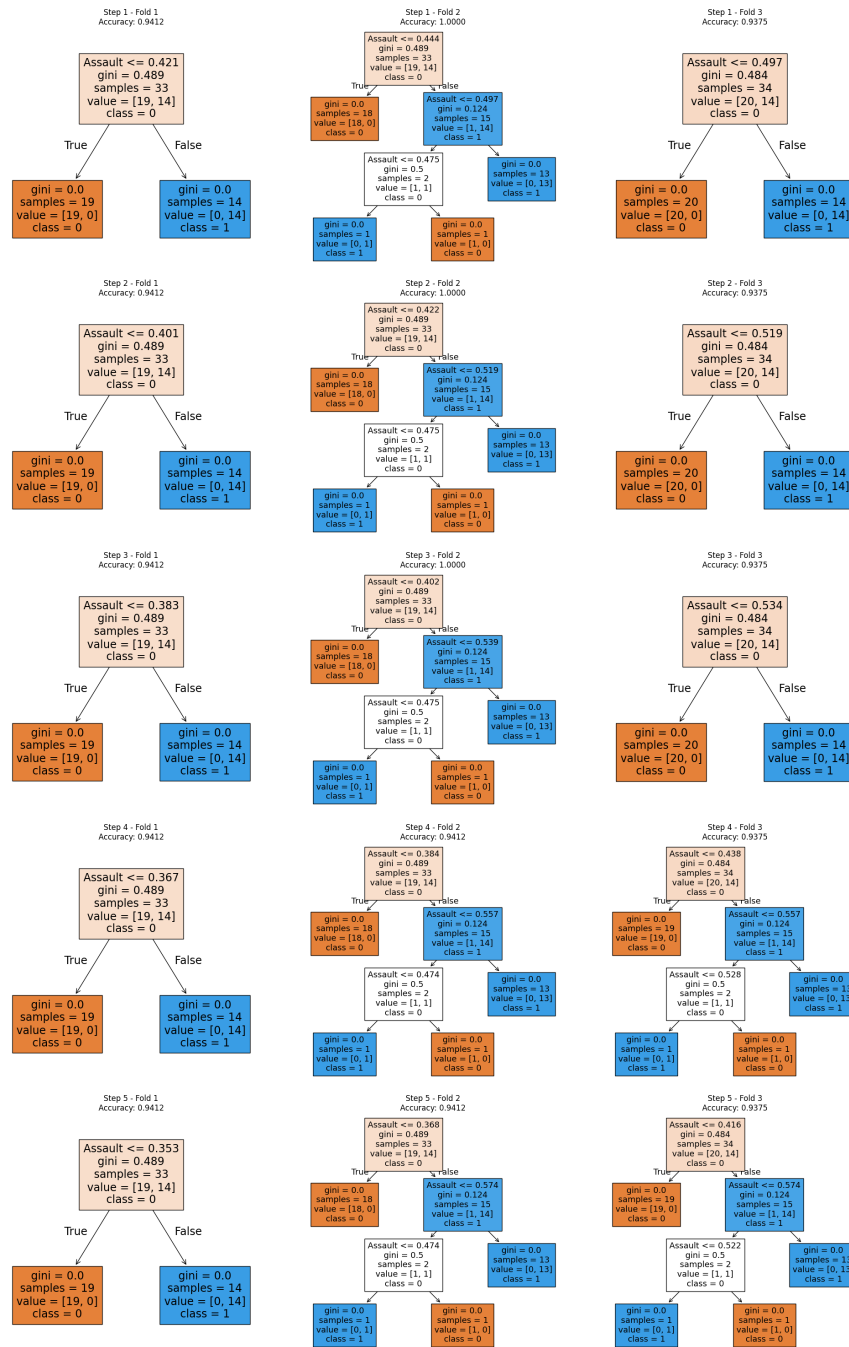


Figure 86: Beslissingsbomen per stap van verplaatsing - USArrests dataset

Door naar de accuracies te kijken, die weergegeven worden in figuur 87, is er duidelijk te zien dat er een kleine wijziging is in de gemiddelde accuraatheid van de bomen. De oorspronkelijke gemiddelde bedroeg 98%. Na een verplaatsing van één stap kende de accuraatheid van de bomen een daling naar 96%. Verder bereikten in stap 4 en 5 de bomen een minimum met 94%, maar na stap 5 begon de accuracy terug te stijgen en bleef constant, met een waarde van 96% voor de volgende stappen tot en met de tiende stap. In het algemeen is er dus een zeer kleine wijziging in de accuraatheid van de bomen te zien na de verplaatsingen.

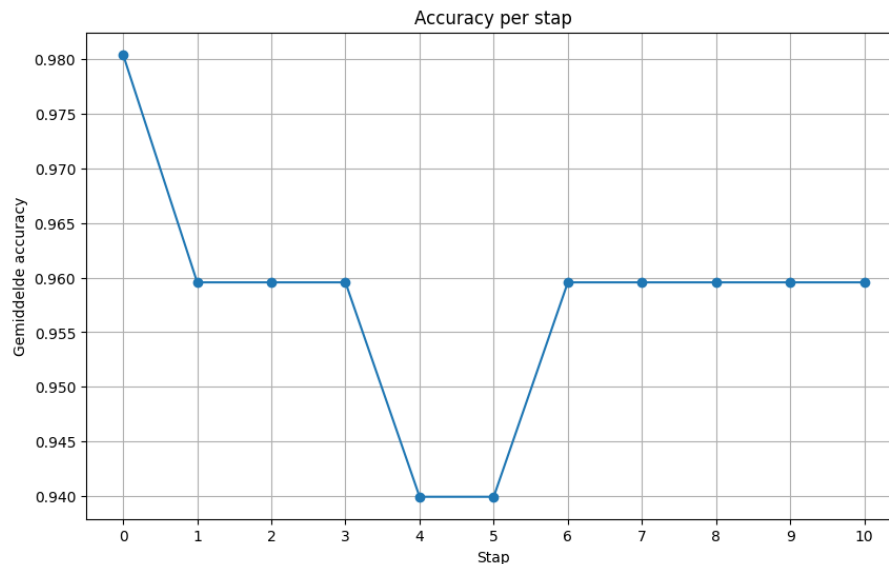


Figure 87: Gemiddelde accuracy beslissingsbomen (10 stappen)

5.2.6 Beslissingsbomen toegepast op supervised K-Means

Door de train-test-split toe te passen op de USArrests, wordt er een trainset aangemaakt van 80% van de data. Vervolgens wordt die trainset gebruikt om aan supervised K-Means te doen, waarbij er per klasse één cluster wordt aangemaakt. Die clusteringmethode wordt als basis gebruikt en daarop worden de beslissingsbomen toegepast. In figuur 88 worden bomen weergegeven die uit de drievoudige crossvalidatie resulteren. Het aantal splitsingen per fold verschilt. Tussen de drie bomen is er een verschil in de waarde waarop de eerste splitsing gebeurt. Voor de eerste boom gebeurt het door een smallere beslissingswaarde, wat aangeeft dat de splitsing effectiever is. De voorwaarden van de andere twee bomen liggen korter bij elkaar, maar zijn breder dan de eerste boom.

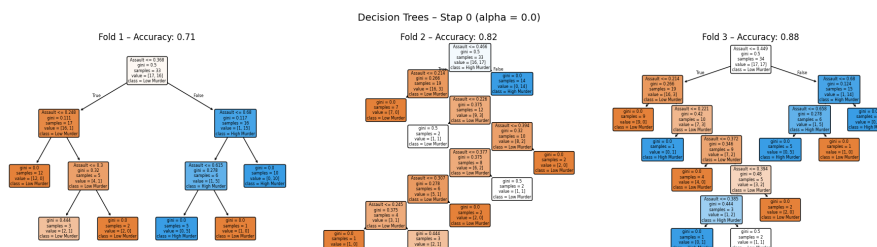


Figure 88: Initiële supervised beslissingsbomen - USArrests dataset

Een verplaatsing van de datapunten naar hun dichtstbijzijnde centroiden zorgt ervoor dat het aantal splitsingen afneemt. Ook worden de splitsingen smaller en effectiever. Vanaf stap 4 is er duidelijk te zien dat de eerste splitsing die op basis van het attribuut *Assault* gebeurt, ongeveer dezelfde beslissingswaarde heeft voor alle drie de bomen. Het verschil is hier heel klein en bedraagt enkel 0,016.

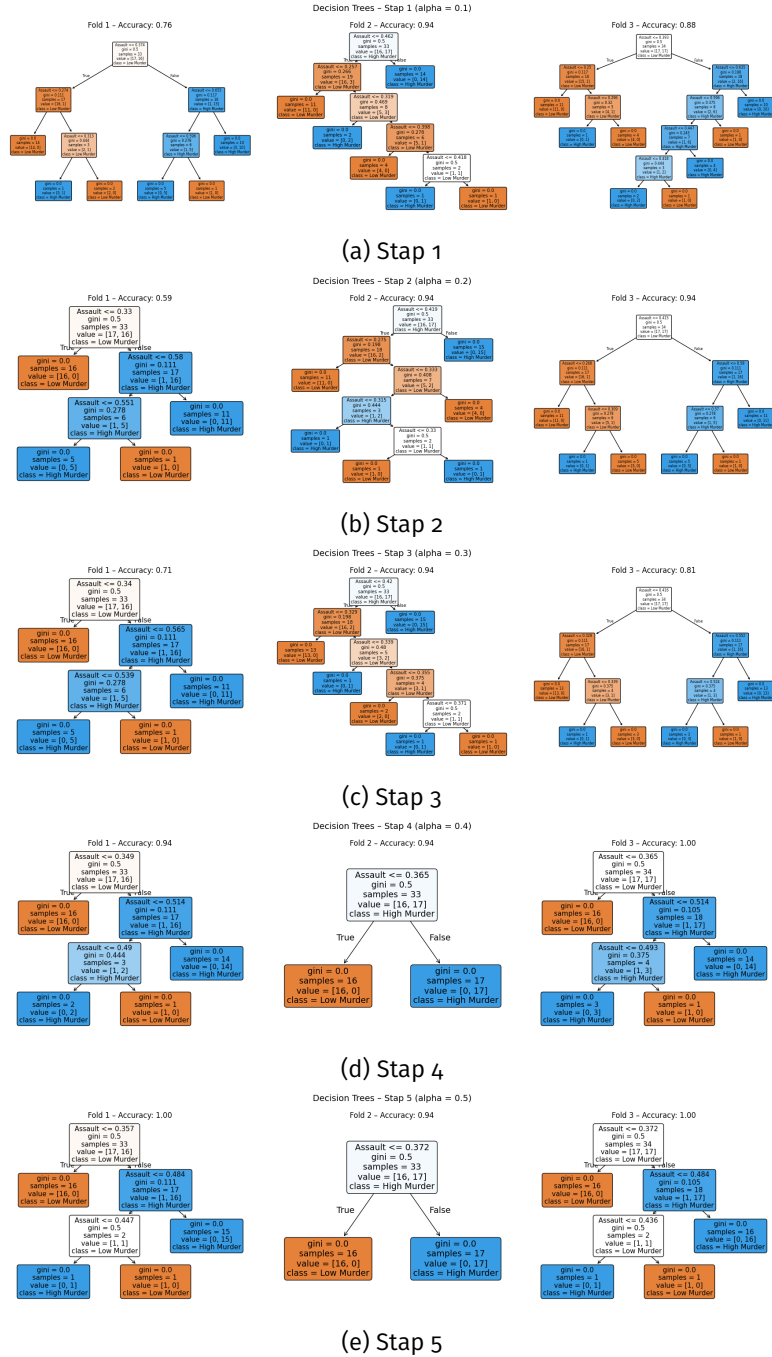


Figure 89: Supervised beslissingsbomen per stap van verplaatsing - USArrests dataset

In figuur 90 worden de accuracies van alle stappen weergegeven. Zowel de train- als testaccuracies worden weergegeven in de figuur. In het begin liggen de testaccuracies rond de 70%. De trainaccuracies stijgen na de eerste stap onmiddellijk naar 100%, dus leert het model perfect de traindata. Na stap 3 is er een stijging in de testaccuracy te zien en stijgt de accuracy van de bomen naar 80%. Die waarde blijft constant behouden voor de overige stappen.

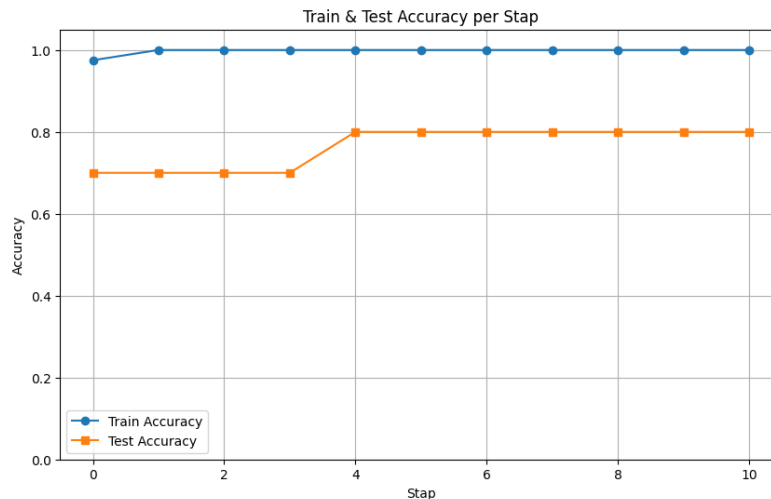


Figure 90: Gemiddelde accuracy supervised beslissingsbomen (10 stappen)

5.3 Wine

Een derde dataset die voor de experimenten wordt gebruikt, is de built-in Wine dataset. De Wine dataset bestaat uit 178 observaties van wijnmonsters, waarbij 13 continue numerieke variabelen van de chemische eigenschappen. Vanwege een te groot aantal variabelen worden in Python de beste features geselecteerd. In ons geval zijn dit de attributen *Proline* en *Flavanoids*. *Proline* is een aminozuur en *Flavanoids* zijn een groep natuurlijke antioxidanten. Beide variabelen helpen om de wijnsoorten van elkaar te onderscheiden. De doelvariabelen van de Wine dataset bestaan uit 3 klassen van druivensoorten die in de wijnmonsters worden gebruikt ("load_wine", 2025).

5.3.1 Unsupervised K-Means

Aan de hand van de elbow- en silhouettemethode wordt het optimale cluster aantal voor de Wine dataset bepaald om aan K-Means clustering te doen. Hiervoor worden beide methoden weergegeven in figuur 91. Uit die grafieken blijkt dat de optimale keuze voor de Wine dataset bij een cluster aantal van 3 ligt. Voor de Wine dataset wordt er dus verder gewerkt met de volgende clustercombinaties: 3, 6 en 9.

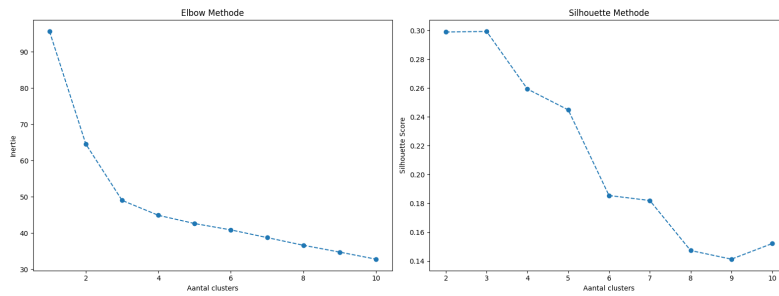


Figure 91: Elbow- en Silhouettemethode - Wine dataset

3 clusters

Allereerst worden de drie clusters samen met hun centroiden weergegeven in figuur 92. Uit de figuur blijkt dat de drie clusters evenveel datapunten bevatten. De spreiding van die punten ziet er voor de rest ook hetzelfde uit voor de clusters.

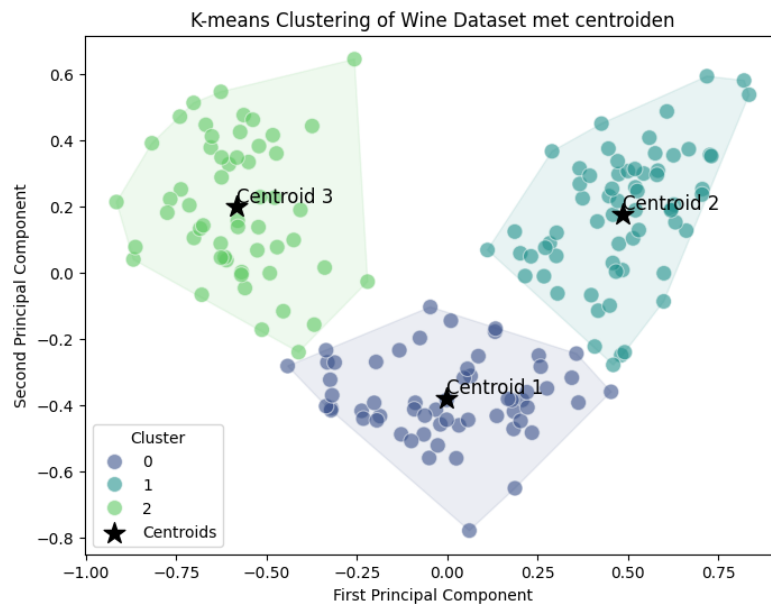


Figure 92: Initiële clustering Wine dataset ($k = 3$)

Vervolgens worden de datapunten naar hun dichtstbijzijnde centroiden verplaatst. De verplaatsingen worden stapsgewijs weergegeven in figuur 93. Op het eerste gezicht kan geconcludeerd worden dat de clusters compacter worden en dat de spreiding van de punten afneemt. Om dit te bevestigen, worden de prestatie-indicatoren berekend.

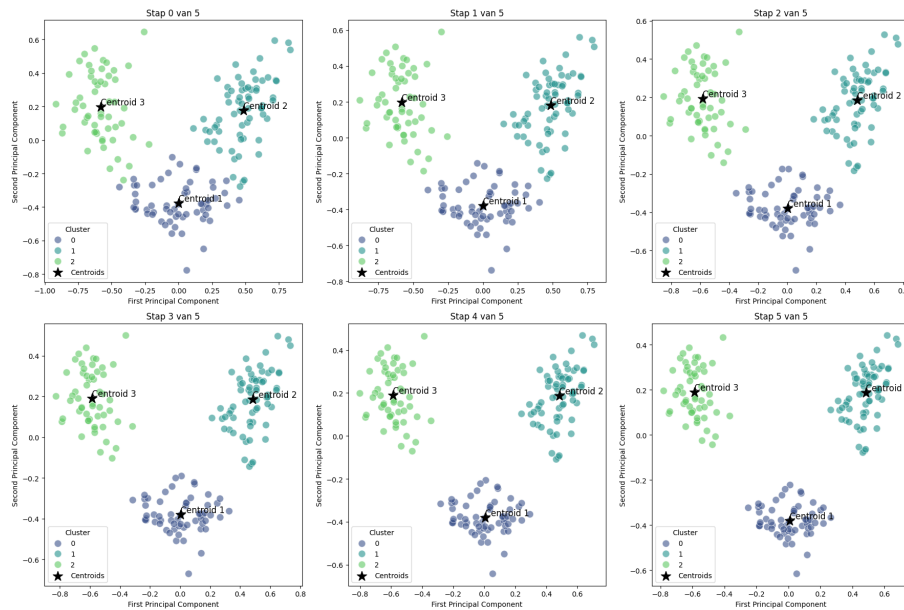


Figure 93: Clusters na verplaatsing van 0,1 grootte - Wine dataset ($k = 3$)

Uit de elbow- en silhouettemethode in figuur 91 was het al duidelijk dat er voor een clusteraantal van 3 een lage silhouette score mee gepaard ging. De silhouette scores in figuur 94 geven ook aan dat de waarden laag zijn, maar de waarden kennen een stijging naarmate de stappen vorderen. Dit geeft aan dat de datapunten beter gedefinieerd worden. Verder neemt, zoals eerder vermeld, de spreiding van de datapunten af. Dit komt overeen met een daling in de standaardafwijking, wat af te leiden is uit figuur 95. De WSS-scores weergegeven in figuur 96 hebben een dalende trend, wat aangeeft dat de clusters compacter worden en leidt tot goede clusterings.

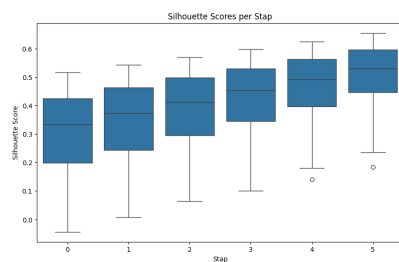


Figure 94:
Gemiddelde Silhouette ($k = 3$)

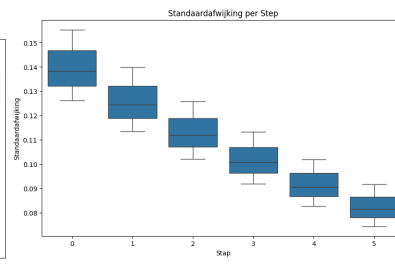


Figure 95:
Gemiddelde Standaardafwijking
($k = 3$)

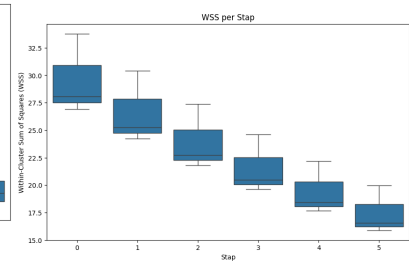


Figure 96:
Gemiddelde WSS-scores ($k = 3$)

6 clusters

Het clusteraantal wordt vervolgens toegenomen van 3 clusters naar 6 clusters. Die 6 clusters worden in figuur 97 gevisualiseerd samen met hun centroiden. Er is duidelijk te zien dat de clusters elkaar overlappen.

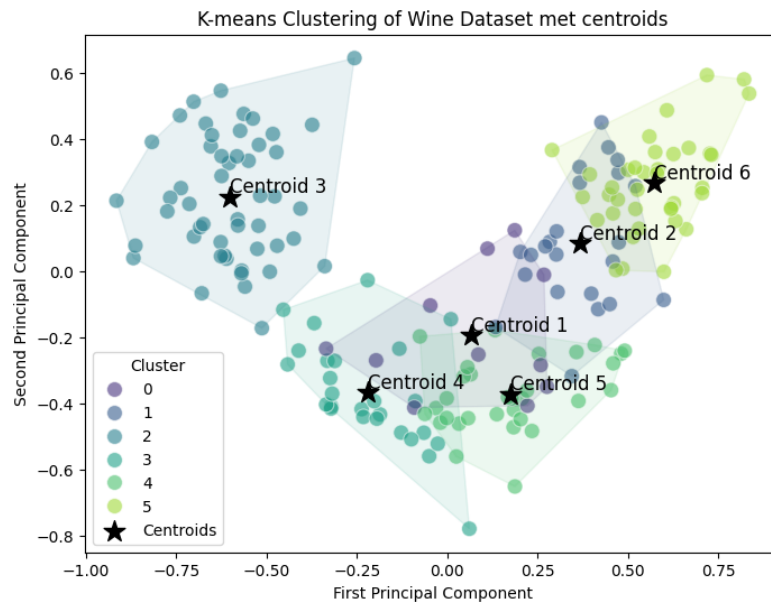


Figure 97: Initiële clustering Wine dataset (k = 6)

De datapunten worden in vijf stappen met telkens een grootte van 0,1 verplaatst in de richting van hun dichtstbijzijnde centroiden. Naarmate de stappen vorderen, is er duidelijk in figuur 98 te zien dat de overlappings verminderen. Ieder cluster krijgt beter zijn eigen vorm.

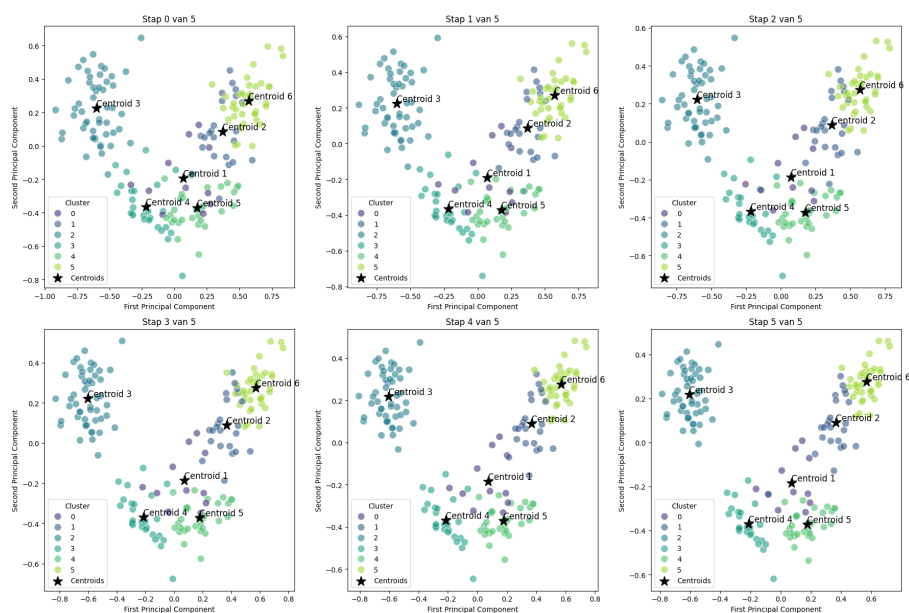


Figure 98: Clusters na verplaatsing van 0,1 grootte - Wine dataset (k = 6)

Ten slotte worden de prestatie-indicatoren berekend voor elke stap. Het gemiddelde silhouette score per stap heeft een lagere waarde dan bij een clusteraantal van 3. Ook is er in figuur 99 te zien dat de silhouette scores toenemen per stap. De datapunten worden beter gedefinieerd in de clusters zelf, maar nog steeds slechter dan bij 3 clusters. In figuur 100 worden de standaardafwijkingen in de vorm van boxplots weergegeven. De waarden van de standaardafwijkingen liggen bij 6 clusters lager dan bij 3, wat aangeeft dat er bij 6 clusters minder spreiding is binnen een cluster. In figuur 101 worden de

WSS-scores per stap weergegeven. De gemiddelde WSS-scores liggen bij 6 clusters veel lager dan bij 3. De clusters bij het clusteraantal van 6 zijn dus compacter, wat ook te zien was in figuur 94.

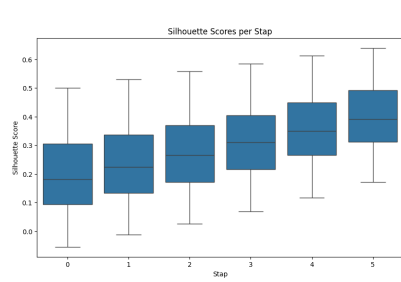


Figure 99:
Gemiddelde Silhouette (k = 6)

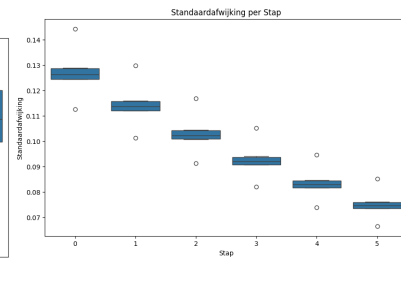


Figure 100:
Gemiddelde Standaardafwijking
(k = 6)

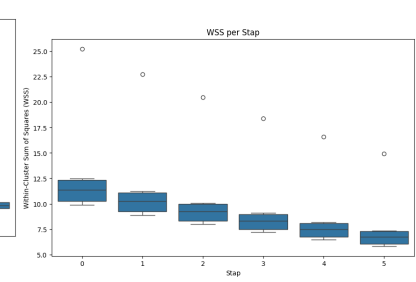


Figure 101:
Gemiddelde WSS-scores (k = 6)

9 clusters

In figuur 102 worden de clusters met hun centroiden weergegeven van de Wine dataset waarbij het clusteraantal is toegenomen tot 9. Uit de figuur is er duidelijk te zien dat er alleen maar overlappende clusters zijn. De overlappings geven aan dat de clusters eerder niet goed gedefinieerd zijn.

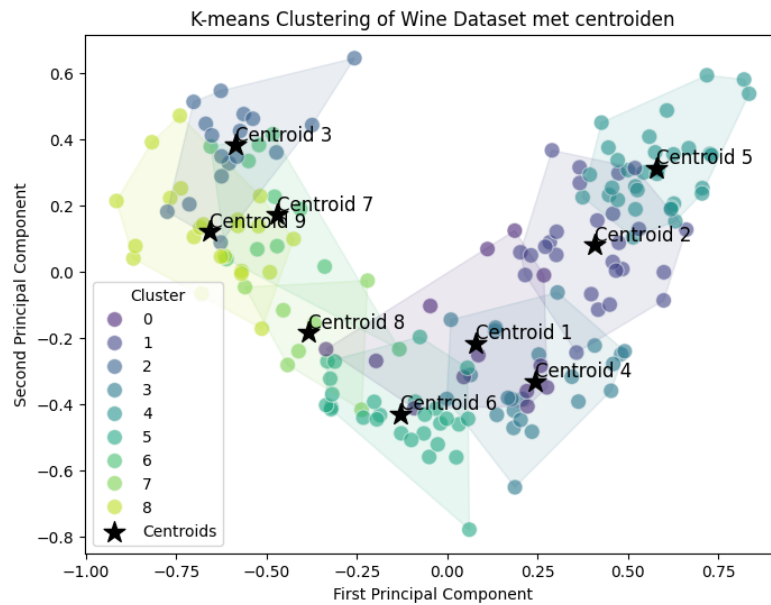


Figure 102: Initiële clustering Wine dataset ($k = 9$) met centroiden

Het verplaatsen van de datapunten in vijf evengrote stappen wordt weergegeven in figuur 103. De verplaatsing zorgt ervoor dat de overlappings rustig aan verdwijnen. De clusters kunnen beter geïdentificeerd worden. Echter, het is moeilijk om uit deze figuur conclusies te trekken.

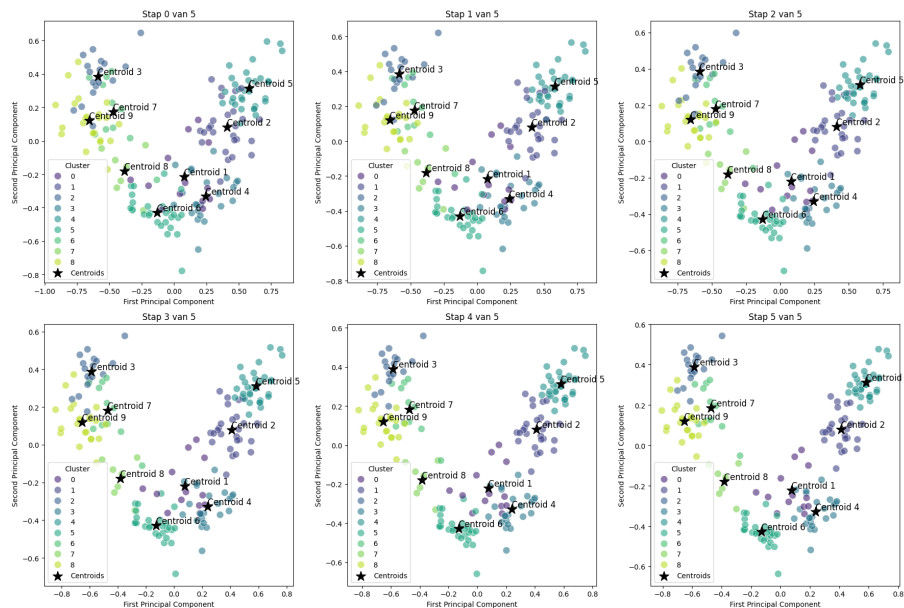


Figure 103: Clusters na verplaatsing van 0,1 grootte - Wine dataset ($k = 9$)

Daaruitvolgend worden de prestatie-indicatoren berekend van de stappen. De silhouette scores per

stap, weergegeven in figuur 104, geven aan dat de clusters niet zo goed de datapunten definiëren. De gemiddelde waarden bij 9 clusters liggen zelfs lager dan bij 6 clusters. Naarmate de stappen vorderen, vertonen de silhouette scores een stijgend patroon. De datapunten binnen een cluster zijn minder verspreid in vergelijking met 6 clusters. Dit is te achterhalen door naar de gemiddelde waarden van de standaardafwijkingen te kijken, weergegeven in figuur 105. Ook is er op de figuur te zien dat er een daling is van de standaardafwijkingen naarmate de datapunten de centroiden naderen. Uit voorgaande clustervisualisaties was te zien dat de clusters kleiner en compacter zijn. Dit is te bevestigen met de WSS-scores weergegeven in figuur 106. De gemiddelde WSS-scores liggen lager dan bij een cluster aantal van 6. Verder is er ook een daling te zien naarmate de datapunten in de richting van hun centroiden verplaatsen.

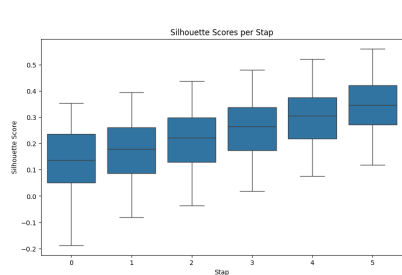


Figure 104:
Gemiddelde Silhouette ($k = 9$)

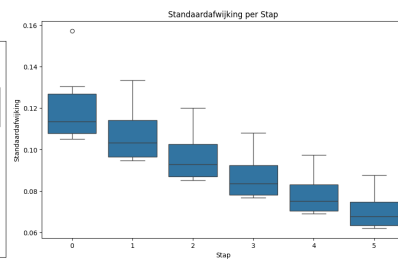


Figure 105:
Gemiddelde Standaardafwijking
($k = 9$)

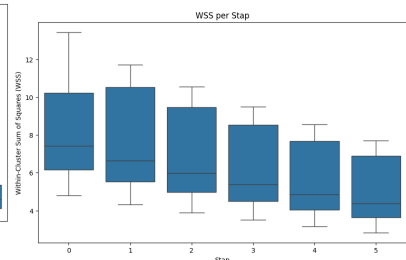


Figure 106:
Gemiddelde WSS-scores ($k = 9$)

5.3.2 Supervised K-Means

Na het afronden van de unsupervised K-Means clusteringmethode, wordt de Wine dataset opgesplitst in train- en testdata, respectievelijk 80% en 20% van de oorspronkelijke dataset. Na het opsplitsen van de data wordt er verdergegaan met de supervised K-Means clusteringmethode. Er worden dus per klasse één, twee en drie clusters gevormd. De clustercombinaties resulteren voor de Wine dataset die drie klassen bevat in de volgende: 3, 6 en 9.

3 clusters

Initieel wordt er in figuur 107 de drie clusters gevisualiseerd met hun centroiden. Uit de figuur blijkt dat de positionering van de centroiden op een gelijkaardige plaats ligt als bij de unsupervised methode. Hierdoor wordt er niet verder gewerkt met dit cluster aantal, want dezelfde resultaten worden behaald, maar dan voor een kleinere dataset (de traintdata).

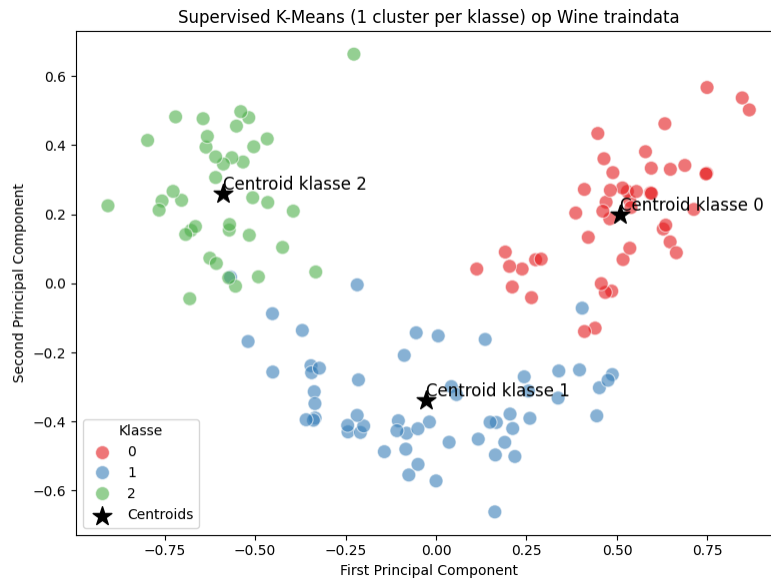


Figure 107: Initiële supervised clustering Wine dataset ($k = 3$) met centroiden

6 clusters

Vervolgens wordt het clusteraantal toegenomen van 3 naar 6. De clusters worden in figuur 108 gevisualiseerd. Uit de figuur is duidelijk af te leiden dat de centroiden op een andere positie liggen dan bij de unsupervised methode.

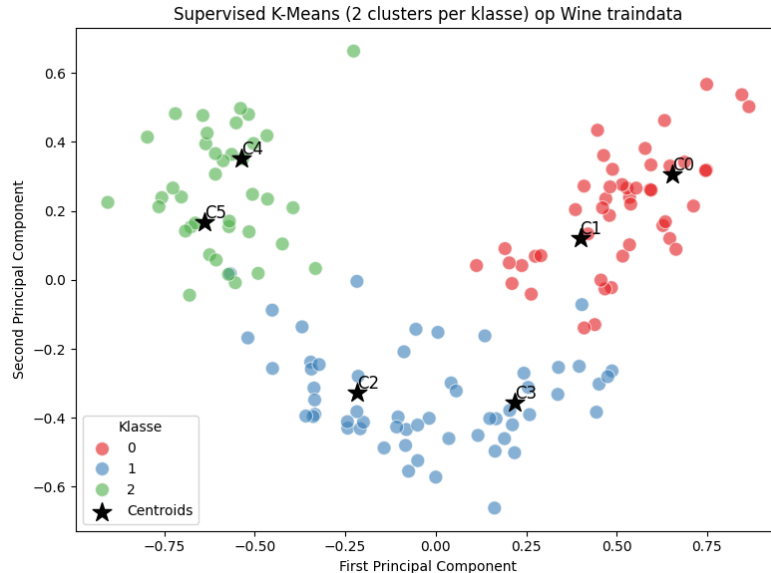


Figure 108: Initiële supervised clustering Wine dataset ($k = 6$) met centroiden

De datapunten worden in de richting van hun centroiden verplaatst. De verplaatsing gebeurt in 5 stappen met telkens een stapgrootte van 0,1. Elke verplaatsing wordt weergegeven in figuur 109.

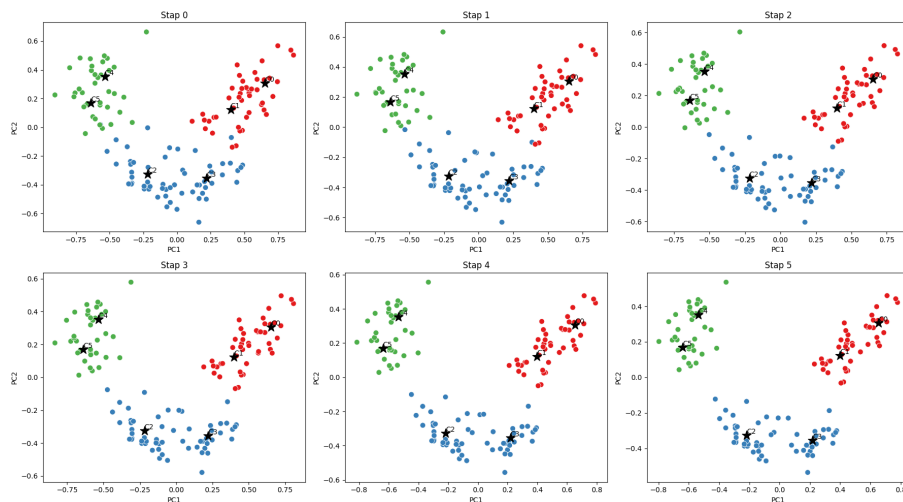


Figure 109: Supervised clusters na verplaatsing van 0,1 grootte - Wine dataset ($k = 6$)

Ten slotte worden de prestatie-indicatoren berekend. In figuur 110 worden de silhouette scores per stap voor de supervised methode met 6 clusters weergegeven in de vorm van boxplots. Uit die figuur blijkt dat de gemiddelde silhouette scores iets hoger liggen dan bij de unsupervised methode. Dit duidt aan dat de supervised methode iets beter datapunten in clusters definieert, maar nog steeds een lage score behaalt. Daarnaast is er wel een stijging van de silhouette scores te zien naarmate de stappen vorderen, net als bij de unsupervised methode. Weliswaar is de stijging bij de supervised clusteringmethode veel lichter. Verder liggen de gemiddelde waarden van de standaardafwijkingen op eenzelfde niveau als bij de unsupervised methode. Ook blijkt uit figuur 111 dat de standaardafwijkingen dalen, net als bij de unsupervised methode. De datapunten hebben dus minder variatie binnen een cluster. De WSS-scores die in figuur 112 weergegeven worden, vertonen eenzelfde dalingsgedrag als bij de unsupervised methode. Opmerkelijk is dat de gemiddelde waarden van de WSS-scores veel lager liggen dan bij de unsupervised methode, wat aangeeft dat de clusters compacter zijn bij de supervised methode.

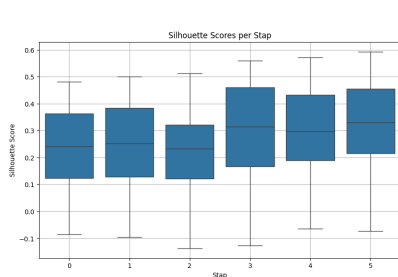


Figure 110:
Gemiddelde Silhouette ($k = 6$)

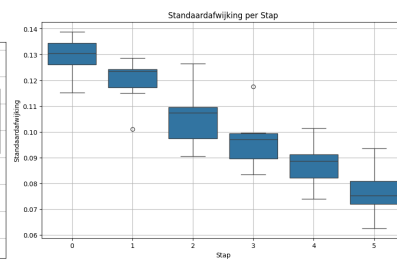


Figure 111:
Gemiddelde Standaardafwijking
($k = 6$)

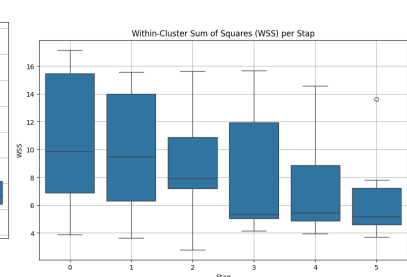


Figure 112:
Gemiddelde WSS-scores ($k = 6$)

9 clusters

Ten slotte wordt het clusteraantal voor een laatste keer toegenomen van 6 naar 9. De centroiden van de clusters, weergegeven in figuur 113, bevinden zich hier ook op een andere positie dan bij de unsupervised methode. Ook is er hier te zien dat de centroiden van de clusters dichter bij de andere centroiden gaan liggen, wat meer kans op overlappende clusters zorgt binnen de klassen.

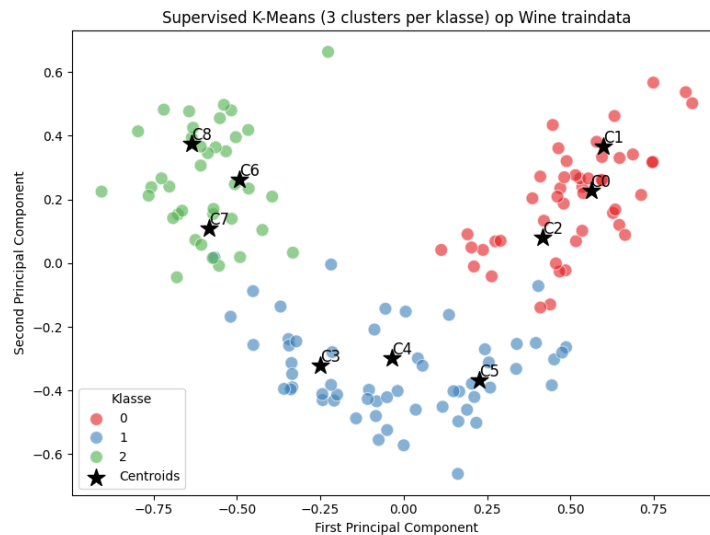


Figure 113: Initiële supervised clustering Wine dataset ($k = 9$) met centroiden

De stappen van verplaatsingen die worden doorgevoerd, worden gevisualiseerd in figuur 114. Uit die figuur blijkt dat de overlappings afnemen naarmate de punten dichter bij hun centroiden liggen. Verder kan er niet veel over de clusters gezegd worden. Daardoor worden de prestatie-indicatoren berekend om meer inzicht over de clusters te verkrijgen.

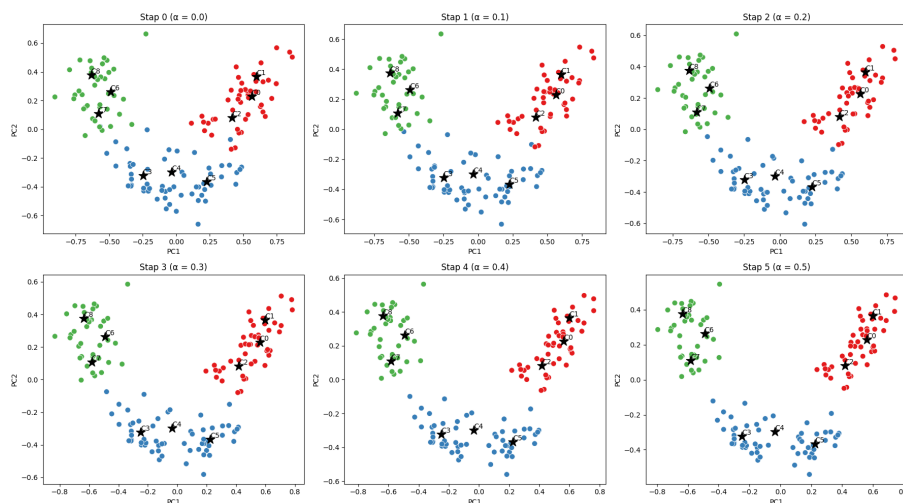


Figure 114: Supervised clusters na verplaatsing van 0,1 grootte - Wine dataset ($k = 9$)

Een eerste prestatie-indicator die berekend wordt, zijn de silhouette scores van de clusters per stap. Die scores worden in figuur 115 weergegeven. Net als bij de unsupervised methode, is er bij de supervised een stijging in de silhouette scores af te lezen uit de figuur. Ook liggen de gemiddelde silhouette

scores in de eerste stappen bij de supervised methode lager dan bij de unsupervised methode. Hoewel die gemiddelde scores tegen stap 5 ongeveer hetzelfde zijn. Vervolgens worden de standaardafwijkingen in figuur 116 weergegeven. Uit de figuur is te zien dat er een daling is in de standaardafwijkingen per stap, wat aangeeft dat de variatie binnen een cluster afneemt, net als bij de unsupervised methode. Ten slotte worden de WSS-scores gevisualiseerd in figuur 117. Uit die figuur is eenzelfde trend herkenbaar als bij de unsupervised methode. De WSS-scores dalen naarmate de datapunten hun centroiden naderen. Ook liggen de waarden ongeveer op eenzelfde niveau als bij de unsupervised methode.

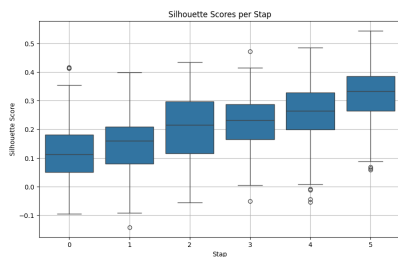


Figure 115:
Gemiddelde Silhouette (k = 9)

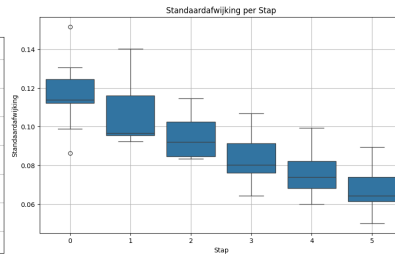


Figure 116:
Gemiddelde Standaardafwijking
(k = 9)

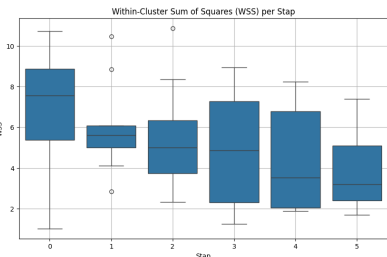


Figure 117:
Gemiddelde WSS-scores (k = 9)

5.3.3 KNN

Na het afronden van beide K-Means clusteringmethoden, wordt de originele Wine dataset gebruikt om het KNN-algoritme toe te passen. Eerst en vooral wordt het beste aantal burens bepaald, door middel van de accuracies van verschillende k's te bestuderen. In figuur 118 worden de accuracies van de verschillende k's weergegeven. Uit die figuur blijkt dat het aantal burens gelijk aan 3 ervoor zorgt dat de accuracy het hoogst is. Die accuracy bedraagt 89%. Er wordt dus voor de Wine dataset verder gewerkt met 3 als het aantal burens.

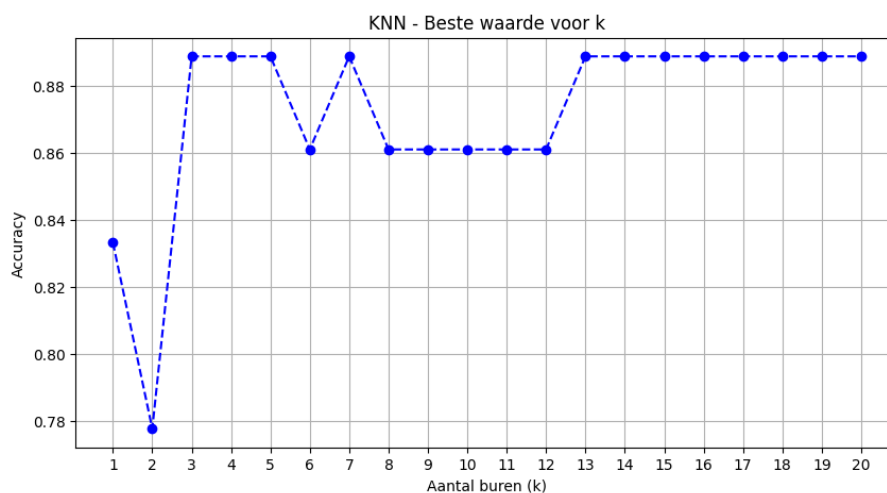


Figure 118: KNN accuracy per k - Wine dataset

Vervolgens worden de datapunten naar hun dichtstbijzijnde centroiden verplaatst. De verplaatsingen gebeuren in 5 evengrote stappen, met telkens een stapgrootte van 0,1. In figuur 119 wordt de verplaatsing weergegeven voor het KNN-algoritme met 3 burens. Uit die grafiek blijkt dat de punten verder van de scheidingslijn komen te staan, waardoor de scheiding (de opening) veel groter wordt.

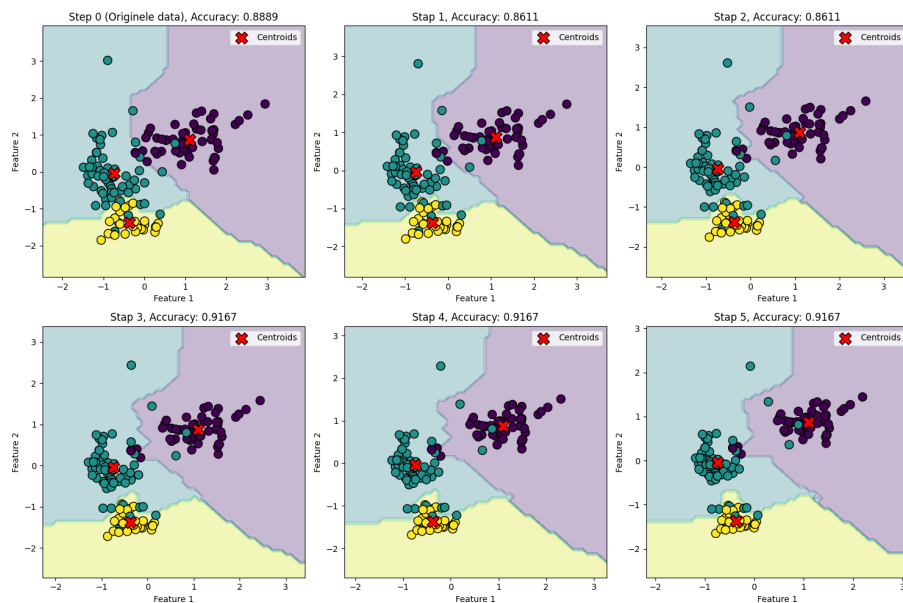


Figure 119: KNN's van alle stappen by verplaatsing van 0,1 - Wine dataset ($k = 3$)

Ten slotte wordt er naar de evolutie van de accuracies gekeken die weergegeven worden in figuur 120. Uit de accuracies blijkt dat er in de eerste twee stappen van de verplaatsing een zeer kleine daling in de accuracies is. Echter, wordt die daling snel weggewerkt in stap drie. De accuracy vanaf stap 3 overschrijdt de oorspronkelijke accuracy van 89%. Vanaf stap vier is er ook te zien dat de accuracy een maximum bereikt en zich zo voordoet voor de overige stappen. Die maximale accuracy bedraagt ongeveer 92%.

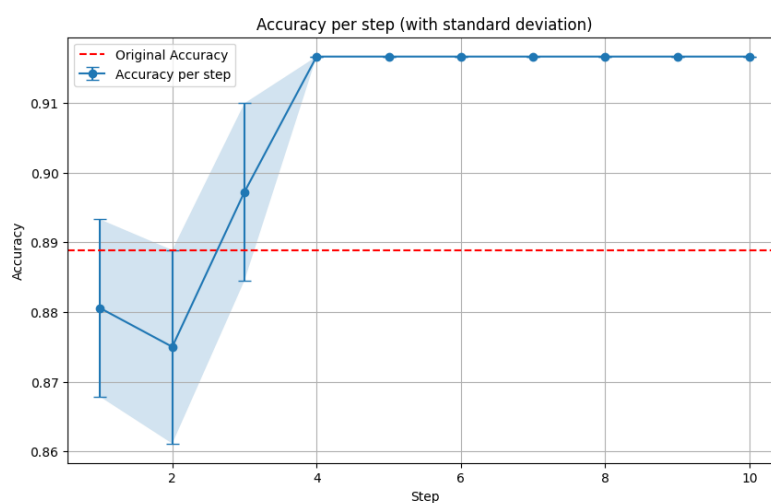


Figure 120: KNN accuracy na 10 stappen

5.3.4 KNN toegepast op supervised K-Means

Voor de KNN-methode toegepast op de supervised K-Means clusteringmethode wordt er verder gewerkt met de traindata die verkregen wordt na het opsplitsen van de originele dataset in 80% traindata en 20% testdata. De KNN's samen met de beslissingsgrenzen worden in figuur 121 weergegeven voor alle stappen. Uit die figuur blijkt dat de openingen nog groter worden dan bij de gewone KNN-methode. Ook worden de scheidelijnen lineair.

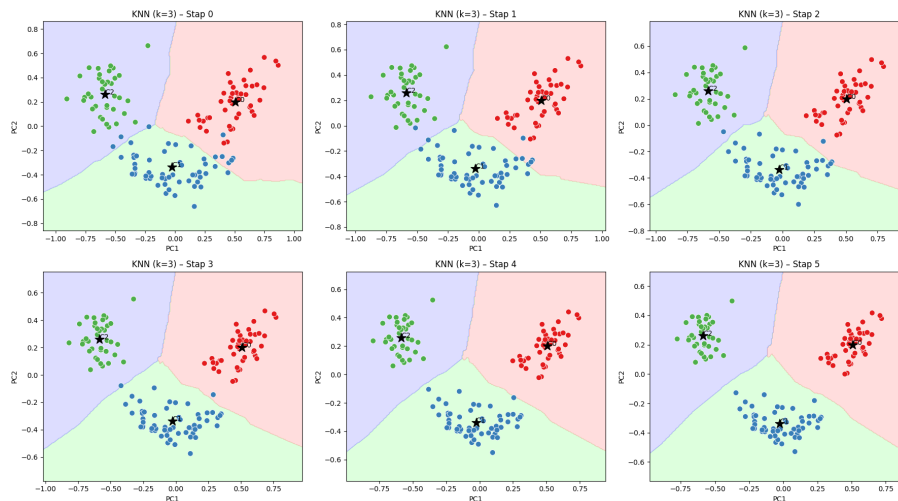


Figure 121: KNN's van alle stappen toegepast op de supervised K-Means bij verplaatsing van 0,1 - Wine dataset ($k = 3$)

In figuur 122 worden de train- en testaccuracies van de KNN-methode toegepast op de supervised K-Means weergegeven, maar dan voor 10 stappen. Eerst en vooral is er te zien dat de oorspronkelijke accuracy, bij stap 0, groter is dan die van de gewone KNN-methode en bedraagt 93%. Na de eerste verplaatsingsstap kent het model een stijging in zowel de train- als testaccuracy. De testaccuracy stijgt tot 97% en blijft die waarde constant behouden voor de overige stappen. De trainaccuracy kent op hetzelfde moment als de testaccuracy een stijging. De trainaccuracy bereikt na twee stappen een waarde van 100% en blijft die waarde voor de 10 stappen ook behouden.

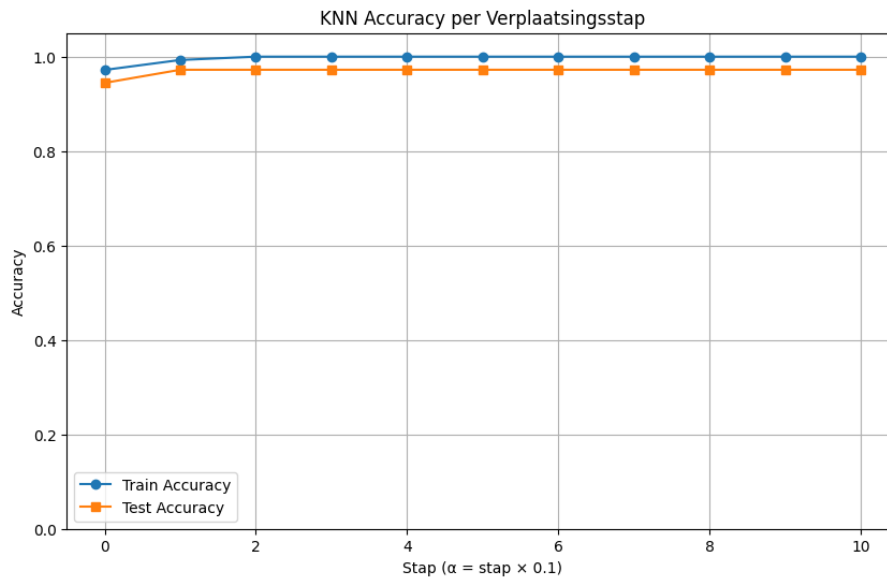


Figure 122: KNN toegepast op de supervised K-Means accuracy (10 stappen)

5.3.5 Beslissingsbomen

Een laatste methode die op de Wine dataset wordt toegepast, zijn de beslissingsbomen. De beslissingsbomen worden aangemaakt op basis van de eerder geselecteerde attributen. Die attributen zijn *Proline* en *Flavanoids*. De eerste beslissingsbomen worden weergegeven in figuur 123. Uit die figuur blijkt dat de structuur van de bomen sterk verschilt. Elke fold heeft een ander aantal splitsingen, maar de attributen waarop gesplitst wordt, komen overeen.

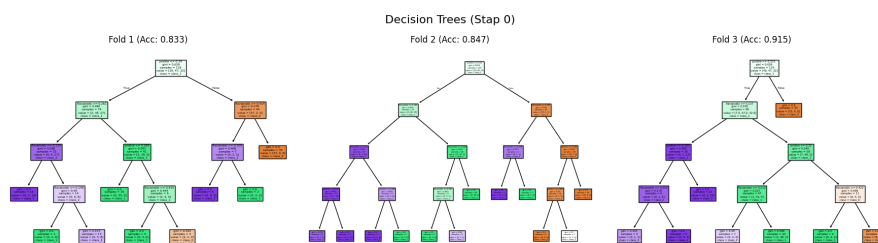


Figure 123: Initiële beslissingsbomen - Wine dataset

Vervolgens worden er telkens drie bomen gegenereerd aan de hand van 3-voudige crossvalidatie om per stap van verplaatsing naar de structuur te kijken. Per stap worden er 3 bomen weergegeven in figuur 124. Naarmate de stappen vorderen, neemt het aantal splitsingen per boom af. Ook gebeuren de splitsingen op basis van hetzelfde attribuut en dezelfde beslissingswaarde. Dit valt het felst op in stap 3.

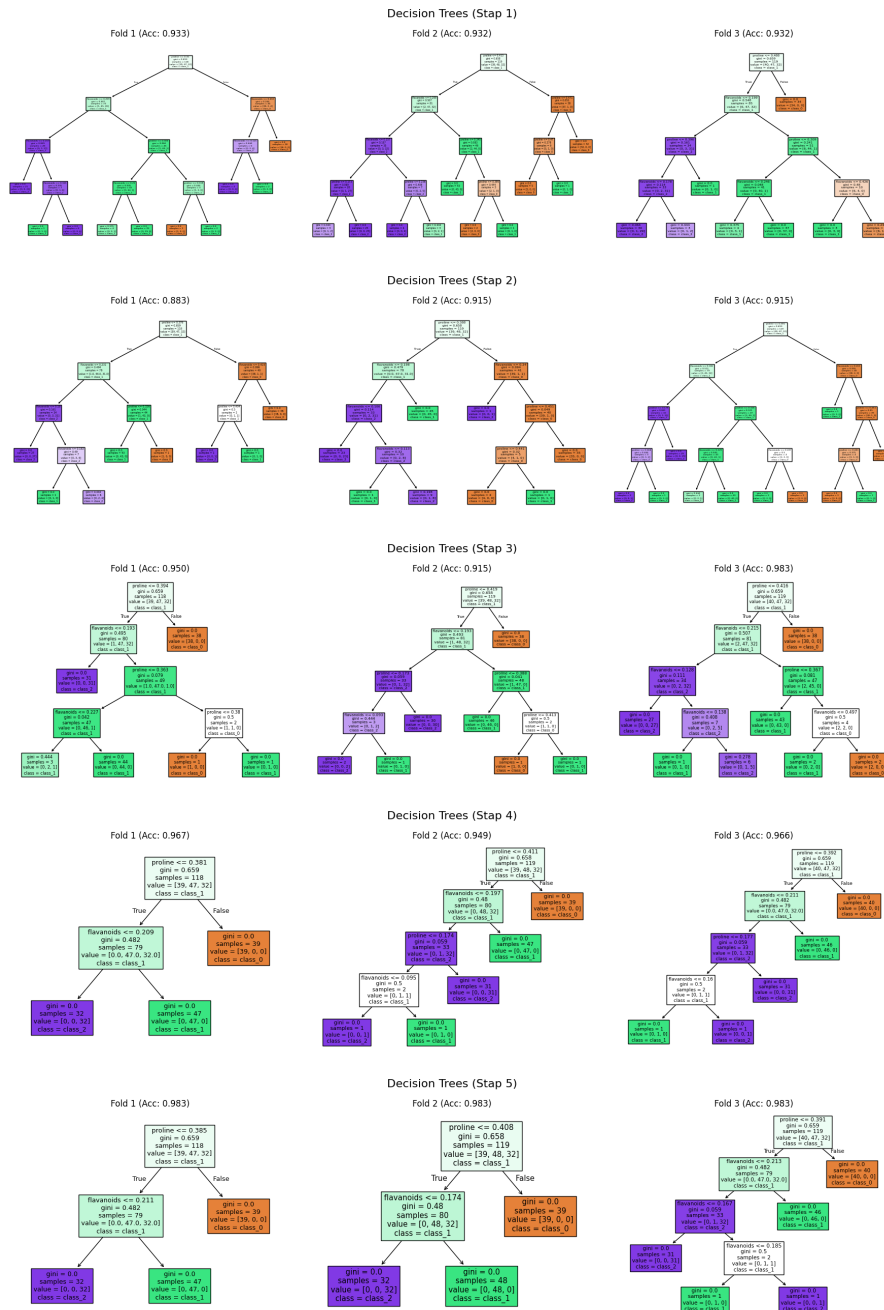


Figure 124: Beslissingsbomen per stap van verplaatsing - Wine dataset

In figuur 125 worden de gemiddelde accuracies per stap weergegeven. De gemiddelde accuracy van de stap zonder verplaatsing bedraagt 87%. Verder stijgt de accuracy naar 93% na de eerste verplaatsing. Echter blijft die accuracy niet behouden en daalt de accuracy tot 90% voor stap 2. Vanaf stap 3 is er een stijging te zien en bereikt de methode in stap 8 een accuracy van 100%.

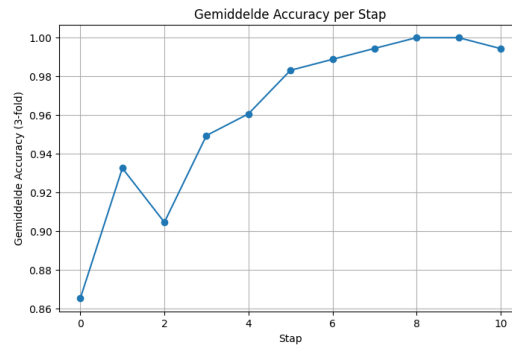


Figure 125: Gemiddelde accuracy beslissingsbomen (10 stappen)

Uit de grafiek van de feature importances per stap, weergegeven in figuur 126, is af te leiden dat beide attributen belangrijk zijn. De attributen hebben beide een belangrijkheid van ongeveer 50%. Echter, is vanaf stap 9 een felle stijging te zien in het belang van het attribuut *Flavanoids*.

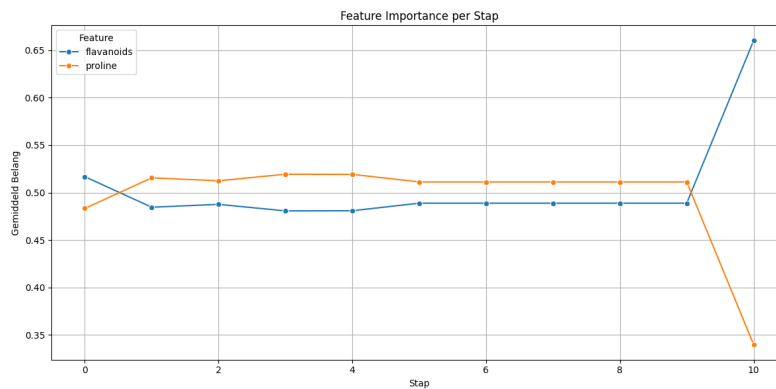


Figure 126: Feature Importance per stap (10 stappen)

5.3.6 Beslissingsbomen toegepast op supervised K-Means

De beslissingsbomen die weergegeven worden in figuur 123, zijn gemaakt door de beslissingsboom-methode toe te passen op de supervised K-Means methode. Hiervoor wordt de traindata van de Wine dataset gebruikt die bekomen is uit de train-test-split. De eerste twee splitsingen van de bomen gebeuren voor alle drie de folds door hetzelfde attribuut. Enkel de waarde van de voorwaarden waarop de splitsbeslissing gebeurt, verschilt.

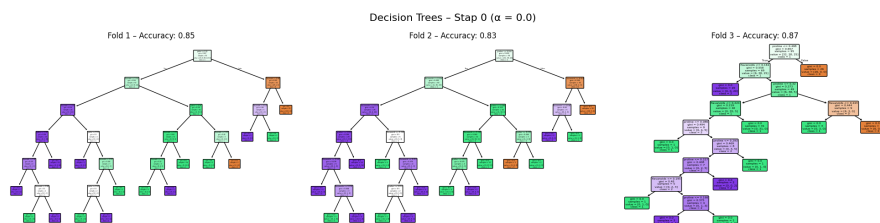


Figure 127: Initiële supervised beslissingsbomen - Wine dataset

Door de datapunten naar de dichtstbijzijnde centroiden te verplaatsen, worden de beslissingsbomen in figuur 128 gegenereerd. Naarmate de datapunten dicht bij hun centroiden liggen, nemen het aantal

splitsingen van de beslissingsbomen af. Ook gebeuren de splitsingen op basis van dezelfde attributen en neemt het verschil in beslissingswaarde af. De voorwaarden waarop de splitsbeslissingen gebeuren zijn minder breed naarmate de punten verplaatsen. Dit geeft aan dat de bomen beslissingen beter en effectiever zijn. In de finale stap, stap 5, is er duidelijk te zien dat de tweede splitsing gebeurt op basis van het attribuut *Flavanoids*. Ook bedraagt de beslissingswaarde voor fold 1 en 2 hetzelfde. De waarde van fold 3 ligt dicht bij de waarde van de andere folds.

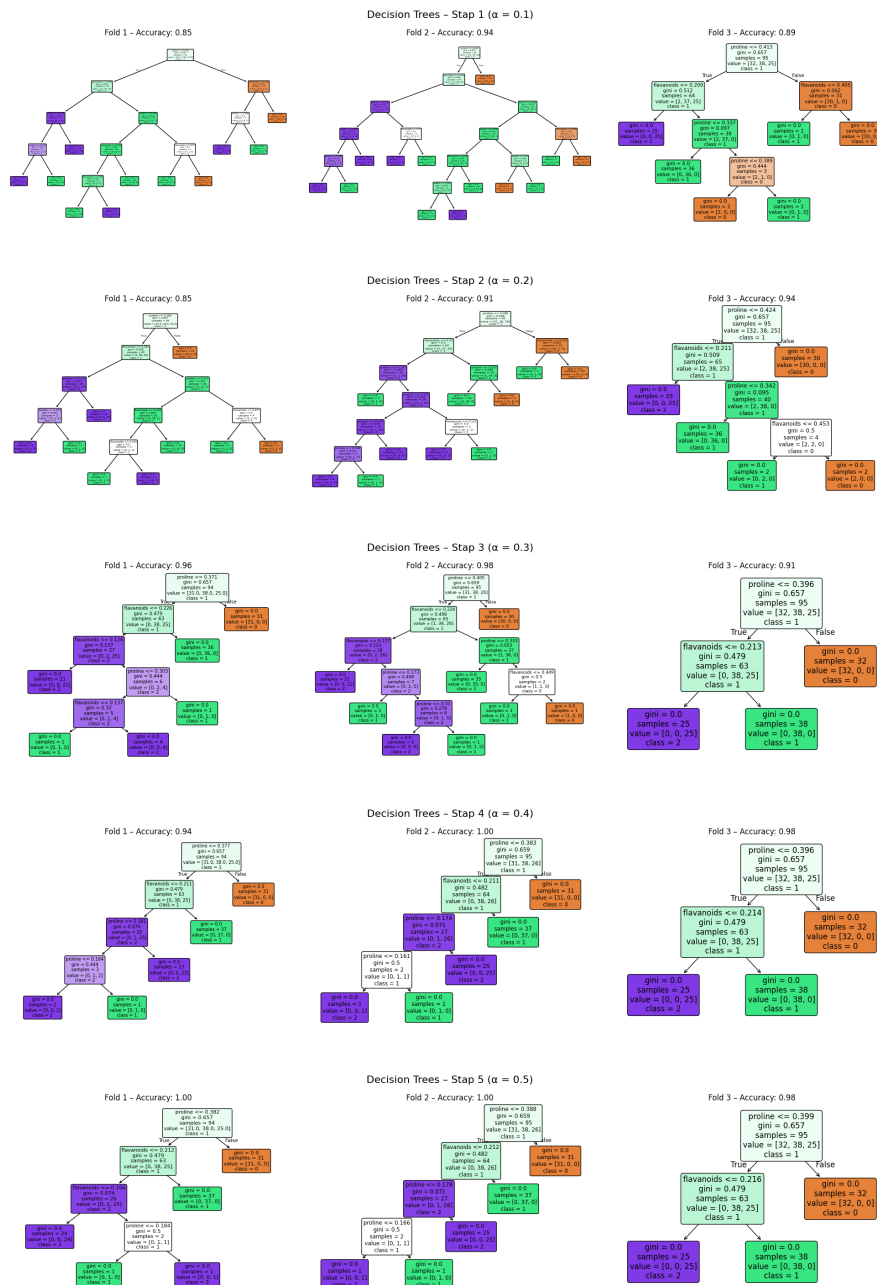


Figure 128: Beslissingsbomen toegepast op de supervised K-Means per stap van verplaatsing - Wine dataset

Naast de structuur van de bomen, wordt er ook naar de accuracies van de bomen gekeken. In figuur 129 worden de train- en testaccuracies van alle stappen weergegeven. De testaccuracy bij stap 0, waar geen verplaatsing was, bedraagt 80%. Die waarde ligt iets lager dan bij de gewone beslissingsbomen. Verder is er een stijging van de testaccuracies waarneembaar en bereiken de bomen een accuracy

van 88%. De trainaccuracies daarentegen blijven voor alle stappen een constante waarde van 100% behouden.

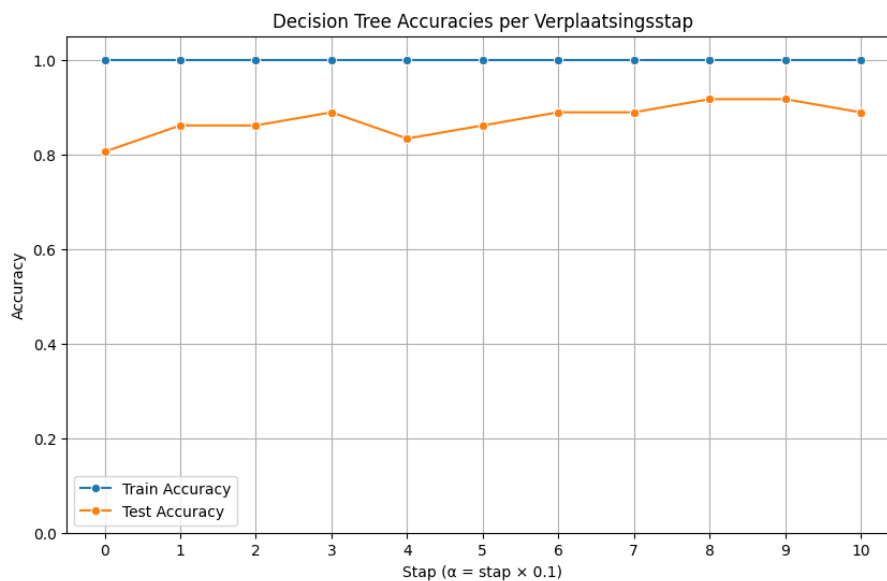


Figure 129: Gemiddelde accuracy beslissingsbomen toegepast op supervised K-Means(10 stappen)

Ten slotte wordt er gekeken naar de feature importance van de attributen *Proline* en *Flavanoids*. Initieel was het attribuut *Flavanoids* het meest dominant in de beslissingen van de bomen, maar naarmate de punten verplaatsen, neemt de invloed van het attribuut af. Uit figuur 130 is er af te leiden dat beide attributen ongeveer hetzelfde belang hebben van 50%. Ook is er te zien dat als er een stijging in belang van het attribuut *Proline* is, dan daalt het belang van het attribuut *Flavanoids*.

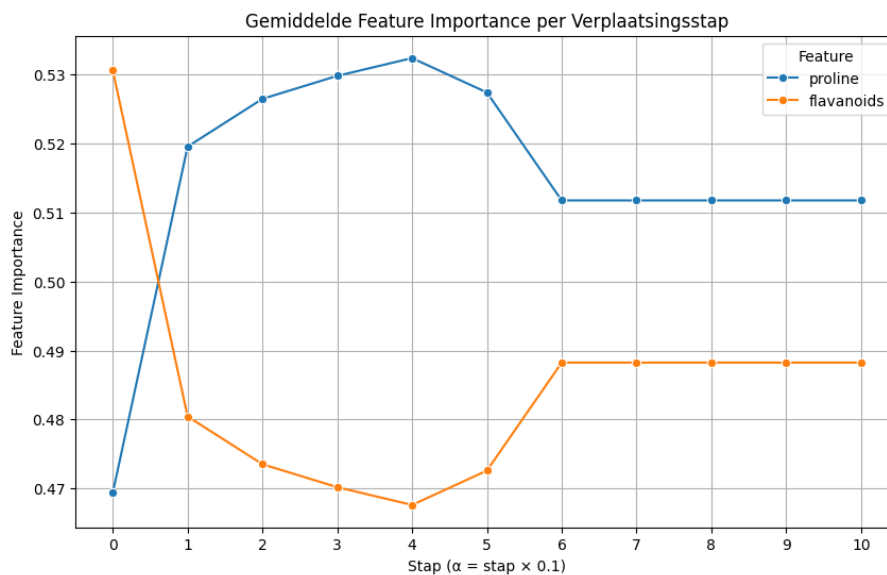


Figure 130: Feature Importance per stap (10 stappen)

5.4 Titanic

De laatste dataset die voor de experimenten wordt gebruikt, is de Titanic dataset. De Titanic dataset wordt vaak gebruikt om classificatieproblemen te oefenen, rond het voorspellen van de overleving van passagiers. De dataset bestaat uit 15 variabelen van 891 passagiers. Uit die 15 variabelen worden er aan de hand van de *SelectKBest* methode 8 variabelen geselecteerd om verder op te nemen in de analyse. Die 8 variabelen zijn: *Survived*, *Pclass*, *Sex*, *Age*, *Sibsp*, *Parch*, *Fare* en *Embarked*. De *Survived* variabele is een binaire variabele die als doelvariabele wordt gebruikt om na te gaan of de passagiers overleven of niet. De variabele *Pclass* geeft de klasse van de passagier aan. Het geslacht wordt weergegeven onder de variabele *Sex*. De variabele *Sibsp* geeft het aantal broers, zussen of echtgenoten aan boord aan. Voor het aantal ouders of kinderen aan boord wordt de variabele *Parch* gebruikt. De ticketprijzen komen onder de variabele *Fare* terecht. Ten slotte wordt de variabele *Embarked* gebruikt voor de opstapplaatsen (Jaiswal, 2025).

5.4.1 Unsupervised K-Means

Na het selecteren van de beste attributen, wordt de unsupervised K-Means methode uitgevoerd. Eerst en vooral wordt er via de elbow- en silhouettemethode, weergegeven in figuur 131, het optimale aantal clusters bepaald. Uit de grafieken blijkt dat het optimale aantal clusters gelijk is aan 2. Voor de K-Means clusteringmethode wordt er voor de Titanic dataset gekozen om de volgende clustercombinaties te gebruiken: 2, 4 en 6 clusters.

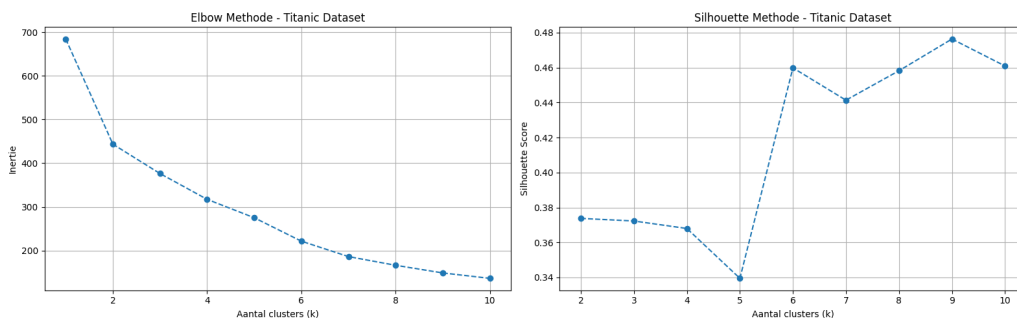


Figure 131: Elbow- en Silhouettemethode - Titanic dataset

2 clusters

Startend met het clusteraantal van 2 worden de clusters weergegeven in figuur 132. Wat er opvallend is bij de Titanic dataset, is dat de datapunten eerder in kleine groepjes staan dan los verspreid. Dit komt namelijk door de categorische variabelen die omgezet worden naar numerieke variabelen.

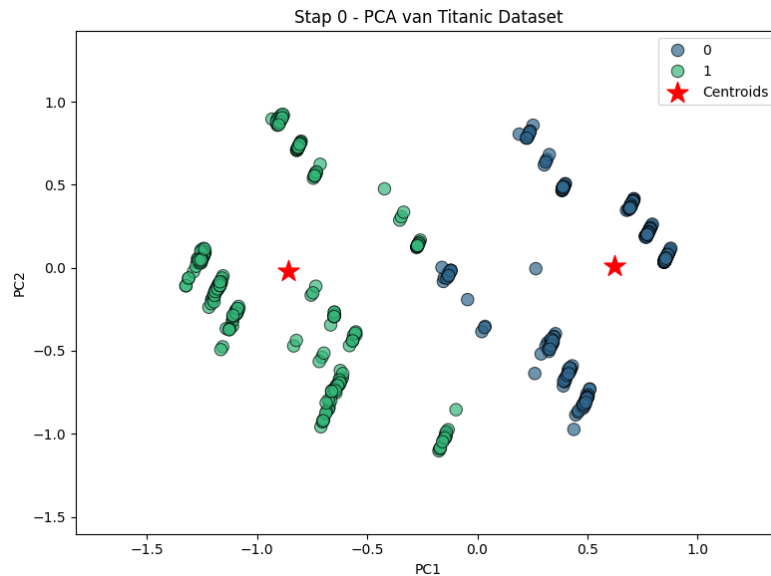


Figure 132: Initiële clustering Titanic dataset ($k = 2$)

Vervolgens worden de datapunten in vijf stappen met telkens 0,1 grootte verplaatst naar hun centroiden. De verplaatsingen worden per stap in figuur 133 weergegeven. Uit de figuren blijkt dat er minder spreiding is van de datapunten binnen een cluster. Ook worden de clustervormen veel kleiner en compacter naarmate de verplaatsingen vorderen.

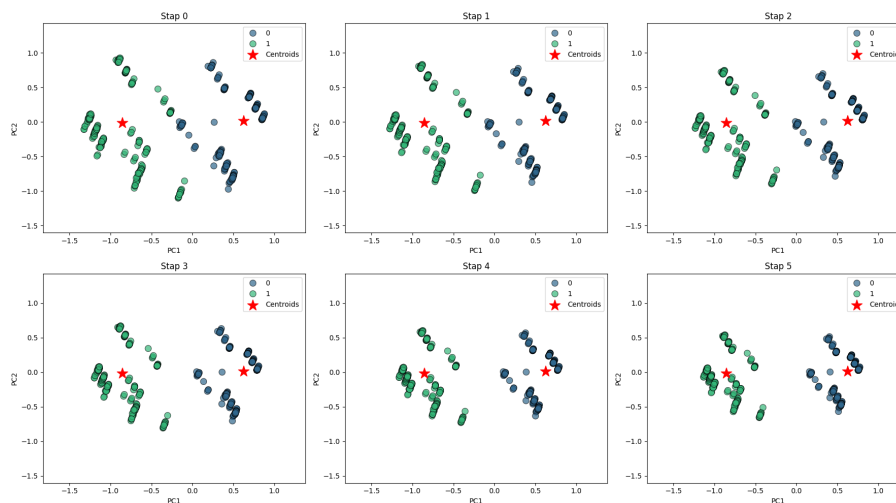


Figure 133: Clusters na verplaatsing van 0,1 grootte - Titanic dataset ($k = 2$)

Ten slotte worden de prestatie-indicatoren berekend. In figuur 134 worden de silhouette scores weergegeven. De gemiddelde silhouette scores bedragen bij geen verplaatsing 0,45. Naarmate de stappen vorderen, worden de datapunten van de clusters beter gedefinieerd en neemt het gemiddelde dus toe. Het gemiddelde neemt in stap vijf toe tot 0,62. Daarnaast worden de standaardafwijkingen weergegeven in figuur 135. Uit die figuur is een herkenbaar patroon te zien, namelijk dat de standaardafwijkingen dalen naarmate de punten dichter bij hun centroiden komen liggen. De WSS-scores, weergegeven in figuur 136, vertonen ook een daling in de waarden.

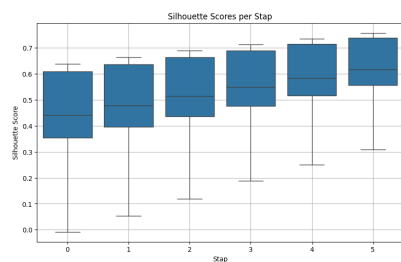


Figure 134:
Gemiddelde Silhouette ($k = 2$)

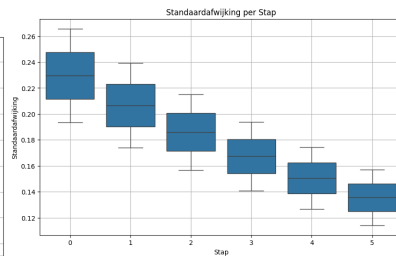


Figure 135:
Gemiddelde Standaardafwijking
($k = 2$)

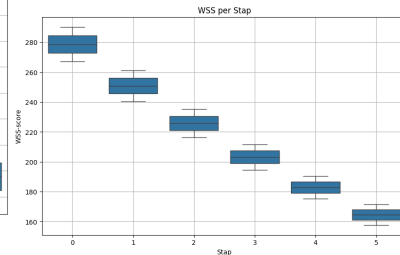


Figure 136:
Gemiddelde WSS-scores ($k = 2$)

4 clusters

Het verhogen van het clusteraantal van 2 naar 4 resulteert in figuur 137. Zoals eerder bij 2 clusters te zien was, zijn de punten hier ook gegroepeerd, maar is er op enkele plaatsen een overlap te zien. De overlap kan aangeven dat er een lagere silhouette score bij 4 clusters is dan bij 2.

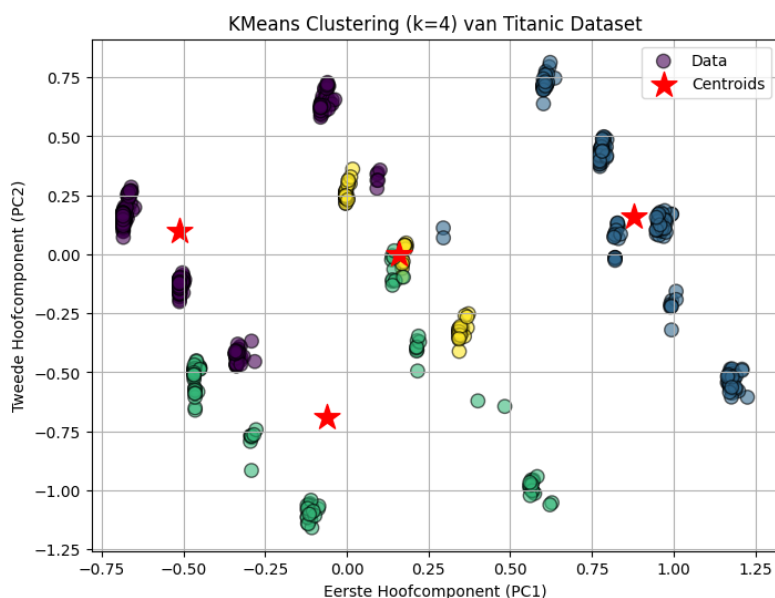


Figure 137: Initiële clustering Titanic dataset ($k = 4$)

Vervolgens worden de datapunten naar hun dichtstbijzijnde centroiden verplaatst in vijf evengrote stappen. Die stappen worden in figuur 138 gevisualiseerd. Vanaf stap 2 is er geen overlap meer tussen de clusters, wat voor een stijging in de silhouette score kan zorgen. Dit betekent dat er een betere clusterdefiniëring is.

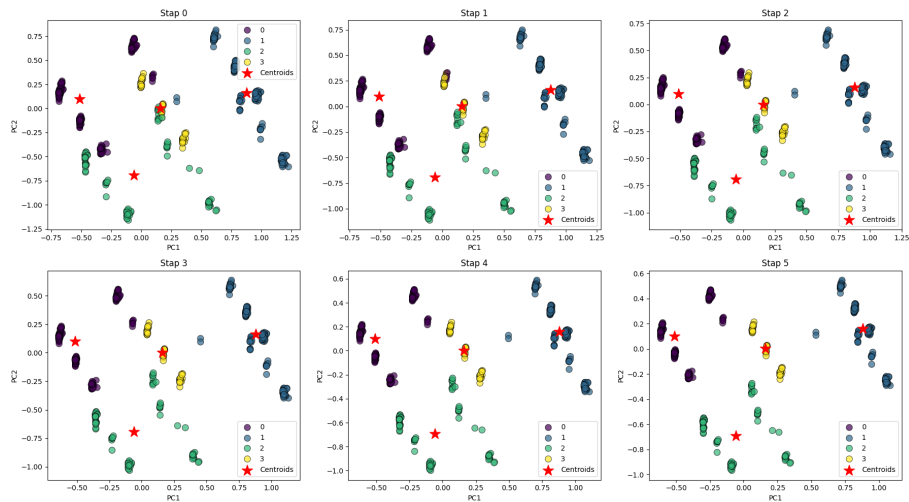


Figure 138: Clusters na verplaatsing van 0,1 grootte - Titanic dataset ($k = 4$)

Ten slotte worden de boxplots van de berekende prestatie-indicatoren gemaakt. De boxplots in figuur 139 geven de silhouette scores per stap weer. Uit die figuur blijkt dat de silhouette scores stijgen naarmate de stappen vorderen. De gemiddelde silhouette scores liggen ook lager dan bij 2 clusters. Dit is te verklaren door de overlap die aanwezig was in de clustervisualisaties. In figuur 140 worden de boxplots van de standaardafwijkingen per stap weergegeven. De waarden van de standaardafwijking nemen af per verplaatsing in de richting van hun centroiden. Ook liggen de gemiddelde waarden van de standaardafwijkingen lager bij 4 clusters dan bij 2. Wat betreft de WSS-scores, is er in figuur 141 te zien dat de scores dalen na elke stap. Ook liggen de gemiddelde waarden van de WSS-scores lager bij 4 clusters dan bij 2. Dit duidt aan dat de clusters compacter zijn bij een clusteraantal van 4.

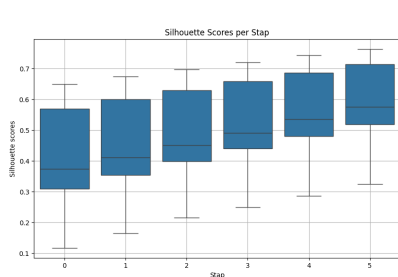


Figure 139:
Gemiddelde Silhouette ($k = 4$)

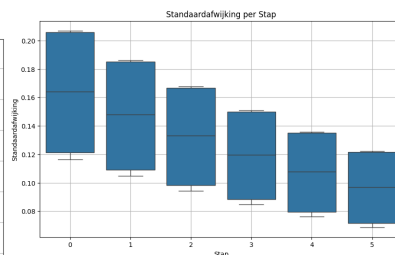


Figure 140:
Gemiddelde Standaardafwijking
($k = 4$)

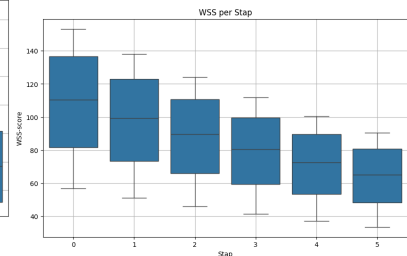


Figure 141:
Gemiddelde WSS-scores ($k = 4$)

6 clusters

Als laatste wordt het clusteraantal verhoogd naar 6 clusters. In figuur 142 worden de 6 clusters weergegeven. Ook hier is er op enkele plaatsen een overlap te zien.

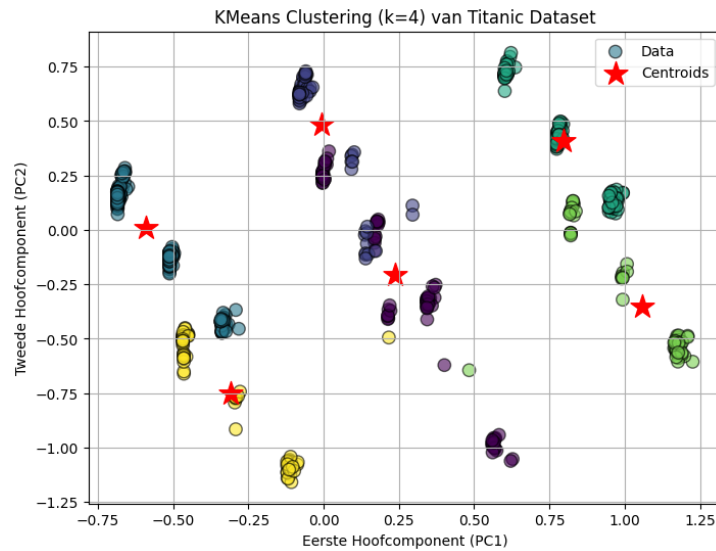


Figure 142: Initiële clustering Titanic dataset (k = 6) met centroiden

Verder worden de verplaatsingen weergegeven in figuur 143. Uit die figuur blijkt dat de overlappings verdwijnen. Ook is er te zien dat de clusters compacter worden.

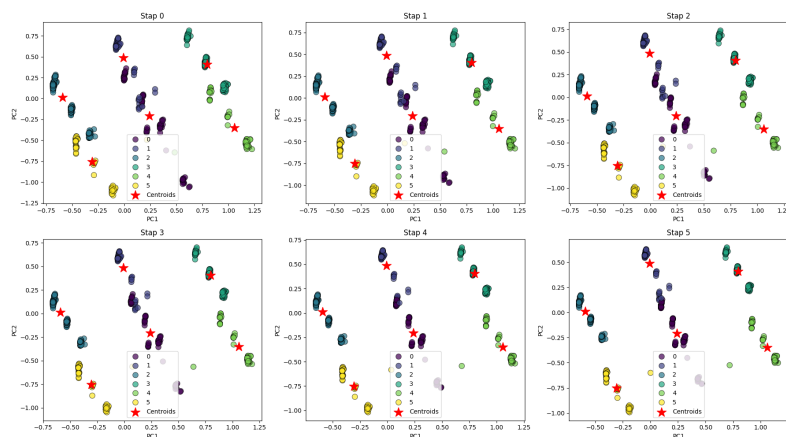


Figure 143: Clusters na verplaatsing van 0,1 grootte - Titanic dataset (k = 6)

Ten slotte worden de prestatie-indicatoren berekend. De silhouette scores weergegeven in figuur 144 vertonen net als bij 4 clusters een stijging in de waarden. De standaardafwijkingen, weergegeven in figuur 145, vertonen ook hetzelfde gedrag; de waarden dalen. Ook is er een lagere gemiddelde waarde van de standaardafwijkingen te zien voor 6 clusters. De WSS-scores dalen ook naarmate de stappen vorderen. In figuur 146 is er ook te zien dat de WSS-scores lager liggen dan bij 4 clusters.

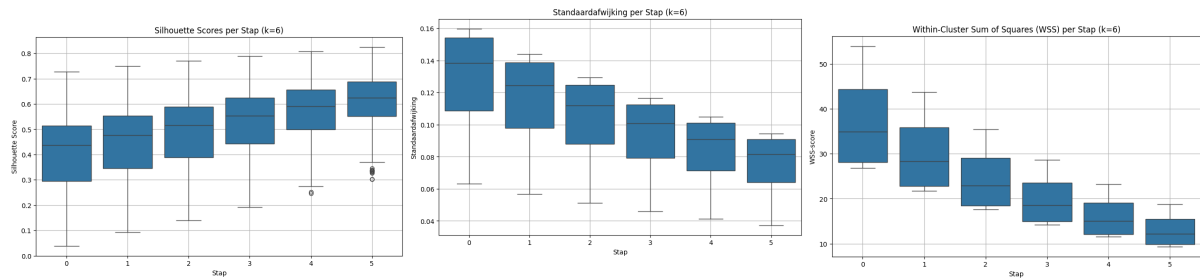


Figure 144:
Gemiddelde Silhouette (k = 6)

Figure 145:
Gemiddelde Standaardafwijking
(k = 6)

Figure 146:
Gemiddelde WSS-scores (k = 6)

5.4.2 Supervised K-Means

Voor de supervised K-Means clusteringmethode wordt de Titanic dataset opgesplitst in train- en testdata, respectievelijk 80% en 20%. De traindata wordt vervolgens gebruikt om de K-Means clusters uit te voeren. Voor de supervised methode wordt de clustercombinatie van de unsupervised methode behouden, maar worden de clusters per klasse gevormd. Die clustercombinatie is 2, 4 en 6 clusters.

2 clusters

De K-Means clusters waarbij het clusteraantal gelijk is aan 2 worden in figuur 147 weergegeven. Uit de figuur is er duidelijk af te leiden dat de centroiden van de twee clusters op een andere locatie liggen. Dit is anders dan bij de vorige drie datasets. Bij de Titanic dataset wordt de verplaatsing van de datapunten naar hun dichtstbijzijnde centroiden uitgevoerd, bij een clusteraantal van 2.

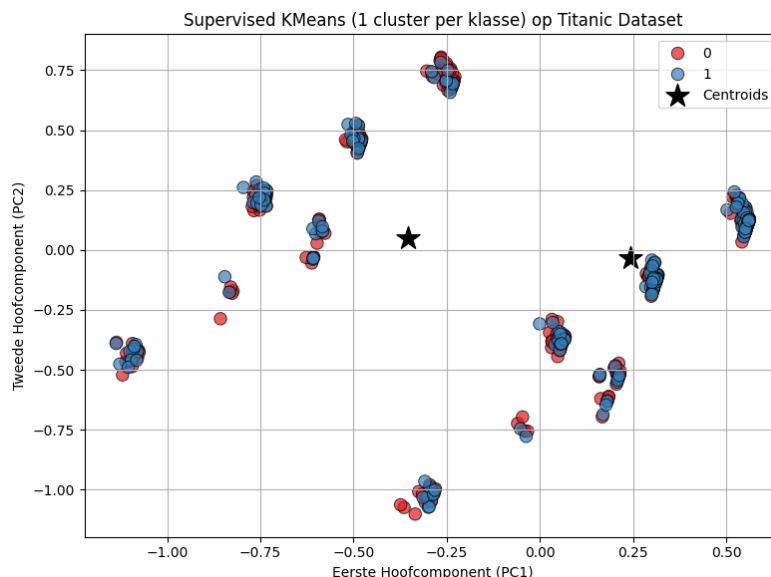


Figure 147: Initiële supervised clustering Titanic dataset (k = 2) met centroiden

De verplaatsing per stap wordt weergegeven in figuur 148. Uit die figuur blijkt dat de clusterpunten van beide klassen elkaar overlappen op sommige plaatsen. Hoewel de punten dicht bij hun centroiden liggen dan bij de unsupervised methode.

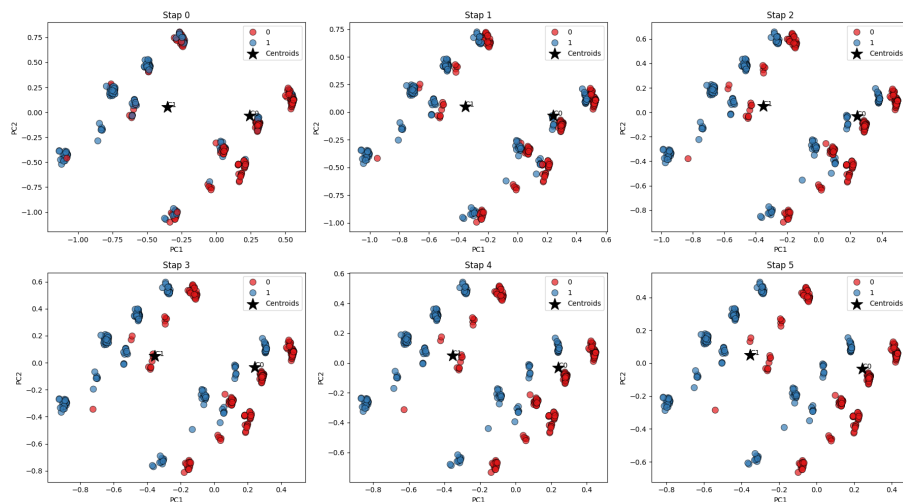


Figure 148: Supervised clusters na verplaatsing van 0,1 grootte - Titanic dataset ($k = 2$)

Uit de silhouette scores, weergegeven in figuur 149, blijkt dat er een stijging is naarmate de punten verplaatsen. Weliswaar is die stijging minder fel dan bij de unsupervised clusteringmethode. Dit geeft aan dat de datapunten beter gedefinieerd worden in hun cluster bij de unsupervised methode naarmate de punten verplaatsen.. Verder is er ook te zien dat de gemiddelde silhouette scores van de supervised methode lager liggen dan bij de unsupervised methode. De standaardafwijkingen, weergegeven in figuur 150, vertonen een gelijkaardig gedrag als bij de unsupervised methode. De waarden van de standaardafwijkingen dalen naarmate de punten hun centroiden naderen. Echter liggen de waarden van de standaardafwijkingen bij de supervised methode lager, wat aangeeft dat de punten minder verspreid zijn binnen de clusters. Ten slotte worden de WSS-scores weergegeven in figuur 151. De gemiddelde WSS-scores liggen veel lager bij de supervised methode dan bij de unsupervised methode. Dit geeft aan dat de clusters bij de supervised methode compacter zijn. Het dalende patroon dat de WSS-score vertoont, is wel gelijkaardig aan de unsupervised methode.

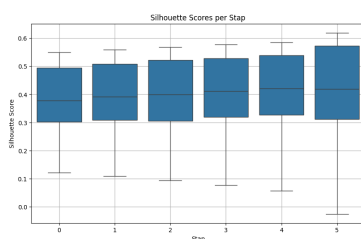


Figure 149:
Gemiddelde Silhouette ($k = 2$)

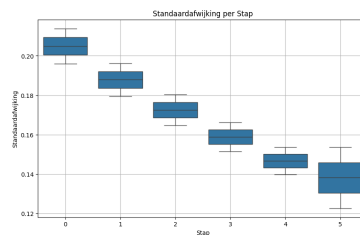


Figure 150:
Gemiddelde
Standaardafwijking ($k = 2$)

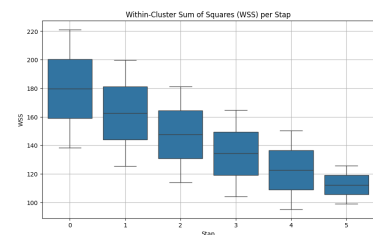


Figure 151:
Gemiddelde WSS-scores ($k = 2$)

4 clusters

Bij een clusteraantal van 4, worden de clusters gevisualiseerd in figuur 152. Uit de figuur blijkt dat de meeste clusters klein zijn en dat de datapunten dicht bij hun centroiden liggen.

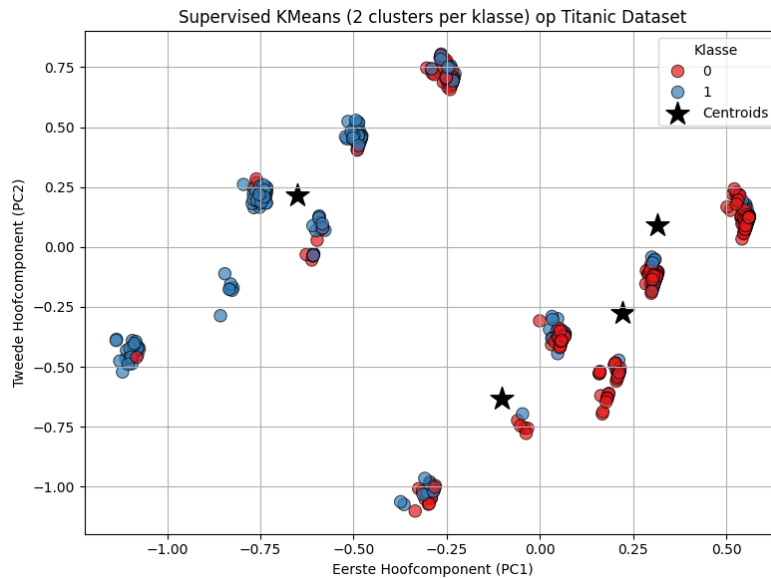


Figure 152: Initiële supervised clustering Iris-dataset ($k = 4$) met centroiden

De verplaatsing van de datapunten resulteert in figuur 153. Uit die figuur blijkt dat de verplaatsing ervoor heeft gezorgd dat de clusters compacter worden. De punten lijken meer verspreid te zijn dan bij de unsupervised methode. Om dit te bevestigen worden de prestatie-indicatoren berekend.

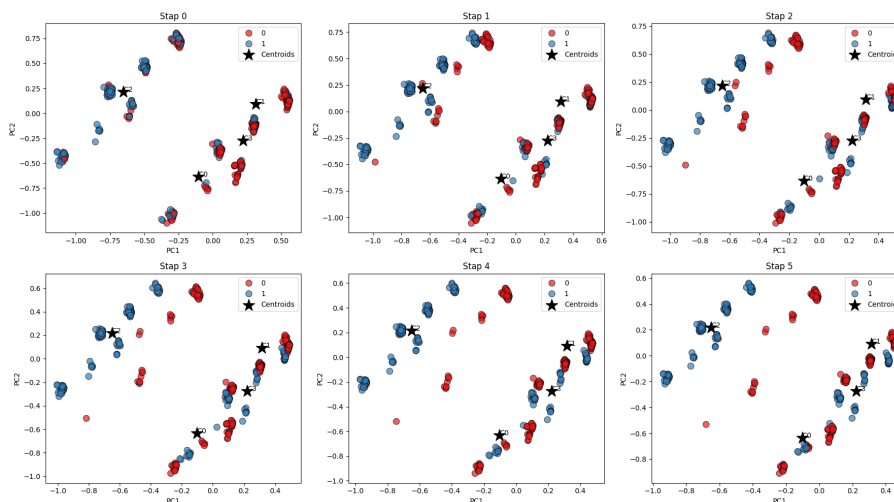


Figure 153: Supervised clusters na verplaatsing van 0,1 grootte - Titanic dataset ($k = 4$)

De silhouette scores, weergegeven in figuur 154, vertonen net als bij de unsupervised methode een stijging in waarde per verplaatsing. Wat er ook op te merken valt, is dat de gemiddelde silhouette score ongeveer op eenzelfde niveau ligt als bij de unsupervised methode. Verder liggen de gemiddelden van de standaardafwijkingen, weergegeven in figuur 155, hoger dan bij de unsupervised methode. Dit was op te merken bij de clustervisualisaties van de supervised methode, waarbij de centroiden veel meer bij de ene helft van de datapunten lag en dus verder van de andere helft. De WSS-scores daarentegen liggen bij de unsupervised methode veel lager, wat aangeeft dat de clusters compacter zijn. Ook is er in figuur 156 te zien dat er een daling is per stap.

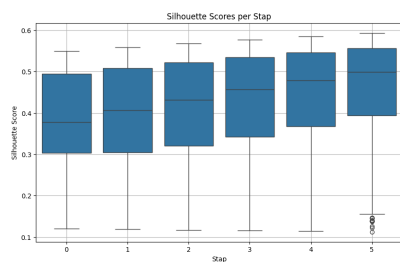


Figure 154:
Gemiddelde Silhouette ($k = 4$)

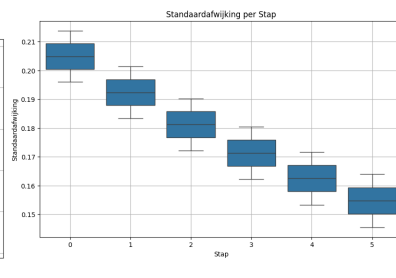


Figure 155:
Gemiddelde Standaardafwijking
($k = 4$)

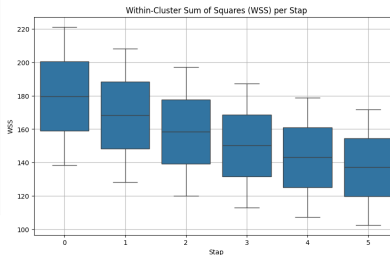


Figure 156:
Gemiddelde WSS-scores ($k = 4$)

6 clusters

Ten slotte wordt het clusteraantal van 6 gevisualiseerd in figuur 157 om de supervised methode af te ronden. Uit die figuur blijkt dat de extra centroiden dicht bij de datapunten komen liggen. Dit gaat later een effect hebben op de standaardafwijkingen van de clusters.

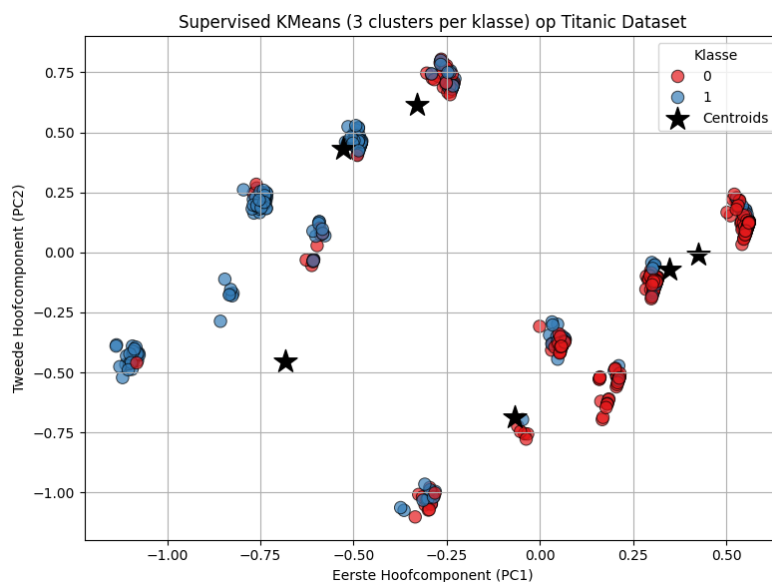


Figure 157: Initiële clustering Titanic dataset ($k = 6$) met centroiden

Verder worden de datapunten naar hun dichtstbijzijnde centroiden verplaatst. In figuur 158 worden de verplaatsingen weergegeven. In vergelijking met de vorige twee clusteraantallen, namelijk 2 en 4, liggen de datapunten van de clusters minder verspreid. Ook lijken de clusters compacter te worden na verplaatsing.

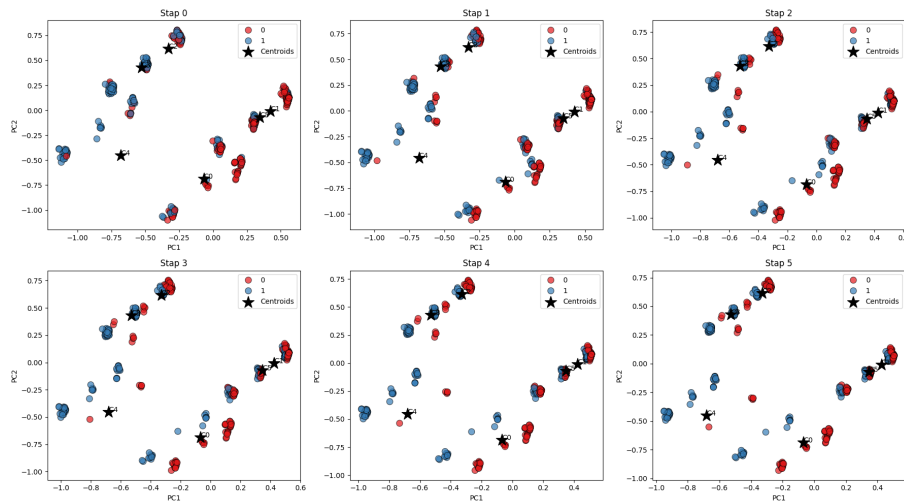


Figure 158: Supervised clusters na verplaatsing van 0,1 grootte - Titanic dataset ($k = 6$)

De prestatie-indicatoren van de clusters worden in een finale stap berekend. In figuur 159 worden de silhouette scores weergegeven. De silhouette scores vertonen niet hetzelfde gedrag als bij de unsupervised methode. Hier blijven de gemiddelde scores ongeveer gelijk voor de verschillende stappen, terwijl de scores een stijging kennen bij de unsupervised methode. Daarnaast worden de standaardafwijkingen in figuur 160 weergegeven. Uit die figuur blijkt dat de standaardafwijkingen dalen naarmate de punten verplaatsen, alsook dat de gemiddelde waarden hoger liggen dan bij de unsupervised methode. Weliswaar zijn de waarden van de standaardafwijkingen lager voor een clusteraantal van 6 dan voor een clusteraantal van 2 of 4. Als laatst worden de WSS-scores van de clusters in figuur 161 weergegeven. Ook hier is een gelijkaardig patroon te zien als bij de unsupervised methode, namelijk dat de scores dalen per verplaatsing en dus aangeven dat de clusters compacter worden. Weliswaar is de daling bij de supervised clusteringmethode veel lichter. De gemiddelde waarden van de WSS-scores liggen hoger bij de supervised methode dan bij de unsupervised methode. Vergeleken met de clusteraantallen 2 en 4 bij de supervised clusteringmethode, zijn de WSS-scores lager gelegen, wat aangeeft dat de clusters bij het clusteraantal van 6 compacter zijn.

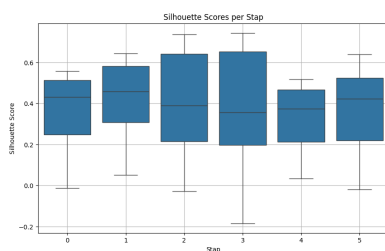


Figure 159:
Gemiddelde Silhouette ($k = 6$)

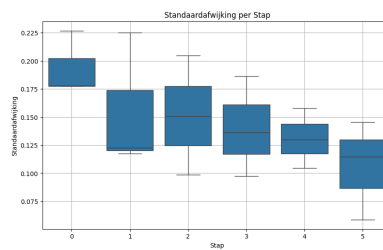


Figure 160:
Gemiddelde Standaardafwijking
($k = 6$)

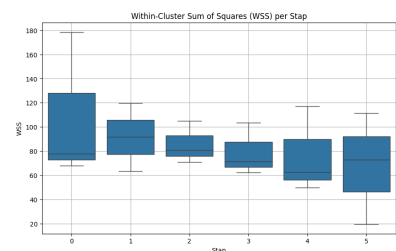


Figure 161:
Gemiddelde WSS-scores ($k = 6$)

5.4.3 KNN

Na het afronden van de K-Means clusteringmethode, wordt er overgeschakeld naar de KNN-methode. Vooraleer de KNN's gevisualiseerd worden, wordt er in figuur 162 het beste aantal burens (k) voor de Titanic dataset bepaald. Uit de figuur blijkt dat de beste accuracy voor $k = 3$ wordt behaald. De accuracy bij het aantal burens gelijk aan 3 bedraagt 84,92%.

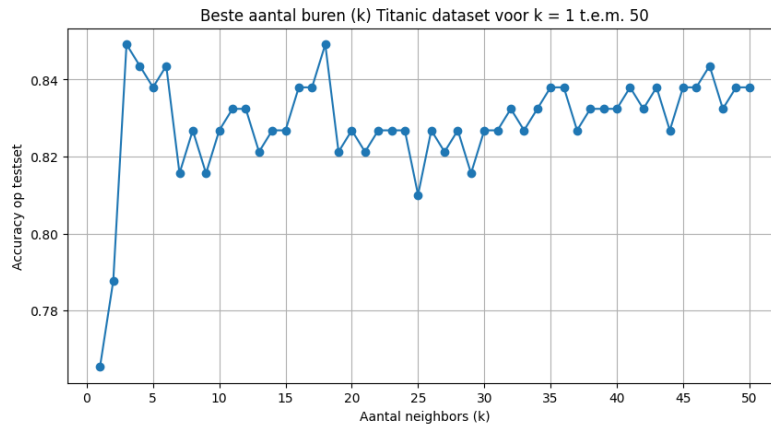


Figure 162: KNN accuracy per k - Titanic dataset

Nadat het beste aantal burens bepaald is, wordt het KNN-algoritme uitgevoerd samen met de verplaatsing van de datapunten naar hun centroiden. In figuur 163 worden de KNN's gevisualiseerd. De KNN's die terug te vinden zijn in de figuur, bevatten visualisaties van alle stappen, inclusief de centroiden en beslissingsgrenzen. De scheidingslijn blijkt hetzelfde te zijn voor alle stappen.

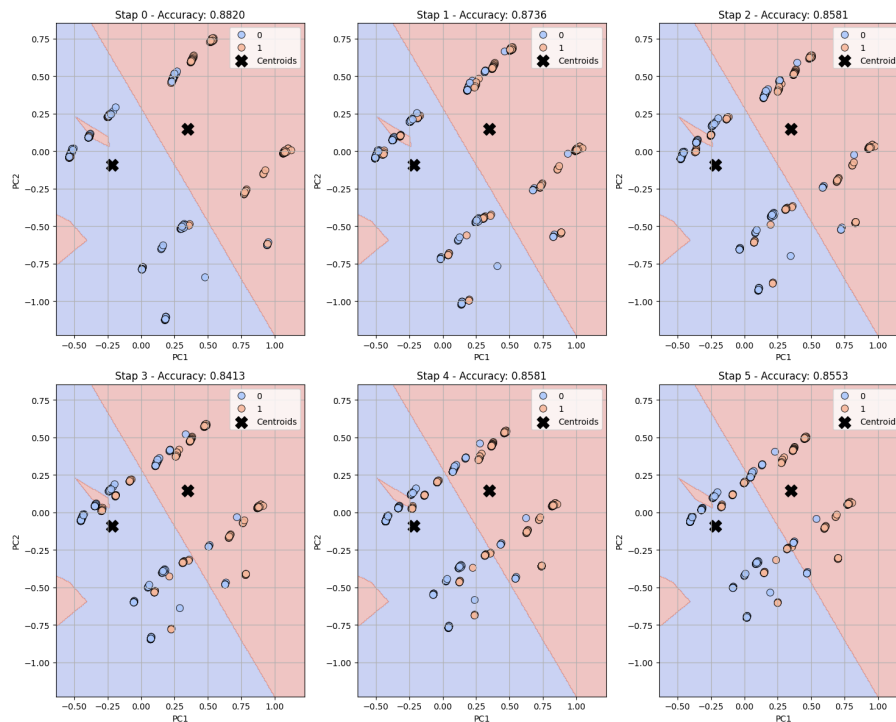


Figure 163: KNN's van alle stappen bij verplaatsing van 0,1 - Titanic dataset

Naast de KNN-visualisaties wordt er gekeken naar de accuracies van de stappen. Om een beter beeld te krijgen van de verplaatsing, worden de stappen toegenomen tot 10 en worden de accuracies voor de 10 stappen in figuur 164 weergegeven. Uit de figuur blijkt dat er weinig variatie is in de accuracy. De accuracy blijft binnen het bereik van 82% tot 89%.

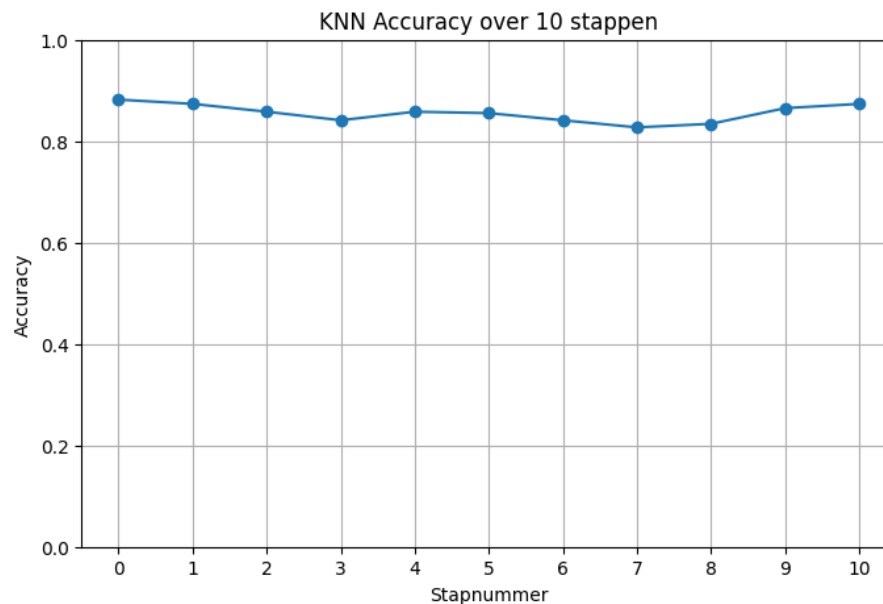


Figure 164: Accuracy na 10 stappen

5.4.4 KNN toegepast op de supervised K-Means

Om de KNN-methode af te ronden wordt de Titanic dataset opgesplitst in train- en testdata, respectievelijk 80% en 20% van de dataset. Na de opsplitsing wordt de traindata gebruikt om het KNN-algoritme toe te passen op de supervised K-Means clusteringmethode. Voor de KNN-visualisaties wordt hetzelfde aantal burens behouden, namelijk $k = 3$. De verplaatsingen van de datapunten worden in figuur 165 weergegeven. Uit die figuur blijkt dat er veel minder punten over de scheidingslijn verplaatsen dan bij de gewone KNN.

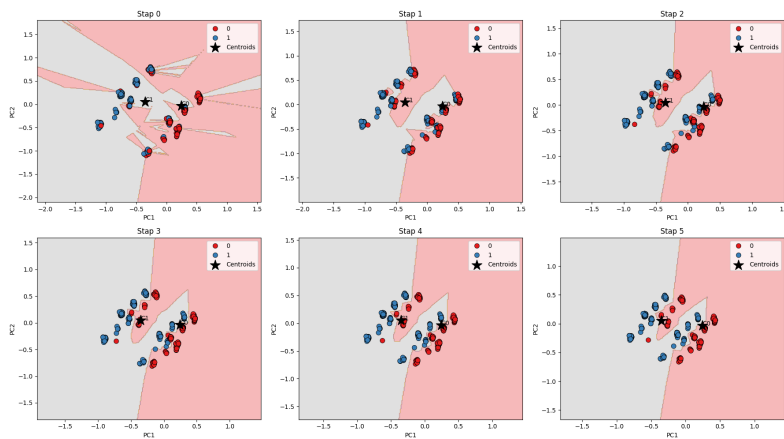


Figure 165: KNN's van alle stappen toegepast op de supervised K-Means bij verplaatsing van 0,1 - Titanic dataset

In figuur 166 worden zowel de train- als testaccuracies van de 10 stappen weergegeven. De testaccuracies van de 10 stappen voor de KNN-methode toegepast op de supervised K-Means gedragen zich hetzelfde als bij de gewone KNN. De testaccuracies blijven ongeveer hetzelfde en schommelen binnen het bereik van 80% en 85%. Weliswaar is er een stijging van de trainaccuracies te zien. Na vijf stappen bereikt de trainaccuracy een waarde van 100%. Het verplaatsen van de datapunten heeft dus niet onmiddellijk effect op de accuraatheid van de testset, maar wel op de trainset. Het model gaat alles perfect leren na enkele stappen.

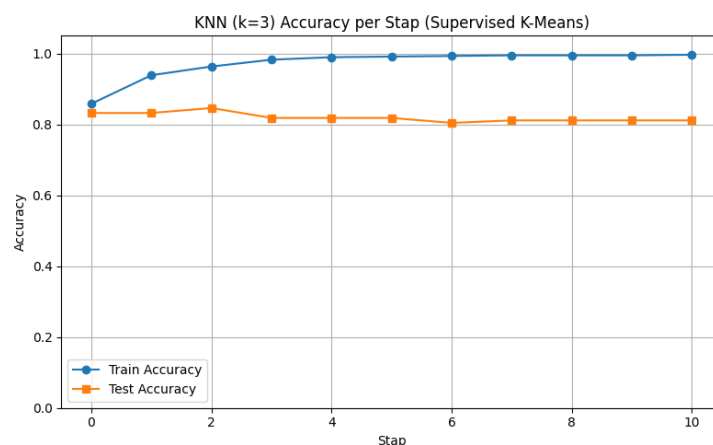


Figure 166: KNN toegepast op de supervised K-Means accuracy (10 stappen)

5.4.5 Beslissingsbomen

Ten slotte wordt er voor de Titanic dataset de beslissingsbomen gemaakt. Ook hier wordt er enkel gewerkt met de 8 attributen. Die 8 attributen zijn: *Survived*, *Pclass*, *Sex*, *Age*, *Sibsp*, *Parch*, *Fare* en *Embarked*. Vanwege de aanwezigheid van heel veel observaties, worden de bomen beperkt tot een maximale diepte van 4. Door de maximale diepte toe te kennen, is het moeilijker om iets over het aantal splitsingen te zeggen, want alle bomen stoppen bij eenzelfde diepte. In figuur 167 worden de drie beslissingsbomen weergegeven voor de originele dataset, waar geen verplaatsing plaatsvindt. Uit die figuur blijkt dat de waarde waarop de beslissingen gebeuren van diepte 2 verschillend is. De eerste twee diepten gebeuren op basis van dezelfde attributen en waarden.

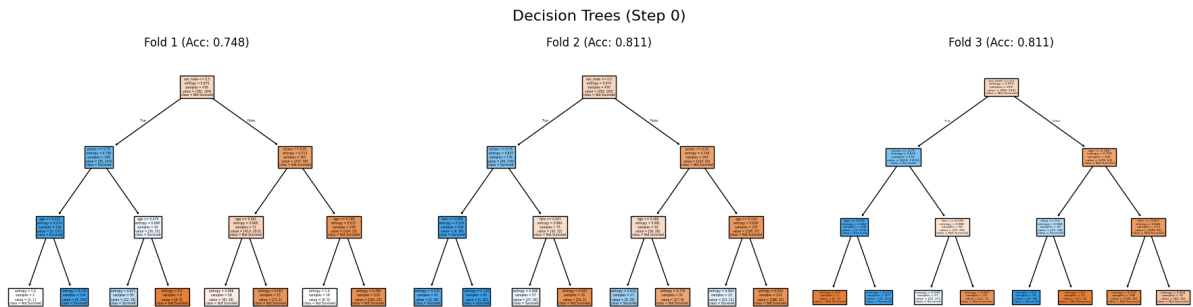


Figure 167: Initiële beslissingsbomen - Titanic dataset

In figuur 168 worden de beslissingsbomen weergegeven nadat de datapunten verplaatst zijn. Uit de figuur blijkt dat vanaf stap 2 er wijzigingen plaatsvinden in de beslissingwaarden van bepaalde splitsingen. In stap 3 zijn sommige splitsingen aangepast van attribuut, wat aangeeft dat de verplaatsing van de datapunten een invloed heeft op het model. Verder worden vertakkingen duidelijker, want de bomen maken nu splitsingen die beter gescheiden zijn. Na de vijf stappen is er duidelijk te zien dat de prestaties verbeteren nu er beter gesplitst wordt.

Vervolgens wordt er gekeken naar de gemiddelde accuracy van de bomen, weergegeven in figuur 169. Uit de figuur blijkt dat de accuracy relatief hoog begon, maar er is een duidelijke lichte daling te zien tot en met stap 3. Vanaf stap 4 stijgt de accuracy terug naar de originele. Dit geeft aan dat de verplaatsing ervoor zorgt dat de splitsingen beter gedefinieerd en consistent zijn over de folds. Na stap 8 blijft de accuracy stabiel. De stabielere accuracy geeft aan dat de verplaatsing geen significant effect meer heeft.

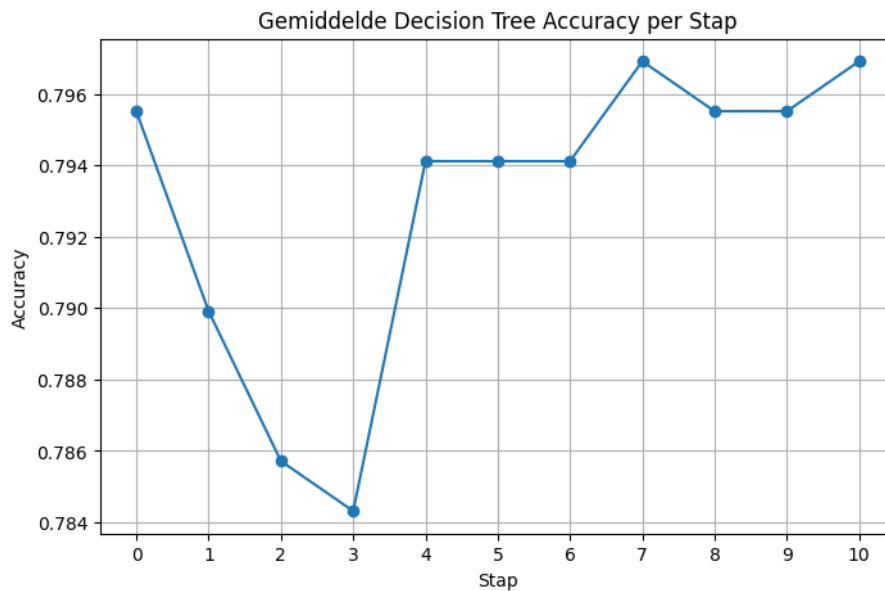


Figure 169: Gemiddelde accuracy beslissingsbomen (10 stappen)

Wat betreft het belang van de attributen op de beslissingen, in figuur 170 worden de belangen van de attributen weergegeven. Uit die figuur blijkt dat het attribuut *Sex* de meeste invloed heeft op de beslissingen, gevolgd door het attribuut *Pclass*. Naarmate de stappen vorderen, neemt het belang van het attribuut *Sex* af en neemt die van *Pclass* toe. Hoewel er een daling is voor het attribuut *Sex*, blijft het attribuut de meeste invloed hebben op de beslissingen.

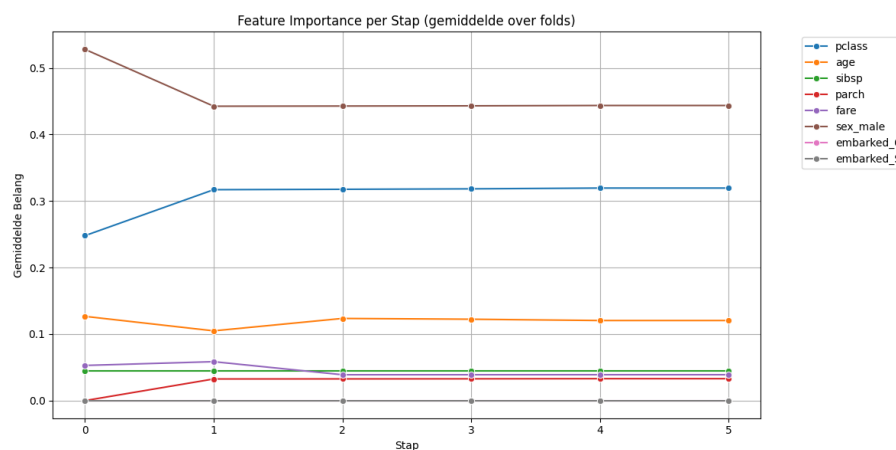


Figure 170: Feature Importance per stap (5 stappen)

5.4.6 Beslissingsbomen toegepast op de supervised K-Means

Voor de beslissingsbomen wordt er verder gewerkt met de traindata van de Titanic dataset. De traindata stelt 80% van de dataset voor, de overige 20% is de testdata die pas later wordt gebruikt. In figuur 171 worden de beslissingsbomen van stap 0 tot en met 5 weergegeven. In stap 0 worden de beslissingsbomen gemaakt met de onverplaatste data. Elke fold heeft evenveel splitsingen. De splitsingen die voorkomen in de bomen hebben een relatief grote waarde, wat aangeeft dat de splitsingen breed zijn. Het verplaatsen van de datapunten zorgt ervoor dat de beslissingswaarden dalen en dus voor effectievere splitsingen zorgen. Dit is duidelijk te zien op de tweede splitsing die gebeurt op basis van het attribuut *Pclass*. In de eerste twee stappen heeft de voorwaarde een waarde van 0,75. Naarmate de punten verplaatsen, daalt de beslissingswaarde naar 0,656. Dit geeft aan dat de splitsingen minder breed en effectiever worden.

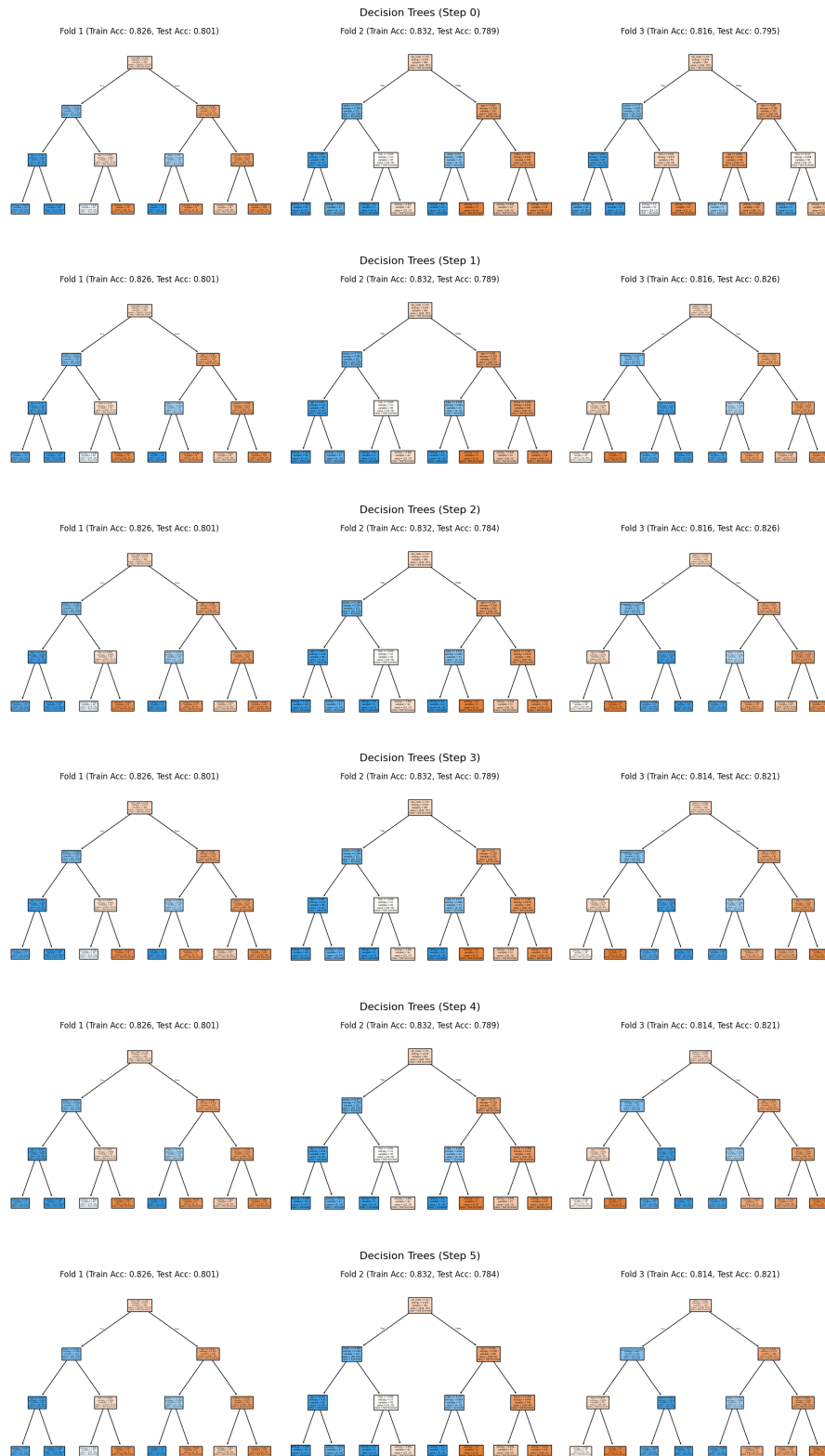


Figure 171: Beslissingsbomen toegepast op de supervised K-Means per Stap (5 stappen)

Verder worden de gemiddelde train- en testaccuracies per stap van de bomen weergegeven in figuur 172. Uit die figuur blijkt dat beide accuracies voor de stappen ongeveer hetzelfde blijven. De waarde van de testaccuracy ligt ongeveer rond de 80% en die van de trainaccuracy bedraagt 82%. In vergelijking met de accuracies van de gewone beslissingsbomen is er weinig verschil. Weliswaar ligt de accuracy

van de bomen toegepast op de supervised K-Means methode een klein beetje hoger.

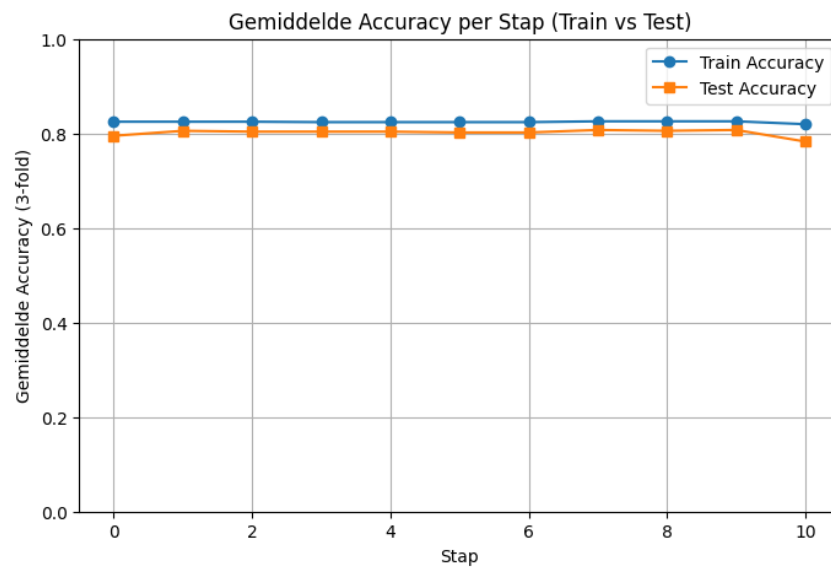


Figure 172: Gemiddelde accuracy beslissingsbomen toegepast op de supervised K-Means (10 stappen)

Ten slotte wordt er in figuur 173 de feature importances weergegeven. Uit die grafiek blijkt dat, net als bij de gewone beslissingsbomen het attribuut *Sex* het meest dominant is voor de stap zonder verplaatsing. Na de verplaatsing neemt de invloed van het attribuut *Pclass* toe. De stijging bij de beslissingsbomen toegepast op de supervised K-Means is een beetje feller. Na de eerste stap hebben de attributen *Sex* en *Pclass* ongeveer hetzelfde belang. Dit blijft zo doorgaan tot en met stap 9. Na stap 9 is er een felle stijging in het belang van het attribuut *Fare* te zien, terwijl bij de gewone beslissingsbomen alles constant bleef na de eerste stap.

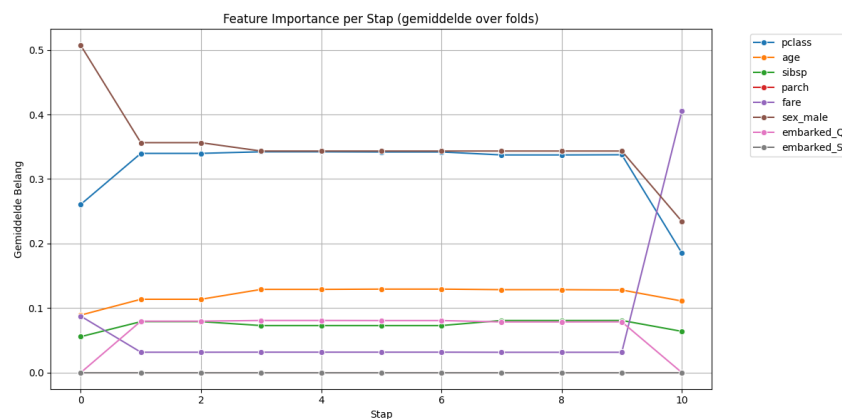


Figure 173: Feature Importance per stap (5 stappen)

6 Discussie

In dit onderzoek werd de nieuwe *feature engineering* methode, namelijk de preprocessingtechniek, geïntroduceerd alsook geëvalueerd. De techniek is, zoals eerder vermeld, nieuw en origineel waar nog geen gelijkaardig literatuuronderzoek voor bestaat. Bij deze aanpak worden de datapunten iteratief naar hun dichtstbijzijnde centroiden verplaatst. De focus van dit onderzoek lag dus voornamelijk op de preprocessingtechniek die op verschillende supervised- als unsupervised machine learning-algoritmen werd toegepast. De resultaten toonden aan dat die techniek, het verplaatsen van de datapunten naar hun dichtstbijzijnde centroiden, effecten heeft. Die effecten waren vooral te zien bij de K-Means- en KNN-modellen. De bestaande, alternatieve *feature engineering*-methoden, namelijk discretisatie of binning, normalisatie en standaardisatie, featureselectie en featurecreatie, zijn technieken die de representatie van attributen gaan wijzigen of enkele relevante attributen gaan selecteren. Terwijl de preprocessingtechniek de ligging van de datapunten zelf gaat wijzigen. Hierdoor is de nieuwe en originele methode anders dan de alternatieven. In plaats van de structuur te wijzigen, worden de inputwaarden gewijzigd door die naar de clustercentroiden te verplaatsen. Met dit onderzoek wordt dus een duidelijke lacune binnen het *feature engineering*s domein gevuld.

Naarmate de datapunten verplaatst werden, werd vastgesteld dat voor de K-Means clusteringmethode een consistente verbetering is in de kwaliteit van de clusters. Zowel de silhouette scores als de WSS-scores verbeterden systematisch, wat op beter gedefinieerde, compacte clusters wijst. Die trend werd op alle vier de datasets waargenomen, hoewel de mate van verbetering varieerde, afhankelijk van de grootte van de datasets. Voor de supervised K-Means clusteringmethode waren er identieke patronen herkenbaar. Weliswaar waren er specifieke gevallen waarbij de silhouette scores ongeveer hetzelfde bleven na verplaatsing. Dit waren enkele configuraties bij de USArrests en Titanic datasets bij de supervised K-Means clusteringmethode. Bijkomend werden per dataset de cluster aantallen verhoogd om het effect verder te analyseren. Uit die verhoging blijkt dat de prestatie-indicatoren hetzelfde gedrag vertonen. Ook was er te zien dat een hoger cluster aantal bij de supervised K-Means clusteringmethode ervoor zorgde dat de silhouette scores ongeveer hetzelfde bleven als bij geen verplaatsing. Het doen toenemen van de cluster aantallen had een duidelijk effect op de standaardafwijkingen binnen clusters. De waarden van de standaardafwijkingen lagen lager bij een groter cluster aantal. Dit komt doordat de clusters veel minder datapunten bevatten en dus kleiner zijn. Die kleinere clusters hadden dan op zich ook datapunten die minder verspreid waren binnen de cluster zelf. Ook was er sprake van een slechtere clusterdefinitie, want de silhouette scores lagen lager bij een groter cluster aantal. Anderzijds was er overlap tussen de clusters te zien, wat ervoor zorgde dat de datapunten van de clusters minder goed gedefinieerd zijn.

Voor het KNN-algoritme resulteerde verplaatsing van de datapunten naar hun dichtstbijzijnde centroiden in hogere classificatie-accuracies. Echter was er op te merken dat de stijging niet lineair bleef, maar na enkele stappen een piek bereikte, gevolgd door stagnatie of een lichte daling. Dit gaf aan dat er een optimum bestaat in het aantal verplaatsingen, waarbij te veel verplaatsing mogelijks leidt tot het verliezen van informatie. Verder was er ook visueel te zien dat de datapunten verder weg van de scheidingslijn gingen staan en er dus een grote opening tussen beide klassegrenzen ontstond. Bij de KNN-algoritmen die op de supervised K-Means werden toegepast, kenden de testaccuracies weinig of geen stijging vergeleken met de waarde van geen verplaatsing. De waarden bleven constant. Weliswaar was er op de KNN-visualisaties te zien dat de scheidingslijn van deze methode meer lineair ging worden. De trainaccuracies van die modellen daarentegen stegen onmiddellijk naar 100% wat

aangeeft dat de modellen perfect de traintdata leren. In sommige gevallen werd de kloof tussen train- en testaccuracy groter, wat aangeeft dat het model beter presteert op de data waarop het geleerd heeft, dan op ongeziene data. Indien de kloof vergroot, is er sprake van overfitting.

De reactie van de beslissingsbomen op de preprocessingstechniek was complexer dan K-Means en KNN. De structuur van de bomen werd eenvoudiger en consistentier na verplaatsing. Dat was te zien door de vermindering van het aantal splitsingen en de aanpassing van de attributen en de beslissingswaarden waarop de splitsingen gebeuren. Echter leidde dit allemaal niet tot hogere accuracies. Vervolgens ging in sommige datasets de accuracy van de bomen dalen, maar die daling was niet zo groot. Zo was de accuracy van de Iris dataset voor stap 0 gelijk aan 96% en daalde naar 92% voor stap 5. Om een verdere analyse uit te voeren namen de stappen toe naar 10 en was er daar onmiddellijk op te merken dat de accuraatheid van de bomen terug stegen naar 97%. Voor de USArrests dataset bleef de accuraatheid van het model tussen de 96% en 98%. De Wine dataset daarentegen kende een stijging tot 98%. Ten slotte was er een zeer lichte stijging op te merken van 79,6% naar 80% bij de Titanic dataset. Er was dus eigenlijk nauwelijks een wijziging in de accuraatheden van de modellen. In het algemeen kenden de testaccuracies van de beslissingsbomen toegepast op de supervised K-Means clusteringmethode, nauwelijks een stijging of daling. De accuraatheid van de Iris dataset bleef op 80%. Voor de USArrests datasets was er een stijging op te merken van de accuraatheid van 70% in stap 0 naar 80% in stap 10. De accuraatheden van de Wine dataset schommelden tussen 80% en 90% en bereikte in stap 10 een hoeveelheid van 89%. Voor de Titanic dataset bleef de accuraatheid constant op 80%, met enkele lichte dalingen naar 79% in sommige stappen. Wat er voor alle vier de datasets opviel was dat de kloof tussen train- en testaccuracies in sommige stappen kleiner werd. Dit geeft aan dat er minder overfit plaatsvindt en dat het model de data beter gaat definiëren in de zin van generaliseerbaarheid. Ook was er te zien dat de kloof nooit groter werd dan de kloof bij stap 0. Ten slotte had de verplaatsing van de datapunten een effect op de feature importances van bepaalde attributen. Een terugkerende vaststelling is dat bepaalde attributen dominanter werden na enkele stappen, ten koste van de invloed van andere attributen. Zo werd het duidelijk dat bij de Iris-dataset het attribuut *Petal_length* domineerde. Dat kwam dus ook vaak terug in de splitsingen van de bomen. Naarmate de stappen vorderden, nam de invloed van *Petal_length* af en kreeg het attribuut *Sepal_length* meer gewicht. Voor de Wine dataset ging het attribuut *Proline* sneller domineren bij de beslissingsbomen toegepast op de supervised K-Means dan de gewone bomen. Weliswaar had het attribuut *Flavanoids* ongeveer een even grote invloed als het attribuut *Proline*. Bij de Titanic dataset viel op dat het attribuut *Sex* voor een lange tijd domineerde. De andere attributen stegen na enkele stappen ten koste van de invloed van het dominerende attribuut. Bij de beslissingsbomen toegepast op de supervised K-Means kwam de variabele *Fare* ter sprake na enkele stappen van verplaatsing.

In vergelijking met de resultaten van de alternatieve *feature engineering*-methoden, besproken in Sectie 3.4, zijn er duidelijke gelijkenissen te zien met de resultaten van de preprocessingstechniek die werd uitgevoerd en besproken in Sectie 5. In het werk van Zhang (2021) en Bonilla (2023) leidde zowel featureselectie, via SelectKBest, en featurecreatie tot een stijging van de accuraatheid. Voor de featureselectie, toegepast op de Iris dataset, bleven de relevante attributen *Petal_length* en *Petal_width* behouden en steeg de accuraatheid van 91% naar 96%. Bij de featurecreatie van de attributen *FamilySize* en *IsAlone* in de Titanic dataset was er ook een verbetering op te merken in de accuraatheid van de modellen. De accuraatheid steeg van 81% naar 86%. De toepassing van de KNN- en K-Means-methode bij de preprocessingstechniek resulteerde in een gelijkaardige prestatieverbetering. Bijkomend werd er bij K-Means ook naar de wss, silhouette scores en standaardafwijkingen gekeken. Die resulteerden ook in toenemende cijfers. Wat betreft de beslissingsbomen was er noch een stijging, noch een daling op te merken bij zowel discretisatie of binning als de preprocessingstechniek (Zaidi et al., 2017; Paghdal, 2025).

Beide technieken zorgden ervoor dat de structuur van de bomen verbeterde en beter interpreteerbaar werd. Discretisatie zorgde ervoor dat het beslissingsboomalgoritme vergemakkelijkte, dankzij de categorische intervallen. Die intervallen resulteerden in effectievere splitsingen en betere datadefiniering (Zaidi et al., 2017; Paghdal, 2025). De preprocessingtechniek zorgde ook voor een betere datadefiniering en werden de splitsingcondities veel smaller. Daarnaast kregen de bomen ook minder splitsingen, wat ervoor zorgde dat de interpretatie vergemakkelijkte (Rokach & Maimon, 2005).

Hoewel de resultaten van de preprocessingtechniek op de verschillende machine learning-algoritmen overwegend positief zijn, brengt dit onderzoek enkele uitdagingen met zich mee. Eerst en vooral zijn de gebruikte datasets voor de experimenten klein en goed gestructureerd. Met andere woorden bevatten de datasets weinig tot geen ruis of ontbrekende waarden, wat de generaliseerbaarheid naar complexere datasets onduidelijk maakt. Bovendien vereisen de experimenten een featureselectie, waarbij enkele variabelen willekeurig worden gekozen via de *SelectKBest*-methode, terwijl dit systematischer aangepakt kan worden door datasetafhankelijke featureaantallen vooraf te bepalen. Dit kan ervoor zorgen dat relevante features niet meer worden weggelaten of dat irrelevante features mee in het model worden genomen.

Ten tweede werden enkel traditionele, interpreteerbare modellen zoals K-Means, KNN en beslissingsbomen gebruikt. Die modellen laten toe dat het effect van de preprocessingtechniek zichtbaar en interpreteerbaar is, terwijl complexere modellen buiten beschouwing worden gelaten. Complexere modellen zoals Support Vector Machines (SVM) of DBSCAN scoorden positief op alternatieve *feature engineering*-methoden, beschreven in Sectie 3.4 (Schubert et al., 2017; Carlsen and Bruggemann, 2020; Zhou, 2021; Jo, 2021). Daarnaast is SVM interessant om toe te passen waarbij de schaal en positie van de attributen van belang zijn (Aggarwal & Reddy, 2013).

Ten derde zijn er in dit onderzoek verschillende *feature engineering*-methoden gecombineerd in beperkte mate. Zo wordt er voor de K-Means-methode de preprocessingtechniek toegepast op genormaliseerde data en worden de beste features van de Titanic en Wine dataset geselecteerd voor de machine learning methoden om de prestaties van de modellen te verbeteren. In toekomstig onderzoek zou de discretisatie- of binningtechniek gecombineerd kunnen worden voor de beslissingsbomen (Han and Kamber, 2006; Bonilla, 2023; Paghdal, 2025).

Ten slotte bestaat er nog tot op heden geen wetenschappelijke literatuur, waardoor dit een methodologische uitdaging vormt. Er ontbreken theoretische of empirische referenties, waardoor het vergelijken, valideren en onderbouwen van de resultaten bemoeilijkt. Toekomstig onderzoek zou deze techniek daarom verfijnen door het toe te passen op realistischere, complexe datasets en door de aanpak uit te breiden naar meer geavanceerde algoritmen. Daarnaast is er een nood aan theoretische verdieping en het opstellen van een kader (framework) waarin de logica van de preprocessingtechniek bestudeerd kan worden. Op die manier kan deze aanpak toegepast worden op een grotere schaal en kan de techniek verder onderbouwd en gevalideerd worden door literatuurreferenties te gebruiken.

Verder kan toekomstig onderzoek zich richten op de optimale combinaties van *feature engineering*-methoden die afstemmen op de complexiteit van de datasets (Han and Kamber, 2006; Zhang, 2021; Bonilla, 2023).

References

- Aggarwal, C. C., & Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. CRC Press LLC.
- Bonilla, J.-E. (2023). Feature Engineering 101: Surviving the Titanic with Data Science.
- Carlsen, L., & Bruggemann, R. (2020). The Iris Dataset Revisited – a Partial Ordering Study. *Informatica*, 44(1), 35–44.
- Chakraborty, A., Faujdar, N., Punhani, A., & Saraswat, S. (2020). Comparative Study of K-Means Clustering Using Iris Data Set for Various Distances, 332–335.
- Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient *k*NN classification algorithm for big data. *Neurocomputing*, 195(1), 143–148.
- El Morr, C., Jammal, M., Ali-Hassan, H., & El-Hallak, W. (2022). *Machine Learning for Practical Decision Making: A Multidisciplinary Perspective with Applications from Healthcare, Engineering and Business Analytics* (Vol. 334). Springer International Publishing.
- Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster Analysis*.
- Han, J., & Kamber, M. (2006). *Data mining: Concepts and techniques* (2nd ed).
- Jaiswal, A. K. (2025). Beginner's Tutorial to Titanic using Scikit-learn.
- Jo, T. (2021). *Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning*.
- Jolly, K. (2018). *Machine Learning with Scikit-Learn Quick Start Guide: Classification, Regression, and Clustering Techniques in Python*.
- Kuo, C.-F. J., Huang, C.-C., Yang, C.-H., & Chen, S.-H. (2021). Integration of multivariate control charts and the decision tree classifier to determine the faults of the quality characteristic(s) of a melt spinning machine used in polypropylene fiber manufacturing. Part II: The application of multivariate control charts and the decision tree classifier to determine the faults of quality characteristic(s). *Textile Research Journal*, 91(21-22), 2567–2580.
- Lee, W.-M. (2019). *Python Machine Learning*. Wiley.
- Load_wine. (2025).
- Lu, S., & Braunstein, S. L. (2014). Quantum decision tree classifier. *Quantum Information Processing*, 13(3), 757–770.
- McNeil, D. R. (2025). USArrests function - RDocumentation.
- Nandan, M., Raj Gupta, H., & Mondal, M. (2023). Building a Classification Model based on Feature Engineering for the Prediction of Wine Quality by Employing Supervised Machine Learning and Ensemble Learning Techniques, 1–7.
- Paghdal, S. (2025). Encoding Numerical Data with Discretization and Binning.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., & Thirion, B. (2011). Scikit-learn: Machine Learning in Python. 12(1), 2825–2830.
- Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., da F Costa, L., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLoS One*, 14(1), 210–236.
- Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers - a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4), 476–487.
- Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.*, 42(3), 19:1–19:21.
- Singh, P. (2019). *Machine Learning with PySpark: With Natural Language Processing and Recommender Systems*.
- Trost, C. O. W., Žák, S., Schaffer, S., Walch, L., Zitz, J., Klünsner, T., Leitner, H., Exl, L., & Cordill, M. J. (2025). Explainable machine learning and feature engineering applied to nanoindentation data. *Materials & Design*, 253, 113897.
- Varoquaux, G., & Grobler, J. (2025). The Iris Dataset.

-
- Vasconcellos, E. C., de Carvalho, R. R., Gal, R. R., LaBarbera, F. L., Capelato, H. V., Frago Campos Velho, H., Trevisan, M., & Ruiz, R. S. R. (2011). DECISION TREE CLASSIFIERS FOR STAR/GALAXY SEPARATION. *The Astronomical Journal*, 141(6), 189.
- Xing, W., & Bei, Y. (2020). Medical Health Big Data Classification Based on KNN Classification Algorithm. *IEEE Access*, 8, 28808–28819.
- Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2), 165–193.
- Zaidi, N. A., Du, Y., & Webb, G. I. (2017). On the Effectiveness of Discretizing Quantitative Attributes in Linear Classifiers.
- Zhang, J. (2021). Data Engineering: A Feature Selection Example with the Iris Dataset.
- Zhou, Z.-H. (2021). *Machine Learning*. Springer.