UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Benchmarking machine learning performance on small datasets

Mike Vilters

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

PROMOTOR :

dr. Frank VANHOENSHOVEN

2024
2025

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Benchmarking machine learning performance on small datasets

Mike Vilters

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

PROMOTOR :

dr. Frank VANHOENSHOVEN

Benchmarking machine learning performance on small datasets

Mike Vilters
Promotor: dr. Frank Vanhoenshoven

Universiteit Hasselt, Faculteit Bedrijfseconomische Wetenschappen, Agoralaan Gebouw D, België

Abstract
Artificiële Intelligentie (AI) is bijna niet meer weg te denken uit de huidige samenleving. Ook KMO's willen AI inzetten om zo meer datagedreven en operationeel efficienter te worden [25]. Het probleem bij KMO's is echter dat zij een gelimiteerde beschikbaarheid van data hebben om AI modellen mee te trainen [2, 58]. Daarom is het doel van deze masterproef om een benchmark te bekomen van de performantie van machine learning (ML) modellen in een small-data scenario. Om dat doel te bereiken wordt de performantie van drie ML-modellen op kleine datasets (100 tot 1000 trainingsamples) gemeten. Ook wordt de effectiviteit van zes verschillende data-augmentatietechnieken voor het verbeteren van modelprestaties geëvalueerd. Daarnaast is er ook een dimensiereductiealgoritme getest om te controleren of het aantal features ook impact had op de prestaties van de ML-modellen. De studie gebruikte drie state-of-the-art ML-algoritmen en vergelijkt de prestaties van zes data-augmentatiemethoden op twee datasets: de CDC Diabetes dataset en de Student Dropout dataset [32]. Voor elke trainingsgrootte worden 10 steekproeven genomen van de volledige dataset, en de modellen worden geoptimaliseerd en geëvalueerd met behulp van een stratified KFold cross-validation. De performantiemetrieken die werden vergeleken zijn: accuracy, precision, recall en F1-score. Om een vergelijking te maken tussen de performantie van de ML- modellen op kleine datasets en grote datasets worden er ook telkens populatiemodellen getraind. De populatiemodellen worden getraind met 70% van de beschikbare data in de volledige dataset. De resultaten tonen aan dat de ML- modellen getraind op de kleine datasets vergelijkbare resultaten kunnen halen met ML-modellen getraind op de grote datasets. De augmentatietechnieken die de verkleinde datasets verrijken, geven amper tot geen verbetering in de F1-scores van de ML-modellen getraind op de diabetesdataset. Principal Component Analysis (PCA) blijkt bij de diabetesdataset wel effectief, de gemiddelde F1-score stijgt met 20 procentpunten. Op de studentendataset werkt SVMSMOTE, BorderlineSMOTE, SMOTE en ADASYN op het RF- en XGB-model om de gemiddelde F1-scores te verbeteren ten opzichte van de baseline. De stijging in F1-score was echter zeer gering, amper 1%. Bij het LogReg model werkte geen enkele augmentatietechniek.

Keywords: · small sample learning · small-data scenario · small datasets · tabular data augmentation · virtual sample generation · few-shot learning · data augmentation

1 Introductie

Artificial Intelligence (AI) is tegenwoordig een populair onderwerp, voornamelijk door de opkomst van Generative AI en Large Language Models. AI kent verschillende vormen en toepassingen [1]. Zo kan het worden toegepast in fysieke objecten, zoals robots en machines, maar ook in digitale systemen [1]. Een veelvoorkomende digitale toepassing is bijvoorbeeld de inzet van hypergepersonaliseerde chatbots voor klantenservice [21]. Daarnaast kan AI ook ingezet worden in private toepassingen, zoals het voorspellen van frauduleuze transacties of het voorspellen van de omzet van een bedrijf voor het volgende kwartaal. Voorgaande voorbeelden zijn eerder een toepassing van Machine Learning (ML), een onderdeel van het bredere AI. Ook dat wordt steeds meer ingezet om bepaalde taken te automatiseren en slimmere, datagedreven beslissingen te maken. Dat is mogelijk geworden door onder andere de opkomst van Big Data [4, 51]. Ook steeds meer KMO's willen datagedreven worden en hun data inzetten om op termijn een grotere operationele efficiëntie te behalen [25]. Op dit moment bestaan er echter verschillende barrières om AI en ML in te zetten bij KMO's [35]. KMO's hebben vaak niet het kapitaal hebben om grote technologische investeringen te maken. Andere veelgenoemde barrières zijn ook het gebrek aan kennis over AI binnen de organisatie, evenals het verzet en scepticisme ten opzichte van AI, omdat sommigen vrezen dat het vooral zal leiden tot sociaal onwenselijke resultaten (i.e. veel ontslagen) [34, 35]. Dat zorgt ervoor dat ze waardevolle opportuniteiten met betrekking tot AI/ML soms mislopen [17]. Verder zijn er ook bedrijven die veel minder sceptisch staan tegenover AI en

ML. Deze bedrijven hebben echter nood aan voldoende kwalitatieve data om AI/ML-modellen te kunnen trainen zodat deze patronen kunnen herkennen die niet vooraf hoeven geprogrammeerd te worden [47]. AI-modellen leren namelijk van grote hoeveelheden voorbeelden, vaak duizenden tot miljoenen datapunten. Een veelvoorkomend probleem bij KMO's is echter dat zij niet over dergelijke grote datasets beschikken [2, 58]. In de studie van Nyiri et al. (2023) [39] worden verschillende problemen besproken voor het werken met kleine datasets in ML. Een van de problemen die wordt aangehaald is dat beperkte data vaak weinig variatie bevat, waardoor ML-modellen belangrijke inherente kenmerken van de data kunnen missen. Daarom is het doel van deze paper om na te gaan of ML toch ingezet kan worden indien er beperkte hoeveelheid data is. Ook worden verschillende data-augmentatietechnieken toegepast om te onderzoeken of die de prestaties van de ML-modellen kunnen verbeteren. Op die manier kan een richtlijn bekomen worden voor kleinere Belgische bedrijven en KMO's, zodat zij AI op een effectieve manier kunnen implementeren. Dat kan hen in staat stellen om een datagedreven organisatie te worden, hun bedrijfsprocessen te optimaliseren, kosten te verlagen en op langere termijn hun omzet en winst te verhogen [17].

Deze paper levert de volgende bijdrages:

1. Een overzicht bieden van de belangrijkste leeraparadigma's die bij machine learning met kleine datasets worden toegepast, en de onderlinge samenhang daartussen in kaart brengen.
2. Een benchmark opzetten om de prestaties van ML modellen getraind op beperkte datasets te evalueren en te bepalen of deze gelijkaardig presteren als modellen getraind op grote datasets.
3. Aangeven of de performantie van ML-modellen getraind op kleine datasets verbeterd kan worden indien er synthetische data toegevoegd wordt aan de datasets

De eerste bijdrage wordt gerealiseerd via een exploratieve literatuurstudie die de leeraparadigma's voor small-data scenario's toelicht en aan elkaar relateert. Voor de tweede en derde bijdrage wordt een experiment uitgevoerd op twee benchmarkdatasets, waarbij de volgende onderzoeksvragen beantwoord worden:

- **OV1:** Kunnen ML-modellen getraind met kleine trainingsets een gelijkaardig resultaat behalen als ML-modellen getraind op grote datasets?
- **OV2:** Kunnen kleine trainingsets verrijkt met behulp van data-augmentatietechnieken betere prestaties behalen dan trainingsets zonder synthetische data?

De rest van de paper is gestructureerd als volgt: sectie 2 toont op basis van een exploratieve literatuurstudie verschillende paradigma's binnen de wetenschappelijke literatuur om effectieve AI/ML modellen te ontwikkelen wanneer er gelimiteerde data aanwezig is. In sectie 3 wordt de methodologie van de experimentele setup beschreven. Sectie 4 toont de resultaten van het experiment. In sectie 5 worden de resultaten besproken en wordt er een antwoord geformuleerd op de onderzoeksvragen. In sectie 6 wordt er een conclusie geformuleerd. Als laatste haalt sectie 7 de limitaties van dit onderzoek aan en worden er enkele onderzoeksdirecties geformuleerd die gevolgd kunnen worden door toekomstige onderzoekers.

2 Gerelateerd onderzoek

Het doel van deze sectie is het uitvoeren van een exploratieve literatuurstudie naar bestaande technieken en methoden voor machine learning bij kleine datasets. Eerst wordt er dieper in gegaan op Small Sample Learning (SSL), met nadruk op de paradigma's concept learning en experience learning [50]. Daarnaast wordt Few-Shot Learning (FSL) besproken, waarbij transfer learning en meta learning verder uitgediept worden [52]. Vervolgens wordt er een uitgebreid overzicht gegeven van diverse data-augmentatiestrategieën. In de studie van Mohamad et al. (2020) [33] werd vastgesteld dat de 53 onderzochte Britse KMO's voornamelijk over gestructureerde data beschikten. Tabulaire, gestructureerde data blijft daarmee een belangrijke databron binnen KMO's. Daarom wordt in sectie 2.4 verder ingegaan op Tabulaire Data Augmentatie (TDA). Verder komen semi-supervised en self-supervised learning methoden aan bod, evenals een bespreking van de impact die modelarchitectuur en hyperparameterinstellingen hebben op de prestaties van machine learning modellen. Tot slot zullen enkele toepassingen van machine learning in situaties met inherent kleine datasets worden toegelicht. Deze sectie biedt een globaal overzicht van de huidige kennis en ontwikkelingen hieromtrent.

Voor gedetailleerdere informatie over de besproken technieken wordt verwezen naar de aangehaalde bronnen. Vanwege de uiteenlopende terminologieën die gehanteerd worden in de verschillende leerparadigma's rondom small data, werd figuur 1 ontwikkeld. Deze flowchart is gebaseerd op de geanalyseerde literatuur en brengt de verschillende begrippen die in de verdere literatuurstudie voorkomen met elkaar in verband. Op die manier biedt de figuur een overzichtelijk kader van de bestaande paradigma's voor het toepassen van machine learning op kleine datasets.

2.1 Small sample learning (SSL)

Shu et al. (2018) [50] voerden een uitgebreide studie uit naar small sample learning en identificeerden daarin twee belangrijke onderzoeksstromen: **experience learning** en **concept learning**.

Bij **experience learning** leren machine learning modellen op basis van eerdere ervaringen. Dat omvat technieken zoals data-augmentatie, meta learning, en transfer learning, die allemaal reeds bestaande en beschikbare ervaring gebruiken om nieuwe patronen te leren uit beperkte gegevens.

- **Data augmentatie** genereert aanvullende, sterk correlerende data om de bestaande voorbeelden te versterken (zie sectie 2.3).
- **Transfer learning** maakt gebruik van eerder getrainde modellen die ervaring hebben met vergelijkbare taken, en past deze aan op de nieuwe taak (zie sectie 2.2.2).
- **Meta-learning** richt zich op het snel aanpassen aan nieuwe voorbeelden door te leren hoe een model effectief leert (zie sectie 2.2.1).

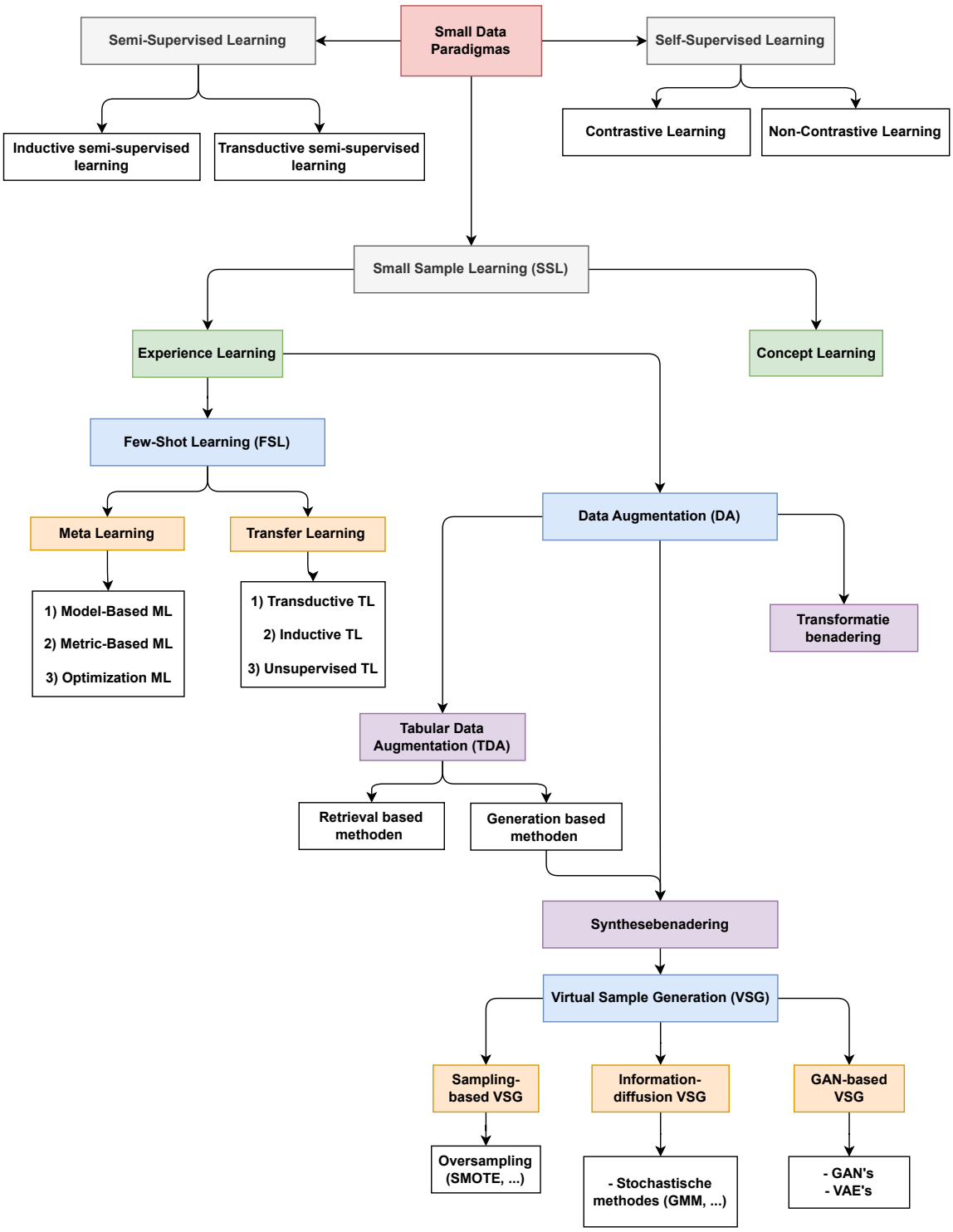
Bij **concept learning** ligt de focus op het leren van nieuwe concepten en het ontdekken van verbanden uit zeer beperkte gegevens. Een essentieel onderdeel van concept learning is het conceptsysteem, waarin elk concept (klasse) zowel een intensionele representatie als een extensionele representatie heeft [50]. Onder een intensionele representatie valt bijvoorbeeld de beschrijving van de kenmerken van een bepaald concept of de gehanteerde definitie. Bij een extensionele representatie gaat het om concrete voorbeelden van het concept. Vervolgens wordt een matchingregel toegepast om deze concepten te koppelen aan de beschikbare data, waardoor het model in staat is om nieuwe concepten te leren of bestaande concepten te herkennen [50]. Voorbeelden hiervan zijn zero-shot learning, waarbij een model concepten kan herkennen zonder dat er enige trainingsdata voor handen is, en one-shot learning, waarbij er getraind wordt op basis van slechts één datapunt [50].

2.2 Few-shot Learning

Few-Shot Learning (FSL) is een relatief nieuwe benadering om met een beperkte hoeveelheid trainingsdata om te gaan. Deze methode maakt gebruik van meta-learning (sectie 2.2.1) en transfer learning (sectie 2.2.2) om modellen te trainen die in staat zijn patronen te herkennen, zelfs met zeer weinig datapunten [52]. FSL wordt ook toegepast in Deep Learning (DL) modellen, vanwege de brede toepasbaarheid van FSL in verschillende domeinen. Zo heeft Rather et al. (2024) [46] een studie gevonden waar een DL FSL anomaliedetectiemodel beter presteerde dan state-of-the-art methoden op zes benchmark datasets. FSL biedt om zijn brede toepasbaarheid daarom ook potentieel. Desalniettemin blijft overfitting een probleem, omdat een FSL model blootgesteld is aan een heel beperkt deel van de datadistributie [19]. Daarom is er verder onderzoek nodig naar manieren om FSL en DL-modellen robuuster te maken tegen overfitting. Data-augmentatie wordt door Rather et al. (2024) [46] als mogelijke manier voorgesteld om overfitting gedeeltelijk te verminderen. Het is echter geen perfecte methode en biedt geen garantie dat een model, door een grotere blootstelling aan de datadistributie, goed zal kunnen generaliseren [46]. Data-augmentatie is tevens ook een veelgenoemde methode om FSL-problemen aan te pakken [19, 62, 70]. Er wordt dieper ingegaan op data-augmentatie in sectie 2.3.

2.2.1 Meta learning

In verschillende studies [19, 39, 43, 52, 62, 70] wordt meta-learning geïdentificeerd als oplossing voor de problemen bij small sample learning en few-shot learning. Meta learning heeft als doel om een model te



Figuur 1: Overzicht van verschillende leerparadigma's met betrekking tot small data.

ontwikkelen wat zeer flexibel is ingesteld. Zo wordt het initiele model opgezet om zich snel te kunnen aanpassen als het een nieuwe taak met nieuwe data moet uitvoeren. Op die manier moet een model maar een paar nieuwe voorbeelden hebben alvorens het al vrij snel kan generaliseren naar nieuwe voorspellingen. In essentie is dus het doel van meta-learning "leren om te leren"[15]. Traditioneel worden bij meta learning de klassieke train- en testsplits van een dataset nog eens onderverdeeld in twee sets. Zo ontstaat er een voor zowel de train- als testsplit een support en query set [11]. De train-support set wordt dan gebruikt om een meta learner te trainen waarbij deze op de train-query set zijn predicties zal maken om zo zijn juistheid te evalueren en zichzelf te updaten. Daarna wordt er een classifier opgebouwd aan de hand van de test-support set, die geëvalueerd zal worden aan de hand van de test-query set [11]. De test-support set gebruikt de informatie die geleerd is door de meta-learner in de vorige fase om de classifier te trainen [11]. Op die manier kan er van weinig voorbeelden gebruik gemaakt worden om een effectieve classifier te bouwen. De manier waarop de meta learner leert en zichzelf update wordt in de literatuur voornamelijk onderscheden door drie soorten meta learning: Model-based meta learning, metric-based meta learning en optimization-based meta learning [15, 43].

Model-based meta learning focust zich op het snel aanpassen aan nieuwe taken door middel van een interne geheugenstructuur. In deze context wordt een taak gedefinieerd als het vinden van een functie $f(x)$ waarbij de features (x) worden gerelateerd aan het target (y) volgens $y = f(x)$. Op basis van deze functie kunnen voorspellingen worden gedaan van y gegeven een input x. Het model zal terwijl het getraind wordt op een specifieke taak, de kennis van deze taak opslaan in zijn intern geheugen [61]. Dat zal het ook doen bij nieuwe taken. Op basis van de kennis van de taken die op deze manier verzamelt worden, kan een model zichzelf snel aanpassen aan nieuwe taken, en zo ook met weinig voorbeelden nieuwe taken generaliseren [61].

Metric-based meta-learning richt zich op het leren van een interne afstandsmetriek, die ervoor zorgt dat een ML-model eenvoudig gelijkaardige classificatietaken kan herkennen en zich snel kan aanpassen aan nieuwe taken. Een model dat een nieuwe taak ontvangt, gebruikt deze interne afstandsmetriek om te bepalen met welk type classificatietaak de metriek het meest overeenkomt, en classificeert vervolgens volgens dat systeem.

Optimization-based meta-learning zorgt ervoor dat door middel van een optimalisatiealgortime de parameterinitialisatie zo wordt geoptimaliseerd dat het model, bij het ontvangen van voorbeelden van een classificatietaak, snel kan convergeren naar de optimale waarden voor die specifieke taak [61].

2.2.2 Transfer Learning

Bij transfer learning wordt er gebruik gemaakt van vooraf getrainde ML-modellen: de bronmodellen. Deze bronmodellen zijn getraind op data die vergelijkbaar is met de nieuwe doeldata. Het voordeel hiervan is dat het bronmodel al blootgesteld is aan een rijkere en mogelijk bredere datadistributie, soms over verschillende domeinen heen [72]. De doeldataset is de kleine dataset die doorgaans wel beschikbaar is voor het trainen van ML-modellen. Die kan dan gebruikt worden om het voorgetrainde model aan te passen aan de huidige context. Er zijn verschillende soorten transfer learning. Zo is er transductive transfer learning, inductive transfer learning en unsupervised transfer learning [72].

Bij **transductive learning** is er gelabelde data beschikbaar in het brondomein, maar niet in het doeldomein. Het doel is om de kennis dat een machine learning model heeft verworven in het brondomein over te dragen naar het doeldomein voor dezelfde taak, maar met andere data. Hierbij probeert het model voorspellingen te maken op basis van zijn eerdere ervaring, zonder dat het specifieke voorbeelden heeft gezien van de nieuwe data in het doeldomein. De te voorspellen targetvariabele blijft wel hetzelfde tussen het bron- en doeldomein.

In **inductive learning** is er gelabelde data beschikbaar in zowel het bron- als doeldomein. Hierdoor kan het model zich effectief aanpassen aan de nieuwe taak in het doeldomein, omdat de eerder opgedane kennis van het brondomein het model helpt om het label in het doeldomein te voorspellen. Aangezien de taak in het doeldomein enigszins lijkt op die in het brondomein, kan het model snel en efficiënt leren van de nieuw gelabelde data in het doeldomein.

In **unsupervised transfer learning** is er geen gelabelde data beschikbaar in zowel het bron- als doeldomein. In deze situatie probeert men de kennis die een model in het brondomein heeft verworven in te zetten om het model op een slimme manier te laten leren in het doeldomein. Er wordt bijvoorbeeld geprobeerd om met behulp van informatie uit het brondomein data in het doeldomein te clusteren of om de dimensies in het doeldomein te reduceren [42]. Volgens Niu et al. (2020) [38] is dit onderzoeksdomein echter redelijk

beperkt, gegeven dat het moeilijk toe te passen is op realistische cases. Feature based learning is volgens Niu et al. (2020) [38] één van de weinige onderzoeksstromen binnen unsupervised transfer learning. Self-taught clustering staat hier centraal. Hierbij wordt geprobeerd om de features van de verschillende domeinen in éénzelfde feature space te mappen, zodat de data uit het brondomein kan worden gebruikt om de data in het doeldomein te clusteren [38].

2.3 Data-augmentatie (DA)

Een belangrijk probleem met kleine datasets is de gevoeligheid voor overfitting [46]. Dat gebeurt wanneer een model niet alleen de patronen in de data leert, maar ook de (willekeurige) ruis erin. Hierdoor kan het model goede resultaten behalen op de trainingsdata, maar is het minder goed in het maken van nauwkeurige voorspellingen op nieuwe data. Het model is dus weinig generaliseerbaar. Daarnaast kan een kleine dataset leiden tot een oververtegenwoordiging van bepaalde klassen en een ondervertegenwoordiging van anderen, wat resulteert in modellen die biased kunnen zijn. Om het probleem van beperkte beschikbaarheid van data aan te pakken, zijn er in de literatuur verschillende methoden geïdentificeerd om synthetische data te genereren. Een van de voorgestelde methoden is data-augmentatie.

2.3.1 Transformatieve- & synthesebenadering

Data-augmentatie is een veelgebruikte methode om een bestaande dataset uit te breiden en tegelijkertijd de diversiteit ervan te vergroten [70]. Mumuni et al. (2022) [36] onderscheiden twee benaderingen binnen data-augmentatie: de transformatiebenadering en de synthesebenadering. Bij de transformatiebenadering wordt bestaande data gemanipuleerd, bijvoorbeeld door middel van heuristiek-gebaseerde methoden. Nyiri et al. (2023) [39] beschrijven dergelijke technieken, waarbij specifieke heuristieken worden toegepast om natuurlijke variaties in de data te creëren. Deze aanpak sluit nauw aan bij de transformatiebenadering zoals gedefinieerd door Mumuni et al. (2022) [36]. De synthesebenadering daarentegen richt zich op het genereren van volledig nieuwe data. Het doel van data-augmentatie is dat het ML-model blootgesteld kan worden aan een groter deel van de onderliggende datadistributie en daardoor beter zal kunnen generaliseren [46, 70]. Het is tevens wel belangrijk om bij het toepassen van data-augmentatie een juiste en doordachte strategie te selecteren, aangezien niet elke aanpak geschikt is voor elke dataset [26].

2.3.2 General Adverserial Networks

Een belangrijke techniek die veel in de literatuur gebruikt wordt die onder de synthesebenadering valt, zijn de Generative Adverserial Networks (GAN's) [13]. Een GAN is een generatief neuraal netwerk dat nieuwe datapunten kan genereren op basis van de oorspronkelijke trainingsdata. Het model bestaat uit twee neurale netwerken die tegelijkertijd getraind worden: de generator en de discriminator. De generator probeert nieuwe data te creëren, terwijl de discriminator leert het verschil te herkennen tussen de gegenereerde data en de echte data [39]. Naarmate de training vordert, wordt de generator steeds beter in het produceren van realistische data, tot het ingezet kan worden als een effectieve data-augmentatietechniek. Uit onderzoek van Rather et al. (2024) [46] blijkt dat deze aanpak vaak effectiever is dan traditionele data-augmentatietechnieken. Een GAN kan ook toegepast worden binnen self-supervised learning als een datagenerator, mits de te minimaliseren kostenfunctie wordt aangepast [12]. Een uitgebreidere toelichting van self-supervised learning is terug te vinden in sectie 2.6.1. Het is wel van belang om kritisch te blijven bij het gebruik van GAN's. Een GAN is immers ook een ML-model en kan daardoor instabiliteit vertonen wanneer de training suboptimaal gebeurt [64]. Het GAN moet namelijk zelf ook getraind worden met de beperkte data, waardoor het verkrijgen van een kwalitatief GAN niet eenvoudig is [28, 55]. Tran et al. (2021) [55] observeerden in hun studie dat een GAN getraind met een willekeurige subset bestaande uit 25% van de volledige dataset divergeerde en dat er zelfs model collapse optrad. Dat kan bijgevolg resulteren in het genereren van onrealistische data of data met weinig variatie. Het blijft dus van groot belang om de performantie van het GAN te monitoren [18].

2.3.3 DA in computer vision

Eerder onderzoek heeft uitgebreid aandacht besteed aan data-augmentatietechnieken die specifiek betrekking hebben op ongestructureerde data [29, 67]. Zo is er een grote hoeveelheid literatuur beschikbaar over de

toepassing van dergelijke technieken binnen het domein van computer vision [36, 67]. Binnen dit onderzoeksgebied wordt voornamelijk gebruikgemaakt van foto's als databron. Voor foto's bestaan er verschillende methoden om extra afbeeldingen te genereren op basis van de bestaande foto's in een dataset. Zo kunnen foto's bijvoorbeeld worden geroteerd, verkleind, gespiegeld of omgekeerd [67]. Verder kunnen ook de RGB-waarden van foto's veranderd worden om zo verschillende kleurvarianten van foto's te verkrijgen. Ook saturatie en contrast kunnen op deze manier bijgewerkt worden om extra voorbeelden te genereren [36, 46]. Verder kunnen ook kernel filters, zoals Gaussian Blur, die een wazig effect aan foto's geven, of edge filters, die foto's scherper maken, gebruikt worden om extra varianten van foto's te verkrijgen. [46, 49]. Een andere veelgebruikte techniek voor het transformeren van foto's is het willekeurig verwijderen van delen uit een afbeelding, ook wel bekend als *image erasing* [46, 67]. Bij *image erasing* wordt een willekeurig gebied in de foto geselecteerd, waarna de pixelwaarden binnen dat gebied worden vervangen door willekeurige waarden. Dat resulteert in afbeeldingen waarin bepaalde delen ontbreken. Deze methode kan worden vergeleken met de drop-out regularisatie in neurale netwerken [49]. Het verschil is echter dat deze drop-out niet plaatsvindt binnen de architectuur van het model of netwerk zelf, maar in de inputlaag [46]. Het is wel van belang om na te gaan of deze augmentatietechnieken geschikt zijn voor de specifieke use case. Hoewel deze technieken over het algemeen geschikt zijn om de trainingsset van foto's uit te breiden, zijn er uitzonderingen waarmee rekening gehouden moet worden. Zo kunnen translaties en rotaties problemen veroorzaken wanneer het spiegelbeeld van een bepaalde klasse een andere klasse vertegenwoordigt (bijvoorbeeld de cijfers 6 en 9) [49]. Daarnaast kunnen kleuren essentieel zijn voor de herkenning van bepaalde objecten (bijvoorbeeld de kleuren van verkeerslichten) [49]. Daar zou er informatie verloren kunnen gaan als de kleuren van de afbeelding willekeurig veranderen.

2.3.4 Manuele data-augmentatie

Nyiri et al. (2023) [39] stellen dat ook het gebruik van domeinexpertkennis kan leiden tot een diversere dataset. Zo kunnen domeinexperts een taxonomie ontwikkelen voor de binaire klassen die voorspeld moeten worden. Met een taxonomie worden deze twee klassen verder opgedeeld in meer specifieke subcategorieën, waardoor het model in staat is om gedetailleerdere kenmerken uit de trainingsdata te leren. Na het maken van voorspellingen kunnen deze specifieke klassen weer geaggregeerd worden naar de oorspronkelijke binaire klassen. Daarnaast wijzen de auteurs erop dat domeinexperts een belangrijke rol kunnen spelen in *feature engineering*. Door nieuwe, specifieke kenmerken aan de dataset toe te voegen, kan het model leren van de nieuwe eigenschappen van de data die voorheen niet beschikbaar waren, wat de nauwkeurigheid en effectiviteit van het model kan verbeteren [39].

2.4 Tabular Data Augmentation (TDA)

In sectie 2.3 worden voornamelijk data-augmentatietechnieken behandeld die gericht zijn op ongestructureerde data, zoals tekst en foto- of videodata. Tabular Data Augmentation (TDA) vormt echter een afzonderlijke onderzoeksstroom die specifiek gericht is op het ontwikkelen van geschikte methoden om tabulaire of gestructureerde datasets te verrijken met synthetische data. Onderzoek naar data-augmentatietechnieken voor ongestructureerde data komt veelvuldig voor en is uitgebreid onderzocht. Onderzoek naar data-augmentatie bij tabulaire data is veel minder voorkomend [10]. Er zijn veel studies waarin TDA slechts een deel van het grotere geheel van de studie is en waarin het onderwerp slechts kort wordt behandeld [5].
Cui et al. (2024) [5] hebben hier verandering in gebracht en hebben een uitgebreide studie uitgevoerd naar methoden en technieken binnen TDA. De auteurs onderscheiden onder andere twee typen methoden voor TDA: retrieval-based methoden en generative-based methoden. **Retrieval-based methoden** maken gebruik van externe tabellen die semantisch gerelateerd zijn aan de oorspronkelijke dataset om deze samen te kunnen voegen en zo mogelijks te verrijken. **Generative-based methoden** maken gebruik van generatieve AI om nieuwe synthetische data te creëren die de verdeling van de oorspronkelijke data benadert, zodat deze als realistische gegevens kunnen worden beschouwd [10].

Verder heeft Cui et al. (2024) [5] TDA voor beide typen methoden gedefinieerd op vier verschillende niveaus: kolomniveau, rijniveau, celniveau en tabelniveau. Hoewel beide typen methoden de dataset op elk niveau op een andere manier verrijken, blijven de conceptuele definities van de niveaus gelijkaardig voor beide benaderingen. Bij TDA op **kolomniveau** worden extra features toegevoegd aan de dataset, die door ML-modellen kunnen worden gebruikt om van te leren. Op **rijniveau** worden extra datapunten toegevoegd aan de oorspronkelijke dataset, waardoor het totale aantal datapunten toeneemt. Op **celniveau** worden

individuele velden in de tabel aangevuld; met andere woorden, missende feature-waarden voor specifieke samples worden aangevuld. Tot slot worden op **tabelniveau** zowel extra kolommen (features) als extra rijen (datapunten) toegevoegd aan de dataset [5].

2.4.1 Retrieval-based methoden

Retrieval-based methoden benaderen tabellen door elementen van deze tabel, zoals de instanties of features, te transformeren naar vectoren [5]. Er zijn verschillende benaderingen om een tabel te representeren en te transformeren in een vector [69]. Volgens Cui et al. (2024) [5] zijn er drie representatiemethoden: content-, context- en metadata-representatie. Deze drie methoden analyseren de rijen, kolommen, metadata en cellen van tabellen om een geaggregeerde weergave te creëren, die geschikt is voor transformatie naar een vector. De specifieke details van deze representatiemethoden vallen buiten de scope van deze literatuurstudie. Voor een uitgebreide beschrijving wordt verwezen naar de studie van Cui et al. (2024) [5]. Door tabellen in een latente vectorruimte te representeren volgens één van de bovengenoemde methodes, kan het verwantschap tussen tabellen kwantitatief worden uitgedrukt [5]. Ook in de survey van Zhang et al. (2020) [69] wordt onderzoek gedaan naar methoden om de verwantschappen tussen tabellen te kwantificeren. Deze stap wordt door de auteurs benoemd als "*table matching*". Door inherente kenmerken van tabellen, zoals kolomnamen en tabelwaarden, te transformeren naar vectorruimtes, kan een semantische gelijkenis tussen tabellen worden vastgesteld [5]. In Zhang et al. (2020) [69] wordt onder andere verwezen naar methoden die gebruikmaken van afstandsmetingen tussen deze vectoren. Hierbij geldt dat hoe dichter bepaalde vectoren bij elkaar liggen in de vectorruimte, hoe groter de gelijkenis tussen bron- en doeltabel. Dergelijke doeltabellen worden dan ook als potentiële kandidaten beschouwd voor het verrijken van de gegeven brontabel.

2.4.2 Generation-based methoden

Generation-based methoden maken gebruik van generatieve technieken om nieuwe data te synthetiseren op basis van de oorspronkelijke dataset [5]. Er wordt dus een vorm van Virtual Sample Generation (VSG) toegepast (zie sectie 2.5). Voor tabulaire of gestructureerde data is het echter complexer om realistische nieuwe samples te genereren dan voor ongestructureerde data. Dat komt onder andere door het heterogene karakter van tabulaire data [60]. Elke tabel heeft een unieke structuur, verschillende datatypes (zoals numerieke en categorische variabelen), en uiteenlopende datadistributies per variabele [31]. Daarnaast hebben er ook andere factoren een invloed op de complexiteit van tabulaire datageneratie. Sommige datasets mogen bijvoorbeeld niet publiekelijk beschikbaar worden gesteld vanwege wettelijke vereisten of privacybeperkingen, zoals in de medische of financiële sector [5, 22]. Bovendien kan het genereren van individuele datapunten zeer computationeel en tijdsintensief zijn, zoals in toepassingen binnen de ingenieurswetenschappen [22]. Tabulaire datasets hebben vaak ook datakwaliteitsproblemen, zoals ontbrekende waarden [31] en klasse-imbalans [30]. Om een oplossing te bieden voor deze uitdagingen wordt in de praktijk gebruikgemaakt van generatieve methoden om de dataschaarste en de datakwaliteitsproblemen bij tabulaire gegevens te overbruggen.

In de huidige literatuur rond Tabular Data Generation (TDG) zijn er ook een aantal TDG technieken en methoden ontwikkelt om tabulaire datasets te kunnen verrijken. Fonseca en Bacao (2023) [10] hebben in hun studie een taxonomie ontwikkeld op basis van een uitgebreide literatuurstudie, waarbij veelgebruikte terminologieën worden geconsolideerd. Deze taxonomie classificeert methoden aan de hand van verschillende kenmerken: **(1) de architectuur van de methode**, **(2) het toepassingsniveau van de methode**, waarbij een onderscheid wordt gemaakt tussen externe methoden (toegepast vóór de primaire ML-taak) en interne methoden (geïmplementeerd tijdens de primaire ML-taak), **(3) de scope van de methode**, waarbij een lokale benadering verwijst naar technieken die zich baseren op de dichtstbijzijnde buren van een specifieke observatie om nieuwe datapunten te genereren, terwijl een globale benadering gebruikmaakt van de statistische eigenschappen van de volledige dataset om synthetische datapunten te genereren, en **(4) de dataspace** waarop het generatieve model wordt toegepast [10].

Kim et al. (2024) [22] hebben een tijdlijn opgesteld rondom technieken en onderzoek naar generatieve methoden. Ze hebben ook een opsplitsing gemaakt in ML-gebaseerde modellen, en stochastische modellen. In het begin waren er voornamelijk stochastische en generatieve technieken op basis van traditionele ML modellen zoals support vector machines (SVM's) en decision trees (CART). Stochastische modellen maken gebruik van de statistische verdelingen van een dataset om synthetische data te genereren [10, 22]. ML-gebaseerde methodes genereren nieuwe samples door een willekeurige instantie te creëren met arbitraire

attribuutwaarden, waarna het ML-model toegepast wordt om de bijbehorende output voor deze instantie te voorspellen [22]. Stochastische methodes komen overeen met probability-gebaseerde architectuur uit [10], terwijl ML-gebaseerde methode eerder onder de random architectuur vallen. Vanaf 2010 begonnen de deep learning modellen hun intrede te maken in TDG [22]. In het begin werd DL op dezelfde manier ingezet als de traditionele ML-modellen. Na verloop van tijd zijn hier gespecialiseerde modellen uitgekomen die gebruik maken van DL, en die als doel hadden om data te genereren. Een voorbeeld hiervan is de Variational Auto Encoder (VAE) [10]. Een VAE is een model dat datapunten encodeert door middel van een encoder. Die encoder zal datapunten encoderen in een latente representatieruimte waar datapunten een probabilistische vectorrepresentatie krijgen. De decoder kan dan uit deze latente representatieruimte, nieuwe datapunten samplen die dicht bij de originele distributie liggen. Naast VAE's kwamen ook GAN's (zie sectie 2.3.2) op als geavanceerde DL-methoden om nieuwe tabulaire datapunten te genereren.

Traditionele diepe generatieve modellen (i.e. GAN's en VAE's) zijn niet ontworpen om data te genereren met conditionele attributen, zoals een target label [10]. Daarom zijn er diverse en gespecialiseerde diepe generatieve methodes ontwikkeld die geschikt zijn voor de generatie van tabulaire gegevens [60]. Zo zijn er gespecialiseerde varianten van GAN's en VAE's ontwikkeld voor tabulaire datasets [60]. Twee frequent voorkomende TDG-methoden in de literatuur zijn de Conditional Tabular GAN (CTGAN) en de Tabular VAE (TVAE) [66]. CTGAN is ook een geschikte methode voor ongebalanceerde tabulaire datasets vanwege zijn training-by-sample strategie [5, 66]. Dat zorgt ervoor dat bij het genereren van synthetische samples de proportionele verdeling van de originele klassen wordt behouden, waardoor zowel de meerderheids- als de minderheidsklasse representatief worden gegenereerd [60]. In tegenstelling tot oversampling technieken herstelt CTGAN klasse-imbalans niet, maar zorgt het wel voor een evenredige generatie van samples in overeenstemming met de oorspronkelijke klasseverdeling. Verder zijn er in de literatuur uitbreidingen en modificaties geweest van deze originele methodes [10, 60]. Deze zijn gespecialiseerd in het type gegevens dat zij genereren, zoals Medical GAN (MEDGAN) [14], een model dat specifiek gericht is op het synthetiseren van medische data. Daarnaast zijn bepaalde methodes ontwikkeld voor een specifieke functie, DPGAN's [65], die synthetische data genereren met behoud van differentiële privacy. Dat stelt gebruikers van DPGAN's in staat om data te synthetiseren zonder dat individuele samples uit de oorspronkelijke dataset kunnen worden achterhaald. Dat is waardevol in toepassingen waarin gevoelige gegevens een rol spelen [65].

2.5 Virtual Sample Generation (VSG)

Wen et al. (2025) [64] hebben een uitgebreide studie gedaan naar virtual sample generation (VSG). Dat zijn methodes om de trainingsets van machine learning tasks uit te breiden en te diversifieren. De auteurs hebben de verschillende methodes onderverdeeld onder drie categoriëen: Sampling-based VSG, information diffusion-based VSG en Generative adverserial networks-based VSG.

- **Sampling-based VSG**: Sampling-based VSG-methoden en algoritmes hebben als doel nieuwe virtuele datapunten te genereren op basis van bestaande gegevens in een dataset. Deze aanpak maakt voornamelijk gebruik van interpolatie en technieken zoals oversampling en undersampling. In de studie wordt specifiek aandacht besteed aan de SMOTE-oversamplingtechniek en haar varianten. Hierbij is het belangrijk om op te merken dat over- en undersamplingtechnieken vooral worden toegepast om ongelijke klassenverdelingen (klasse-imbalans) te corrigeren. Hoewel deze methoden nieuwe datapunten creëeren, is het primaire doel vaak om het aantal datapunten per klasse gelijk te maken, wat kan helpen bij het verbeteren van de modelprestaties.
  Daarnaast wordt de Gaussiaanse verdeling gebruikt als basis voor het genereren van nieuwe datapunten. Deze methoden maken gebruik van eerdere kennis die aanwezig is in de oorspronkelijke dataset. Een belangrijke bevinding uit de studie van Wen et al. (2025) [64] is dat de prestaties van modellen variëren afhankelijk van het aantal gegenereerde datapunten. Meer gegenereerde data leidt niet automatisch tot betere resultaten, omdat deze datapunten bias kunnen bevatten. Indien er zich een toename van biased data voordoet zal dat leiden tot een cumulatieve bias, wat de generaliseerbaarheid van het model negatief beïnvloedt.
- **Information diffusion-based VSG**: Deze methoden zijn gebaseerd op de correlatie en overeenkomsten tussen verschillende datapunten. Door de statistische eigenschappen van de data, zoals spreiding en centralisatie, te analyseren, wordt geprobeerd de verspreiding van de data binnen de dataset te simuleren.

Op basis van deze gesimuleerde verspreiding genereren de methoden realistische extra datapunten. Hierbij wordt gebruik gemaakt van de onderlinge statistische relaties tussen de datapunten om nieuwe datapunten te creëeren die consistent zijn met de oorspronkelijke data [64].
- **Generative adverserial networks-based VSG**: Deze variant van VSG maakt gebruik van GAN's om synthetische datapunten te genereren. Zoals besproken in sectie 2.3.2, bieden GANs uitgerbeide mogelijkheden om nieuwe voorbeelden te genereren, maar kennen ook hun limitaties.

Sutojo et al. (2023) [53] onderzochten de mogelijkheid om de performantie van een corrosiewerend middel te voorspellen door de chemische eigenschappen van de stoffen in het middel te analyseren. In hun studie vergeleken ze vijf verschillende ML-algoritmen. Deze ML-modellen werden geselecteerd op basis van eerdere literatuur.

Om de uitdagingen van de beperkte hoeveelheid data aan te pakken, gebruikten de auteurs VSG. Door de kleine dataset is er sprake van een oneven verdeling van de gegevens en grote verschillen tussen de waarden van verschillende datapunten. VSG genereerde via interpolatie extra synthetische datapunten, wat leidde tot een meer gelijkmatige verdeling van de gegevens en kleinere verschillen tussen de datapunten. De resultaten, gevalideerd op zes kleine datasets (met een bereik van 11-69 datapunten) met betrekking tot de voorspelling van de prestaties van corrosiewerende middelen, toonden aan dat het kNN-model met VSG de hoogste accuraatheid behaalde van alle vergeleken modellen.

Verder is het belangrijk te benadrukken dat niet alle VSG-methoden even effectief zijn in combinatie met elk ML-model. Wanigasekara et al. (2018) [63] onderzochten de invloed van deze combinaties en toonden aan dat de keuze van zowel de VSG-methode als het ML-model bepalend is voor de prestaties. In hun studie werden drie veelgebruikte VSG-methoden vergeleken. Deze methoden werden toegepast in combinatie met vijf verschillende ML-technieken [63]. Het uiteindelijk resultaat was dat een Artificieel Neuraal Netwerk (ANN) met de Trend Similarity Assessment (TSA) VSG methode het meest effectief was.

2.6 Self- & semi-supervised learning

Self-supervised learning en semi-supervised learning zijn twee benaderingen binnen machine learning die gebruikmaken van zowel gelabelde data (supervised learning) als ongelabelde data (unsupervised learning). Deze methoden zijn conceptueel vrij gelijkaardig, toch zijn er wel enkele subtiele verschillen [12]. Beide streven ernaar om zoveel mogelijk informatie te halen uit de beperkte hoeveelheid beschikbare gelabelde data en deze vervolgens aan te vullen met ongelabelde data als extra bron van informatie voor het model [39]. Daardoor zijn deze technieken geschikt voor situaties waarin slechts een beperkte hoeveelheid gelabelde data beschikbaar is, maar er wel toegang is tot een grote hoeveelheid ongelabelde data.

2.6.1 Self-supervised learning

Bij self-supervised learning leren modellen zelf data te genereren [39]. Dat gebeurt met behulp van twee modellen die elk verantwoordelijk zijn voor hun eigen taak. Er worden doorgaans twee soorten taken onderscheiden: pretext tasks en downstream tasks [18, 45]. Pretext tasks zijn taken die een ML-model uitvoert om pseudolabels te genereren op basis van de intrinsieke kenmerken van een gegeven sample [12]. De pretext taak zorgt voor de context, maar is niet verantwoordelijk voor het einddoel. De pseudolabels worden vervolgens doorgegeven aan het downstream ML-model, dat verantwoordelijk is voor de downstream task: de doeltaak van het model. Hierdoor kan het model, zonder manuele labeling, zelfstandig leren om specifieke taken uit te voeren [12, 45]. Rani et al. (2023) [45] beweren echter dat een kleine hoeveelheid manuele labeling nodig is om te starten om een performant self-supervised learning model te bekomen.

Ook worden er binnen self-supervised learning verschillende types geïdentificeerd waaronder contrastive learning en non-contrastive learning [45]. Bij contrastive learning leert een ML-model onderscheid te maken tussen sterk gelijkende samples en sterk verschillende samples [18]. In een computer vision toepassing zou bijvoorbeeld een geaugmenteerde versie van een afbeelding binnen dezelfde klasse worden beschouwd als een positieve sample, terwijl alle andere afbeeldingen als negatieve samples beschouwd worden [18]. Zo zou een originele foto van een dier en de geroteerde foto van datzelfde dier een positieve sample zijn, en een foto van een ander dier een negatieve sample. Op deze manier wordt het model getraind om representaties te leren die gelijkenissen en verschillen tussen samples weergeven. Contrastive learning kan zelf verder onderverdeeld

worden in verschillende types, maar dat valt buiten de scope van deze paper. Voor meer info hierover wordt verwezen naar de studies van Jaiswal et al. (2024) [18] en Gui et al. (2024) [12]
Naast contrastive learning is er ook non-contrastive learning, waarbij het model enkel getraind wordt met positieve samples en op basis daarvan leert om representaties van de onderliggende klassen te vormen. Hoewel dit type self-supervised learning de indruk kan wekken dat het model vatbaar is voor model collapse, waarbij het te veel traint op eerder gesynthetiseerde data en daardoor slecht zal presteren op ongeziene data [6], merken Rani et al. (2023) [45] op dat non-contrastive modellen desondanks toch in staat zijn om goede representaties van de klassen te leren.

2.6.2 Semi-supervised learning

Bij semi-supervised learning worden ML-modellen getraind met zowel gelabelde als ongelabelde data [59]. De gelabelde data wordt gebruikt om het model te leren hoe het labels moet voorspellen, terwijl de ongelabelde data extra context en waardevolle intrinsieke informatie kan bieden [39]. Dat maakt het mogelijk om, naast de kleine gelabelde dataset, ook een grotere en relevante ongelabelde dataset te benutten. Binnen semi-supervised learning kunnen twee hoofdtypen onderscheiden worden: inductive semi-supervised learning en transductive semi-supervised learning [40].

Bij **inductive semi-supervised learning** is het doel een ML-model te ontwikkelen dat in staat is te generaliseren en nieuwe instanties correct te classificeren [71]. De interne optimalisatie binnen een dergelijke inductieve taak richt zich op het generalisatievermogen van de resulterende classifier [59]. Veelgebruikte methodes voor inductive semi-supervised learning zijn wrapper methodes [59] of proxy-labeled methodes [40]. Dat zijn twee verschillende termen in de literatuur die hetzelfde proces omschrijven. Binnen die methodes wordt een supervised model getraind op de beschikbare en gelimiteerde gelabelde data. Vervolgens wordt dat model toegepast op de ongelabelde data om pseudolabels te genereren [40, 59]. Deze pseudogelabelde data wordt daarna opnieuw aan het supervised model gegeven, waar het behandeld wordt als correct gelabelde trainingsdata. Zo kan een classifier bekomen worden die beter generaliseert naar nieuwe en ongeziene data.

Transductive semi-supervised learning of transductieve interferentie [71] heeft als doel de ongelabelde datapunten waarop het semi-supervised learning model getraind heeft van een label te voorzien [40, 71]. Deze heeft dus niet het doel om een supervised classifier te worden dat kan generaliseren naar volledig ongeziene datapunten. Concreet wordt er voor een ongelabeld datapunt X^O een voorspeld label $\hat{y_u}$ gegenereerd [59]. In dit geval wordt de optimalisatie uitgevoerd op de kwaliteit van $\hat{y_u}$ die aan X^O toegekend wordt [59]. Van Engelen et al. (2020) [59] vermelden dat graafgebaseerde methoden ideaal zijn voor transductieve inferentie. In deze aanpak worden de datapunten gerepresenteerd als een graaf, waarbij elke node overeenkomt met een individueel datapunt [40]. De onderlinge afstanden tussen de datapunten worden berekend op basis van hun attributen en vervolgens gebruikt om de edges van de graaf te voorzien van gewichten [59]. Op basis van de labels van de beschikbare gelabelde datapunten en de gewogen afstanden binnen de graaf, worden labels toegewezen aan de ongelabelde datapunten. Dat proces is gebaseerd op het principe dat datapunten die zich dicht bij elkaar bevinden waarschijnlijk hetzelfde label delen. Voorgaand principe is zelfs een belangrijke assumptie waaraan voldaan moet zijn binnen semi-supervised learning en staat bekend als de smoothness assumption of cluster assumption [40, 59, 71]. Dat is tevens ook de assumptie waarop het kNN-classificatiealgoritme op gebaseerd is.

2.7 Smart sampling, architectuur en hyperparameter manipulatie

Smart sampling richt zich op het efficiënt verzamelen van data die gebruikt wordt om modellen te trainen. Nyiri et al. (2023) [39] bespreken methoden zoals over- en undersampling waarbij extra observaties artificieel worden toegevoegd of juist verwijderd om de klassenbalans in de dataset te verbeteren. Een andere mogelijke techniek is importance sampling [39]. Hierbij wordt een nieuwe datadistributie gecreëerd waarin zeldzame gevallen minder zeldzaam worden. Datapunten worden dan uit deze verdeling gehaald, en er worden ook gewichten toegekend aan het datapunt om te corrigeren voor mogelijks ingevoerde bias. Active learning wordt ook als techniek genoemd. Hierbij selecteert men alleen de datapunten die het meeste informatie opleveren, bijvoorbeeld gemeten aan hun entropiewaarde. Zo kan het model efficiënter leren van de geselecteerde datapunten [39].

Ook de instellingen van hyperparameters zijn van essentieel belang bij de ontwikkeling van ML-modellen. Een groot deel van de prestaties van een model wordt bepaald door de keuze en configuratie van deze

hyperparameters [16]. Het is daarom belangrijk om een vorm van hyperparameteroptimalisatie toe te passen bij elk model. In een small-data scenario moet hier zorgvuldig mee omgegaan worden gegeven dat overfitting in deze scenario's zeer snel kan optreden. Op deze manier kunnen de hyperparameters optimaal afgestemd worden op zowel het model als de bijbehorende dataset. Bij het trainen van (diepe) neurale netwerken moet er een architectuur bepaald worden. Hyperparameters zoals het aantal hidden layers, het aantal nodes per laag, de loss function, de learning rate en andere hyperparameters moeten vooraf vastgelegd worden alvorens een neuraal netwerk kan beginnen met leren [68]. Dat impliceert dat de architectuur van een diep neuraal netwerk beschouwd kan worden als een verzameling hyperparameters die geoptimaliseerd moeten worden.

Rather et al. (2024) [46] voerden een uitgebreide literatuurstudie uit naar technieken die in deep learning toegepast kunnen worden om beter om te gaan met kleine datasets. Zij onderzochten onder meer hoe de hyperparameters en de architectuur van diepe neurale netwerken de prestaties op dergelijke datasets beïnvloeden. Een voorbeeld van een cruciale hyperparameter is de loss function, een functie waarmee het model zijn eigen prestaties tijdens het trainingsproces evalueert [68]. Een veelgebruikte loss function is Binary Cross Entropy. Rather et al. (2024) [46] bevinden echter dat de cosine loss function op kleine datasets betere prestaties kan leveren wat betreft de accuraatheid. Ook de de activatiefuncties van de verschillende hidden layers kunnen een impact hebben op het classificatievermogen van het Deep Neural Network (DNN). In dezelfde studie van Rather et al. (2024) [46] wordt de orthogonale softmax-laag (OSL) geïntroduceerd. De OSL zorgt ervoor dat de gewichten in de laatste laag van het neuraal netwerk orthogonaal gemaakt worden. Hierdoor zal het model beter zijn in het onderescheiden van verschillende klasses. Door de OSL te gebruiken in plaats van de gebruikelijke softmax activatiefunctie wordt er ook een hogere accuraatheid bekomen [46].

Shaikhina et al. (2015) [48] hebben in hun onderzoek een raamwerk ontwikkeld voor het toepassen van machine learning op biomedische toepassingen waarvoor weinig data beschikbaar is. De eerste fase van hun raamwerk is gebaseerd op het vinden van de optimale architectuur van het desbetreffende ML-model. Het raamwerk omvat twee onderdelen. Ten eerste richten zij zich op het verminderen van de *randomness* die gepaard gaat met de initialisatie van parameters in neurale netwerken en decision trees, evenals de *randomness* in het trainingsproces van modellen. Dat is van belang omdat de generaliseerbaarheid van modellen die op kleine datasets zijn getraind te afhankelijk is van deze willekeurige initialisatie. Om dat probleem te voorkomen, werden meerdere versies van zowel neurale netwerken als decision trees getraind, waarbij de meest optimale architectuur voor elk model geselecteerd werd voor verdere ontwikkeling. Ten tweede introduceren Shaikhina et al. (2015) [48] een validatietechniek voor regressietaken op basis van beperkte data. Hierbij maken ze gebruik van synthetische data die de algemene verdeling van de werkelijke data volgt, zonder de inherente eigenschappen van die data te weerspiegelen. Op deze manier hebben ze een benchmark gecreëerd voor de modellen die getraind worden op de echte data, waarvan verwacht wordt dat deze beter zouden presteren vanwege blootstelling aan de inherente kenmerken van de werkelijke data.

Het raamwerk wordt toegepast op zowel een classificatie- als een regressieprobleem. Voor het regressieprobleem wordt een neuraal netwerk gebruikt om de botsterkte bij patiënten met artrose te voorspellen aan de hand van botscans en patiëntinformatie, gebaseerd op een dataset van 35 observaties. Voor het classificatieprobleem wordt met behulp van decision trees het risico op vroege afstoting bij niertransplantaties voorspeld, gebaseerd op een dataset van 80 observaties. De resultaten tonen dat het toepassen van het voorgestelde framework leidt tot een R^2 van 98,3% voor de voorspelling van botsterkte en AUC-score van 85% voor het voorspellen van afstotingsrisico. Die resultaten zijn gelijkaardig of beter dan die van vergelijkbare ML-modellen in het bio-ingenieursdomein [48]. Het neuraal netwerk voor botsterkte behaalde een verbetering in R^2 van 8,7 procentpunten ten opzichte van een gelijkaardig model. Het decision tree model realiseert een AUC-score die vergelijkbaar is met die in andere studies, ondanks dat deze een veel grotere trainingsset gebruiken [48].

Kortom, zowel de architectuur als de hyperparameters kunnen een grote invloed hebben op de prestaties van ML-modellen op kleine datasets. Rather et al. (2024) [46] onderzochten hoe aanpassingen in de architectuur en de optimalisatie van hyperparameters bij neurale netwerken invloed kan uitoefenen op de performantie van het netwerk. In eerder onderzoek legden Shaikhina et al. (2015) [48] de nadruk op het beheersen van de *randomness* die voortkomt uit de parameterinitialisatie van zowel decision trees als neurale netwerken. Door de architectuur van beide modellen te optimaliseren, wilden zij de invloed van willekeurige initialisatie minimaliseren. Daarnaast ontwikkelden zij een methodologie om de prestaties van modellen te evalueren op basis van synthetische data die de algemene verdeling van de werkelijke data volgt, zonder de inherente kenmerken ervan te weerspiegelen. Hiermee onderzochten ze of de geoptimaliseerde modellen beter presteren

met deze synthetische data of dat training op de oorspronkelijke data, hoewel schaarser, beter blijft. Ze concludeerden dat de gemiddelde voorspellingscapaciteit van ML-modellen die op de oorspronkelijke data werden getraind ongeveer 35 procentpunten beter was [48].

2.8 Toepassingen van ML op kleine datasets

In de studie van Feng et al. (2019) [8] worden de prestaties van een vooraf getraind diep neuraal netwerk (DNN), getraind op een kleine dataset (n = 487), vergeleken met die van een geoptimaliseerd shallow neural network en een geoptimaliseerd support vector machine-model. De auteurs proberen in deze studie de solidification cracking susceptibility (SCS) te voorspellen, een materiaaldefect waarbij de kans op scheurvorming tijdens de stollingsfase van een bepaald materiaal wordt gekwantificeerd. De voorspelling van SCS in roestvrij staal werd uitgevoerd op basis van verschillende materiaaleigenschappen van het staal. Voor deze drie ML-modellen worden de hyperparameters geoptimaliseerd om een geschikte architectuur te vinden. Het diep neuraal netwerk wordt bovendien gepretrained met stacked auto-encoders, zodat het DNN start met geoptimaliseerde parameters. Uit de experimenten blijkt dat het gepretrainde diep neuraal netwerk een iets hogere testaccuraatheid bereikte van 93%, vergeleken met 89% voor zowel het geoptimaliseerde shallow neural network als de support vector machine [8].

De studie van Ouyang et al. (2021) [41] vergelijkt drie ML-modellen om de sterkte van verschillende betoncomposities te voorspellen. In het onderzoek worden polynomiale regressie, een (shallow) neural network en een random forest gebruikt. De polynomiale regressie wordt beperkt tot een maximale graad van drie, en het (shallow) neural network heeft slechts één hidden layer. Deze beperkingen maken de modellen minder flexibel om complexe patronen te leren, wat de kans op bias vergroot, maar verminderen ook de kans op overfitting. Het doel van de studie is om vast te stellen hoeveel data elk model nodig heeft om zijn maximale accuraatheid te bereiken en welk model dat het snelst kan doen met de minste hoeveelheid data. Uit de resultaten blijkt dat de polynomiale regressie en het (shallow) neural network, die minder flexibel zijn, minder data nodig hebben om hun maximale accuraatheid te behalen. Deze modellen hebben echter consistent een lagere finale accuraatheid dan het meer flexibele random forest, dat in staat is complexe niet-lineaire verbanden te leren. Het random forest heeft bovendien nog ruimte om zijn validatiescore te verbeteren, wat suggereert dat het random forest nog een hogere finale accuraatheid kan behalen indien het verder wordt blootgesteld aan data [41].

Kim (2024) [23] paste compact data learning toe om aan te tonen dat, door optimalisatie van zowel het aantal features als het aantal samples per feature, een vergelijkbare accuraatheid bereikt kan worden met een sterk gereduceerde dataset ten opzichte van het populatiemodel (bestaande uit 1019 initiële trainingssamples). Compact data learning heeft als doel om ML-modellen efficiënter te trainen zonder verlies van performantie [9, 24]. Kim (2024) [23] ontwikkelt hiertoe een raamwerk op basis van statistische tests en metrieken om datacomplexiteit te verminderen. Daarnaast wordt een statistische methode gebruikt om het optimale aantal trainingssamples per feature te bepalen. Door vervolgens het gemiddelde aantal samples per resterende feature (in dit geval gereduceerd van 222 naar 36 features) te berekenen, wordt de optimale trainingssamplegrootte bepaald. Dat resulteerde uiteindelijk in een trainingssamplegrootte die slechts 13% bedraagt van de oorspronkelijke omvang. Kim (2024) [23] bekomt met het gebruik van dit raamwerk een accuraatheid van 90.1% op een dataset voor het voorspellen van hartritmestoornissen op basis van cardiogrammen, wat vergelijkbaar is met het populatiemodel dat een accuraatheid behaalt van 92,4%.

Käppel et al. (2021) [20] onderzochten hoe technieken van small sample learning (SSL) ingezet kunnen worden om event logs te verrijken, met als doel verschillende process mining-methoden te ondersteunen bij *predictive business process monitoring*. Dat houdt in dat men probeert te voorspellen welke activiteit als volgende zal plaatsvinden in de uitvoering van een procesinstantie. In hun studie passen de auteurs verschillende SSL-methoden uit andere domeinen van machine learning aan, zodat ze beter geschikt zijn voor datasets in het veld van process mining, zoals event logs. Zo wordt een data-augmentatietechniek uit het computer vision veld aangepast om ruis toe te voegen aan event logs, waardoor deze variaties in procesuitvoeringen simuleerd. Daarnaast maken ze gebruik van transfer learning door een neuraal netwerk eerst te trainen op geaugmenteerde data of data van een sterk vergelijkbaar proces. Vervolgens wordt het gepretrainde model verder verfijnd met de beperkte beschikbare uitvoeringen van het doelproces.
Volgens Kappel et al. (2021) [20] moet er voorzichtig omgegaan worden met data-augmentatie in business process management, omdat hieruit mogelijks onmogelijke of foutieve processtromen uit voortstromen. Dat

kan resulteren in onbruikbare ML-modellen aangezien de trainingsdata foutief was. Desondanks slagen de onderzoekers erin om door hun aangepaste augmentatie- en transfer learning-technieken de accuraatheid van next event prediction te verhogen. De precieze verhoging is niet gerapporteerd maar het zou gaan om een verhoging van accuraatheid van 10% tot 15% [20].

3 Experimentele setup

Voor het experimentele deel van deze paper wordt een vergelijkende studie uitgevoerd op twee datasets. Het doel van deze experimenten is om tot een antwoord te komen op de volgende onderzoeksvragen:

- **OV1:** Kunnen ML-modellen getraind met kleine trainingsets een gelijkaardig resultaat behalen als ML-modellen getraind op grote datasets?
- **OV2:** Kunnen kleine trainingsets verrijkt met behulp van data-augmentatietechnieken betere prestaties behalen dan trainingsets die niet verrijkt zijn met gesynthetiseerde data?

Om OV1 te beantwoorden, is er gekozen voor een aanpak waarbij de prestaties van een populatiemodel, dat getraind is op een grote dataset, vergeleken wordt met die van een model dat slechts op een subset van deze data getraind is. Het model dat op de volledige dataset getraind is (het populatiemodel) dient als representatie van de volledige populatie en geldt als ground truth. Met 'getraind op de volledige dataset' wordt bedoeld dat het model getraind is op de 70% trainingsdata van een train/test-split waarbij de gehele dataset gebruikt is om de split te creëeren. Daarna worden er uit deze grote datasets subsamples genomen om een small-data scenario te simuleren. Op deze manier kan er onderzocht worden hoe de prestaties van ML-modellen, getraind op de verkleinde datasets, zich verhouden tot die van de populatiemodellen. Door deze resultaten te vergelijken kan OV1 beantwoord worden.

Om OV2 te beantwoorden wordt er een gelijkaardige methode gebruikt. Nu worden de trainingsets van de verkleinde datasets echter verrijkt met synthetische datapunten van verschillende augmentatiemethodes. Zo zullen de datasets enerzijds met verschillende oversamplingmethodes verrijkt worden en anderzijds ook met enkele augmentatiemethodes die enkel als doel hebben meer datapunten toe te voegen aan de dataset en niet de klasse-imbalans op te lossen. Verder zal er ook een dimensiereductiealgoritme worden geïmplementeerd om te controleren of het aantal features van de datasets een invloed heeft op de performantie van de ML-modellen. De ML-modellen getraind op de onverrijkte datasets worden als baseline beschouwd. Daarna wordt er een statistische analyse uitgevoerd om te kijken of de distributie van de evaluatiemetrieken van de verrijkte datasets hoger ligt dan de distributie van de baseline metrieken. Op basis van die analyse kan er een antwoord geformuleerd worden op OV2.

3.1 Datasets

Om de bovenstaande methodologie te kunnen toepassen, zijn datasets vereist die aan de volgende criteria voldoen:

- De dataset moet voldoende groot zijn
- De dataset moet publiek toegankelijk zijn
- Het moet gaan om een classificatieprobleem
- De dataset moet tabulair van aard zijn

Op basis hiervan zijn twee datasets geselecteerd uit de UCI Machine Learning Repository: de CDC Diabetes Health Indicators dataset (= 'diabetesdataset') en de Student's Dropout and Academic Success dataset (= 'studentendataset') [32]. De studentendataset is oorspronkelijk een multi-class classificatieprobleem. Vanwege de aard van de data is deze getransformeerd naar een binair classificatieprobleem, waarbij één van de klassen weggelaten is en dus niet voorspeld zal worden. De keuze hiervoor wordt gedetailleerd toegelicht in sectie 3.1.2.

3.1.1 CDC Diabetes Health Indicators

De eerste dataset is de CDC Diabetes Health Indicators dataset die als doel heeft te voorspellen of een individu diabetes heeft op basis van diverse demografische en gezondheidskenmerken. Elke kolom in deze dataset vertegenwoordigt een kenmerk (feature) van een persoon, terwijl elke rij een afzonderlijk individu voorstelt. Deze dataset is afkomstig van de UCI Machine Learning repository.
Een kenmerk van deze dataset is de grote klasse-imbalans: ongeveer 86% van de samples betreft personen zonder diabetes, terwijl slechts 14% van de samples betrekking heeft op personen met diabetes. Een traditionele oplossing om de klasse-imbalans te corrigeren, is het werken met over- of undersampling. Echter, in dit onderzoek wordt oversampling beschouwd als een Tabular Data Augmentation (TDA) methode en kan daarom niet worden toegepast als preprocessing stap. Undersampling is eveneens geen optie, aangezien de steekproefgrootte van de gesampelde data al relatief klein is. Verdere reductie zou leiden tot een verlies aan waardevolle informatie.

Om ervoor te zorgen dat beide klassen voldoende representatief blijven tijdens het trainen van de modellen, worden gewichten toegekend aan de klassen. Personen met diabetes krijgen een groter gewicht bij het trainen van de modellen, waardoor wordt voorkomen dat het model een te grote bias ontwikkelt richting de oververtegenwoordigde klasse (personen zonder diabetes). De precieze methode wordt verder toegelicht in sectie 3.3.

3.1.2 Student's Dropout and academic succes dataset

De tweede dataset [32], ook afkomstig uit de UCI Machine Learning Repository, bevat diverse kenmerken en heeft als doel te voorspellen of een student een bepaalde cursus vroegtijdig zal stopzetten vanwege tegenvallende resultaten. Bijgevolg zal dus ook het academisch succes van de student proberen voorspeld te worden. De dataset heeft een omvangrijke set aan features, waaronder socio-economische en demografische factoren, evenals eerdere academische prestaties. Elke kolom vertegenwoordigt een specifieke feature, terwijl elke rij overeenkomt met een individuele student.

Net zoals de vorige dataset is deze gericht op classificatie, maar de doelvariabele omvatte initieel drie klassen in plaats van twee. Daardoor was de studentendataset oorspronkelijk een multi-class classificatieprobleem. De doelvariabele kon drie mogelijke klassen aannemen: 'drop-out', wat betekent dat een student de cursus vroegtijdig heeft verlaten; 'enrolled', wat betekent dat de student nog steeds ingeschreven is voor de cursus en 'graduated', wat duidt op een succesvolle afronding van de cursus door de student.

Vanwege de aard van de data is deze dataset getransformeerd naar een binair classificatieprobleem, waarbij de samples met de klasse 'enrolled' zijn verwijderd. Daardoor bleven enkel de klassen 'graduated' en 'drop-out' over. De reden voor deze transformatie ligt bij de tweede klasse 'enrolled': deze studenten zijn nog steeds ingeschreven voor de betreffende cursus. Deze studenten zullen in de toekomst ofwel slagen, ofwel het vak vroegtijdig verlaten. De 'enrolled'-klasse bevat dus zowel studenten die uiteindelijk zullen slagen, en daardoor eigenschappen hebben die vergelijkbaar zijn met de 'graduated'-klasse, als studenten die zullen uitvallen, met gelijkaardige eigenschappen van de 'drop-out'-klasse. Het behouden van de 'enrolled'-klasse in de dataset kan een bias veroorzaken, omdat dit een tijdelijke status betreft die beide uitkomstgroepen combineert. Dat kan ertoe leiden dat patronen die eigenlijk behoren tot één van de twee eindklassen verkeerd worden toegeschreven aan de 'enrolled'-klasse, wat een vertekening in de werkelijke datadistributie van beide klassen kan veroorzaken.

Na de transformatie van multi-class classificatieprobleem naar een binair classificatieprobleem vertoonde de dataset een matige klasse-imbalans, waarbij 61% van de samples aan de klasse 'graduated' en 39% aan de klasse 'drop-out' toebehoren.

3.2 Data preprocessing

De datasets bestaan uit een groot aantal observaties zoals zichtbaar in tabel 1. Om een realistisch small-data scenario te simuleren werd ervoor gekozen om meerdere subsets van deze data te selecteren. Een inschatting van de hoeveelheid data die beschikbaar is in een KMO ligt tussen de 100 en 1000 datapunten [54]. Dat is in dezelfde ordergrootte waarin Tao et al. (2025) [54] kleine datasets classificeert binnen de material sciences. Daarom werden er subsets van de datasets geselecteerd met telkens een grootte tussen de 100 en 1000 datapunten. Om verder de variatie ten gevolge van de willekeurige selectie van de steekproeven te controleren, werden er voor elke trainingsgrootte tien steekproeven genomen. Dat resulteerde in tien

Dataset	Totaal # Observaties	# Features	Taak	Klasse (im-)balans (%)
CDC Diabetes dataset	253.680	21	Binaire classificatie	86:14
Students drop-out dataset [32]	3.630	36	Binaire classificatie	61:39

Tabel 1: Deze tabel geeft de kenmerken weer van de twee gebruikte datasets. De laatste kolom geeft de verhouding weer in het aantal klasses en is oplopend weergegeven (i.e. diabetes dataset heeft 86% observaties voor klasse 0 en 14% observaties voor klasse 1)

trainingsgroottes met telkens tien steekproeven, wat in totaal 100 datasets opleverde die getraind en getuned werden. De uiteindelijke prestaties worden gerapporteerd als het gemiddelde over de tien steekproeven per trainingsgrootte. De gegevens die per trainingsgrootte niet geselecteerd werden voor training, werden gebruikt als testdata.

Op die manier werd het resulterende model gevalideerd en kon er getest worden op een representatief deel van de oorspronkelijke dataset. Op die manier werd ongeziene data gesimuleerd en kon er nagegaan worden of het getunede model ook generaliseerbaar was. Bovendien werd de oorspronkelijke klassenverdeling in elke steekproef behouden. Zo werd voor elke trainingsgrootte in het interval [100, 200, ..., 1000] een set van tien steekproeven bekomen die elk de intrinsieke klasse-imbalans respecteerde, met als doel een realistisch small-data scenario te simuleren. Door een train/test split te gebruiken werd er gezorgd voor een strikte scheiding van train- en testdata. Er werd voor de trainsplit een absoluut aantal datapunten genomen gelijk aan de huidige trainingsgrootte. De testset is de overige data die niet in de trainingsset zat.

In de datasets waren er zowel numerieke als categorische variabelen aanwezig. Om ervoor te zorgen dat de schaal van de numerieke variabelen het model niet zou beïnvloeden zijn de numerieke variabelen gestandaardiseerd. Om de modellen te tunen, werd er gebruikgemaakt van een randomized, stratified KFold-crossvalidatie. Die methode liet toe om op efficiënte manier te zoeken naar een optimale configuratie van hyperparameters. Voor elke steekproef per trainingsgrootte werden 100 iteraties uitgevoerd, waarbij telkens 10 splits werden toegepast. Volgens Vabalas et al. (2019) [57] kan het gebruik van een gestratificeerde KFold cross-validatie echter bias introduceren, doordat een deel van de testdata mogelijk blootgesteld wordt aan het model tijdens één van de splits. Om die bias te vermijden, werd de testdata buiten het crossvalidatieproces gehouden. Op die manier werd gegarandeerd dat de testdata niet werd blootgesteld aan het model.

3.3 Trainen van ML modellen

In de volgende fase van het experiment werden ML-modellen getraind en geëvalueerd op basis van de steekproeven per trainingsgrootte die tijdens de preprocessingfase gegenereerd werden. Omdat er sprake was van een small-data scenario, werd er gekozen om zowel bagging- als boostingmodellen te gebruiken. Concreet werd hierbij gekozen voor Random Forest (RF) als baggingmethode en Extreme Gradient Boosting (XGB) als boostingmethode.

Beide algoritmen zijn gebaseerd op decision trees. Volgens Uddin en Lu (2024) [56] presteren tree-based modellen goed op tabulaire datasets. RF en XGB zijn echter complexe modellen met een hoge representatiecapaciteit, wat bij een beperkte hoeveelheid trainingsdata kan leiden tot overfitting. Toa et al. (2025) [54] zeggen dat complexe modellen met veel (hyper)parameters gevoelig kunnen zijn aan overfitting in een small-data scenario. Wegens de aard van de methodes en de mogelijkheid om uit zwakkere modellen te leren, worden RF en XGB toch als geschikte algoritmes beschouwd in een small-data scenario.

Om te controleren waaraan mogelijke overfitting ligt: het model of de gelimiteerde trainingdata, werd er ook een eenvoudiger model toegevoegd: logistische regressie (LogReg). Logistische regressie is volgens Tao et al. (2025) [54] een geschikt algoritme voor datasets waarbij de traininginssets kleiner zijn dan 1000 datapunten. In vergelijking met RF en XGB bevat LogReg minder hyperparameters, waardoor de representatiecapaciteit lager ligt. Dat vermindert inherent de kans op overfitting. Toch blijft die mogelijkheid bestaan door het lage aantal datapunten in elke dataset.

Tijdens het configureren van de hyperparameters werd er voor gezorgd dat het ML-model rekening hield met de bestaande klasse-imbalans. Verschillende modellen houden hier op verschillende manieren rekening mee. Bij RF werd ervoor gezorgd dat de parameter 'class_weight' altijd de waarde 'balanced' of 'balanced_subsample' aanneemt. Bij het XGB model wordt een gewicht toegekend aan de verschillende klasses. De positieve klasse is

ondervertegenwoordigt en krijgt daarom een gewicht van vier keer meer dan de negatieve klasse. Dat gebeurt aan de hand van de variabele 'scale_pos_weight'. Bij LogReg wordt ook de variabele 'class_weight' ingesteld op 'balanced'. Het instellen van die hyperparameters zorgde ervoor dat de ML-modellen rekening hielden met de klasse-imbalans bij het leren van patronen tussen de data.

3.4 Augmentatie van de datasets

In de literatuur zijn verschillende methoden terug te vinden met betrekking tot Tabular Data Augmentation (TDA). Binnen dit experiment werd onderzocht of dergelijke technieken een meerwaarde kunnen bieden bij het verrijken van beide datasets, en daardoor ML-modellen hun prestaties tevens ook kunnen verbeteren. In totaal werden zes verschillende methoden toegepast. Vier hiervan zijn oversamplingtechnieken: SMOTE, ADASYN, SVMSMOTE en BorderlineSMOTE. Daarnaast werden ook twee zuivere TDA methoden gebruikt, namelijk de Tabular Variational Autoencoder (TVAE) en de Conditional Tabular GAN (CTGAN).

Gegeven de aanwezige klasse-imbalans in beide datasets, kunnen oversamplingtechnieken een belangrijke rol spelen in het verrijken van de trainingsdata. Daarom zijn in beide datasets de ondervertegenwoordigde klassen aangevuld tot dat hun omvang vergelijkbaar was met die van de oververtegenwoordigde klasse. Dat resulteerde in een gebalanceerde trainingsset waarop de ML-modellen getraind konden worden.

Voor de implementatie van CTGAN en TVAE werd gebruikgemaakt van de implementaties uit de Synthetic Data Vault (SDV) [44]. CTGAN is ontworpen om de oorspronkelijke klassenverdeling van de trainingsdata grotendeels te behouden bij het genereren van synthetische data. Voor TVAE geldt dat niet, hier werd voornamelijk de overgerepresenteerde klasse gesynthetiseerd waardoor een alternatieve aanpak nodig was om alle klasses te vertegenwoordigen.

Voor beide datasets werd er voor TVAE gewerkt met twee afzonderlijke modellen. Het eerste model werd getraind op de distributie van de meerderheidsklasse en genereerde synthetische data voor deze klasse. Het tweede model leerde de distributie van de minderheidsklasse en genereerde overeenkomstige synthetische data. De gesynthetiseerde datasets van beide klassen werden vervolgens samengevoegd met de werkelijke data. Dat resulteerde in één coherente dataset bestaande uit zowel echte als synthetische datapunten. Die methode werd toegepast voor elke trainingsgrootte en voor elk van de tien steekproeven per grootte.

Daarnaast is er ook gebruikgemaakt van een algoritme voor dimensiereductie, namelijk Principal Component Analysis (PCA). PCA genereert geen nieuwe synthetische data, maar verrijkt de datasets door de dimensionaliteit van de dataset te reduceren. Beide datasets beschikken over een relatief groot aantal features (i.e. dimensies). Bij een vaste en vooraf bepaalde hoeveelheid trainingsdata neemt de performantie van ML-modellen over het algemeen af naarmate het aantal features (dimensionaliteit) toeneemt [27]. Dat fenomeen wordt in de literatuur beschreven als 'The curse of dimensionality' [27]. Een te groot aantal features kan daarom een negatief effect hebben op de prestaties van modellen bij classificatieproblemen [37]. PCA wordt dus toegepast om te controleren of de prestaties van de ML-modellen beïnvloed worden door de hoge dimensionaliteit van beide datasets.

In tabel 2 zijn de gebruikte ML-modellen en augmentatietechnieken overzichtelijk weergegeven.

ML-modellen	**Augmentatie methoden**	**Dimensionaliteitsreductie algoritme**
Random Forest (RF)	SMOTE	PCA
XGBoost (XGB)	ADASYN	
Logistic Regression (LogReg)	BorderlineSMOTE	
	SVMSMOTE	
	CTGAN	
	TVAE	

Tabel 2: Overzicht gebruikte modellen/methodes

In figuur 2 is er op een overzichtelijke en schematische manier weergegeven hoe het trainen van elk ML model voor elke augmentatiemethode/PCA is gebeurd. Ook geeft deze weer hoe het samplen van de kleine datasets is gebeurd uit de grote datasets.

3.5 Evaluatie ML-modellen

Om een eerlijke evaluatie van de ML-modellen mogelijk te maken en de resultaten onderling te kunnen vergelijken, werden de XGB-, LogReg- en RF-modellen eveneens getraind op de volledige dataset. Hiervoor werd een train/test-split gemaakt waarbij 70% van de volledige diabetesdataset werd gebruikt voor het trainen van de ML-modellen en de resterende 30% voor het testen. Die modellen, ook wel populatiemodellen genoemd, doorliepen dezelfde processing- en evaluatiestappen als de modellen die op de kleinere datasets werden getraind, met als enige verschil de beschikbaarheid van een grotere hoeveelheid data.

Voor de evaluatie werden vier klassieke ML-metrieken gebruikt: accuracy (1), precision (2), recall (3) en de F1-score (4). Gezien de grote klasse-imbalans in de twee datasets zou het misleidend zijn om uitsluitend naar accuracy te kijken. Voor de diabetes dataset zou een model dat enkel de meerderheidsklasse voorspelt, al een accuracy van 86% kunnen behalen, zonder enige toegevoegde waarde voor het detecteren van diabetesgevallen. Daarom werden ook precision, recall en de F1-score opgenomen als evaluatiemetrieken. Enerzijds geven die metrieken de bekwaamheid om diabetesgevallen te detecteren weer, anderzijds zijn die metrieken duidelijk gedefinieerd en makkelijk te interpreteren. De precision geeft aan welk percentage van de voorspelde diabetesgevallen effectief correct is. Recall toont aan welk percentage van de werkelijke diabetesgevallen ook daadwerkelijk werd herkend door het model. De F1-score is het harmonisch gemiddelde tussen recall en precision. Voor de diabetes dataset kan gesteld worden dat recall de meest relevante metriek is, gezien de aard van de voorspelling: het is belangrijk om zoveel mogelijk personen met diabetes correct te identificeren.

Voor de studentendataset worden dezelfde evaluatiemetrieken gehanteerd. Aangezien ook in die dataset een matige klasse-imbalans aanwezig is, worden recall, precision en de F1-score gebruikt om de prestaties van de ML-modellen bij het voorspellen van studenten die zich vroegtijdig uitschrijven voor een bepaalde cursus te evalueren. In deze context geeft de recall het percentage aan van studenten die daadwerkelijk vroegtijdig de cursus hebben verlaten en correct voorspeld zijn door het model. De precision geeft het percentage weer van de studenten die door het model als vroegtijdig uitgeschreven zijn voorspeld en dat in werkelijkheid ook zijn.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

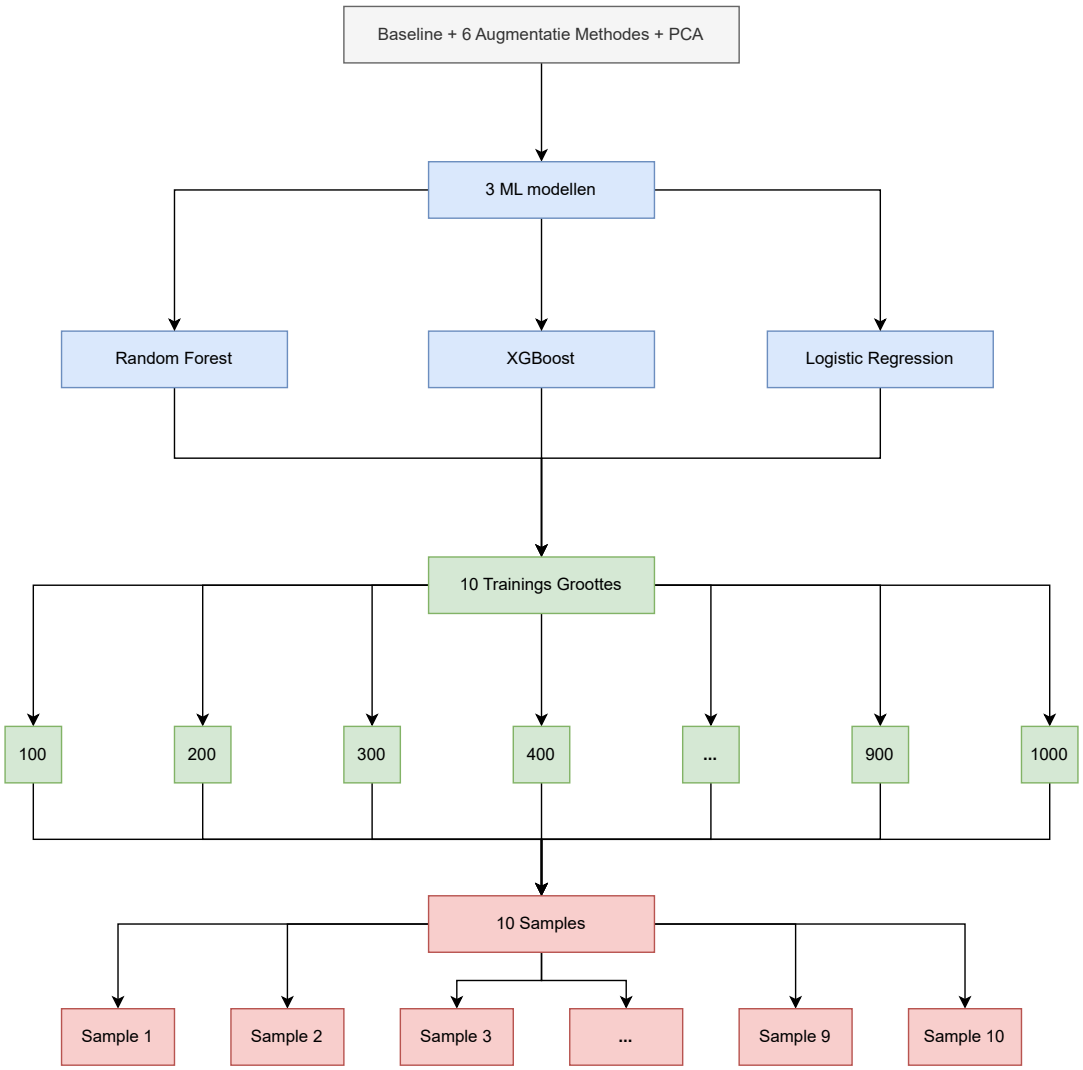
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

3.6 Evaluatie PCA en augmentatietechnieken

Ook de modellen die met geaugmenteerde data of met PCA getraind zijn, moeten met de baseline worden vergeleken om te bepalen of de augmentatietechnieken daadwerkelijk een effect hadden op de datasets. De F1-score vormt de basis voor evaluatie aangezien die metriek de trade-off tussen precision en recall omvat. Voor die vergelijking wordt een Friedman-hypothesetest uitgevoerd, die vaststelt of er significante verschillen bestaan tussen meerdere afhankelijke steekproeven. In deze paper vormen de F1-scores van de modellen die zijn verkregen bij elke trainingsgrootte voor de zes augmentatiemethoden en voor de PCA-training de geobserveerde waarden waaruit de steekproeven bestaan. Voor eenzelfde trainingsgrootte zijn de tien F1-scores onderling onafhankelijk. Tussen elke augmentatiemethode en PCA zijn de scores afhankelijk, omdat ze exact dezelfde rijen gebruiken als de baselinedataset waarop de augmentatie en PCA zijn uitgevoerd. De Friedman-hypothesetest is gekozen wegens zijn non-parametrische karakter aangezien er niet kan aangenomen worden dat de F1-scores per steekproef normaal verdeeld zijn. Verder volgt er een post-hoc Conover-Friedman-test om te identificeren welke methoden onderling en ten opzichte van de baseline verschillen. Indien daaruit significante verschillen tussen augmentatietechnieken en/of PCA naar voren komen, wordt aanvullend de Wilcoxon-rangtest uitgevoerd om na te gaan of de baseline significant betere of slechtere F1-scores behaalt. Er is daarbij rekening gehouden met het feit dat een Wilcoxon-test normaal het verschil tussen twee methodes vergelijkt. Om voor de meerdere methodes te corrigeren is er een Holm-Bonferroni correctie toegepast. Op



Figuur 2: Overzicht training en sampling strategie

die manier kan statistisch vastgesteld worden of een specifieke augmentatietechniek of PCA performanter is dan een onverrijkte dataset in een small-data scenario.

Statistische test	**Reden**
Friedman-hypothesetest	Een non-parametrische test die toetst of er verschillen zijn tussen 3+ afhankelijke samples
Post-hoc Conover-Friedman test	Een post-hoc test die volgt indien Friedman-hypothesetest significant blijkt te zijn. Geeft aan welke methodes verschillen ten opzichte van elkaar
Wilcoxon-Rangtest	Een non-parametrische test die aangeeft in welke richting bepaalde methodes van elkaar verschillen (hogere of lagere performantie)

Tabel 3: Overzicht van de gebruikte statistische tests en de reden waarom ze zijn gebruikt

4 Resultaten

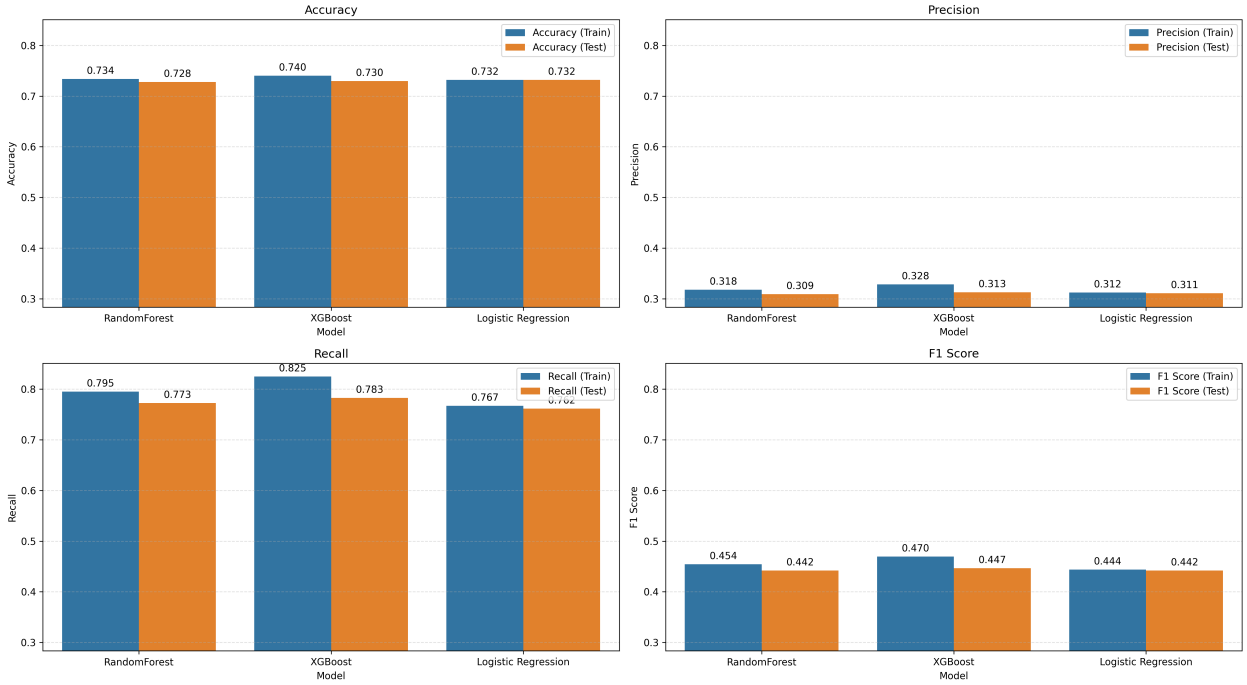
In deze sectie worden de modelprestaties en de statistische analyse van de F1-scores besproken voor alle augmentatietechnieken en PCA in vergelijking met de baseline en de populatiemodellen. Het doel is om te evalueren of de geselecteerde ML-modellen geschikt zijn om betrouwbare prestaties te behalen ondanks het gebruik van kleine datasets. Voor beide datasets is eerst een baseline model opgesteld en daarnaast een populatiemodel getraind op een trainingsset die 70% van de beschikbare data omvatte. De statistische analyse is uitgevoerd op basis van de F1-scores die voor elke steekproef bij elke trainingsgrootte zijn berekend. Concreet gaat het om acht bewerkingen (baseline, zes augmentatietechnieken en PCA), voor elk daarvan zijn drie ML-modellen getraind op tien steekproeven per trainingsgrootte en op tien verschillende trainingsgroottes (zie figuur 2 voor een grafische voorstelling). Dat levert in totaal $8 \times 3 \times 10 \times 10 = 2400$ F1-scores op, die onderling statistisch zijn vergeleken.

4.1 Performantie van ML-modellen

In deze sectie zullen de resultaten van de drie gebruikte ML-modellen (RF, XGB en LogReg) besproken worden. Zoals in sectie 3.5 besproken, zal de evaluatie gebeuren aan de hand van de accuracy, precision en recall van deze modellen. De gehanteerde definities van deze metrieken zijn terug te vinden in vergelijkingen (1) - (3).

4.1.1 Diabetes Dataset

Figuur 3 rapporteert de vier evaluatemetrieken van de drie modellen die zijn getraind op 70% van de volledige dataset, waarbij de overige 30% als testset diende. Ter beoordeling van mogelijke overfitting worden zowel de train- als testscores weergegeven. Deze populatiescores dienen als globaal referentiepunt om na te gaan in hoeverre het small-data scenario vergelijkbare prestaties kan bereiken wanneer een model wel over 'alle' gegevens beschikt.



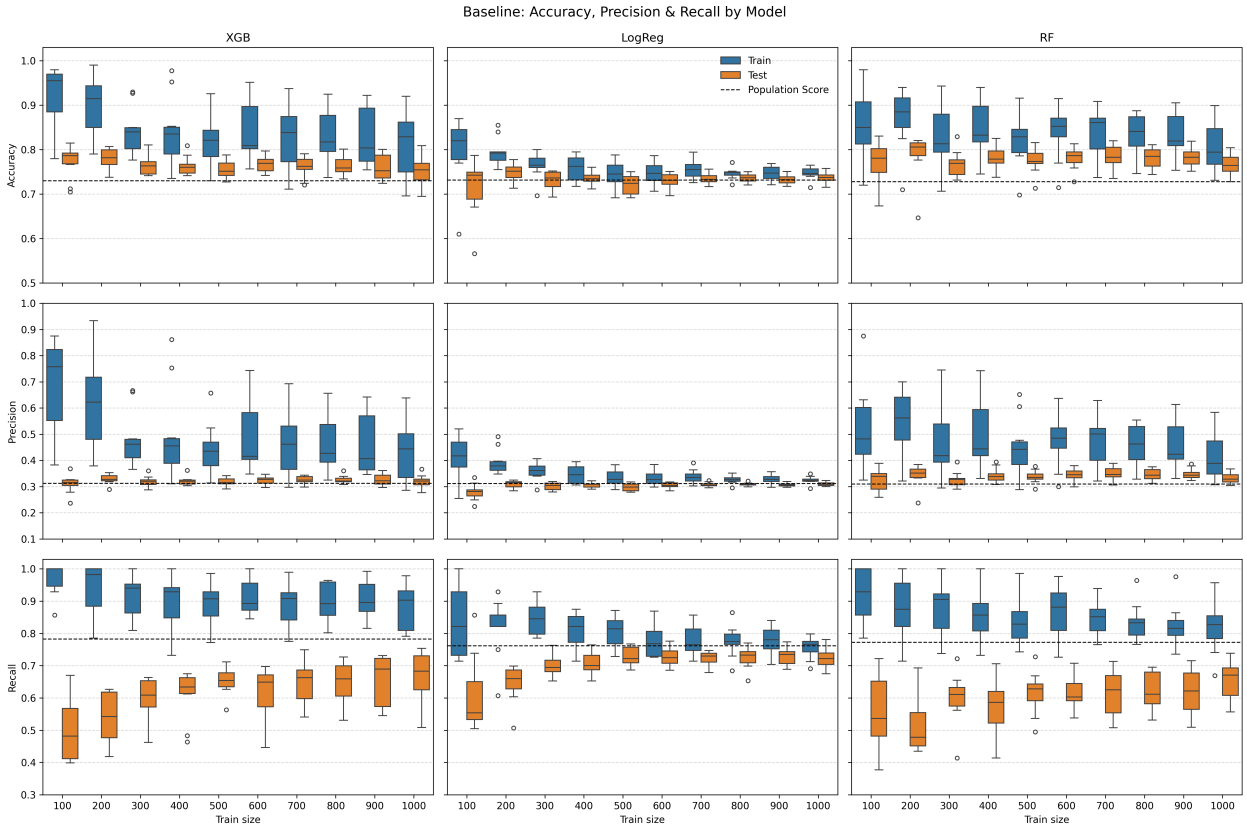
Figuur 3: Evaluatiemetrieken voor de ML modellen getraind op de volledige diabetes dataset

Figuur 3 toont dat de drie ML-modellen gelijkaardige scores leveren op zowel de train- als testdata binnen dezelfde metriek. Dat suggereert dat geen van de modellen in dit geval overfit is op de data. Wat echter opvalt,

zijn de relatief lage precisionscores. Dat betekent dat slechts een klein deel van de als positief voorspelde gevallen effectief diabetes heeft. Alle drie de modellen classificeren een persoon te snel als diabetespatiënt.

In figuren 4–11 worden voor elk ML-model de drie evaluatiemetrieken weergegeven voor de baseline, PCA en de verschillende augmentatietechnieken. De boxplots zijn opgesteld op basis van de gesamplede en verkleinde subsets van de oorspronkelijke (grote) diabetesdataset. Concreet zijn er dus boxplots voor de volgende scenario's:

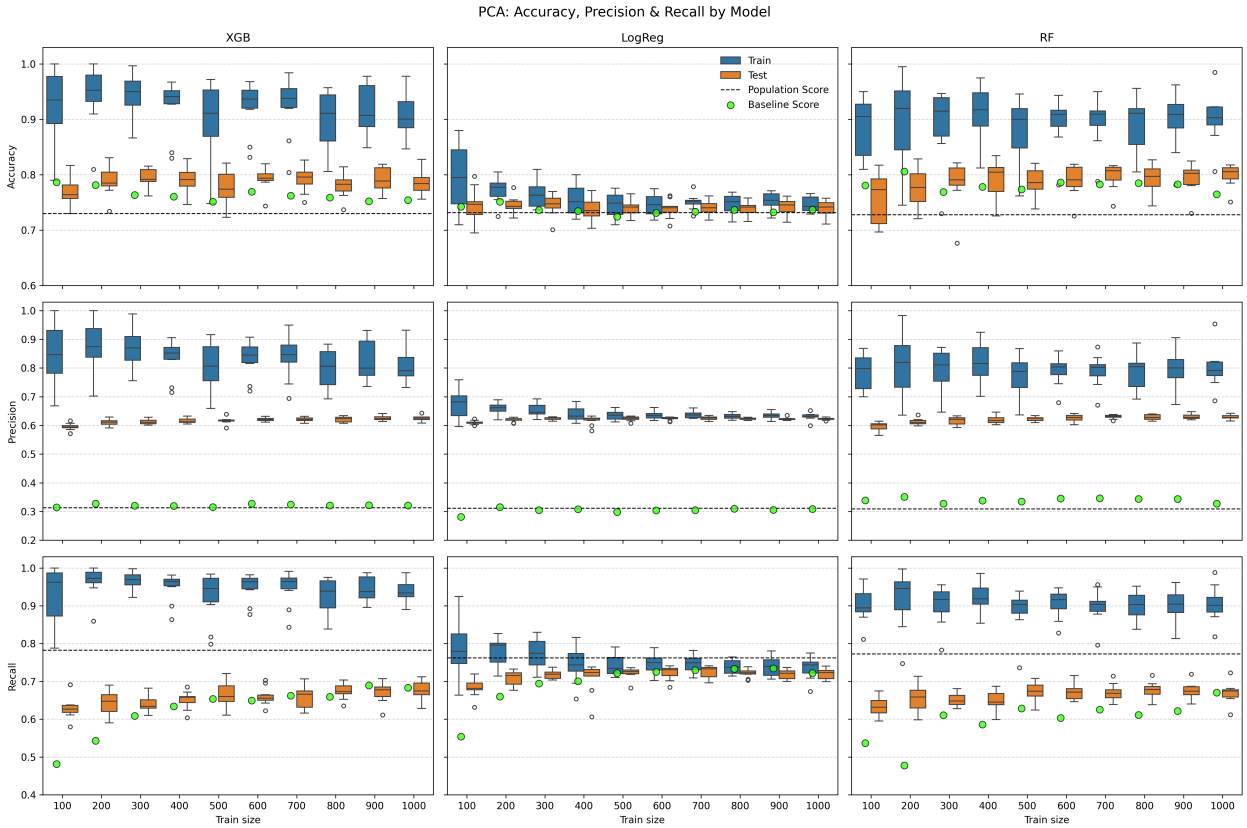
- **Baseline**: resultaten voor het small-data-scenario waarin geen enkele transformatie of augmentatie op de trainingssets is toegepast.
- **PCA**: resultaten nadat op de trainingssets eerst dimensiereductie via Principle Component Analysis is uitgevoerd.
- **Augmentatiemethoden**: prestaties van de drie modellen wanneer de kleine trainingssets zijn verrijkt met de aangegeven methode.



Figuur 4: Boxplots voor de evaluatiemetrieken bekomen met de baseline datasets

Figuur 4 toont de resultaten voor de verkleinde diabetesdatasets. Op de trainingsets van deze datasets is er niet aan dimensiereductie gedaan, noch aan dataaugmentatie. Deze resultaten dienen daarom als baseline voor het small-data scenario. Een duidelijke trend is dat bij de kleinste trainingsgroottes (100–300) de modellen sterk overfitten: de train- en testboxplots liggen dan relatief ver uiteen. Naarmate de trainingsgrootte toeneemt, vermindert deze overfitting, wat naar verwachting is. XGB en RF overfitten de trainingsdata het sterkst. Opvallend is dat zowel de accuracy bij XGB en RF als de precision bij RF gemiddeld iets hoger liggen dan bij de populatiemodellen. De recall daarentegen ligt bij alle drie de modellen lager dan die van hun respectievelijke populatievariant. Hoewel de recall toeneemt met grotere trainingsets, wordt het niveau van

de populatiemodellen niet gehaald. De recallscore van LogReg komt wel in de buurt van het populatiemodel naarmate de trainingsets groter worden. Verder zijn accuracy en precision relatief stabiel, terwijl de recall vooral bij XGB en RF meer variatie vertoont.

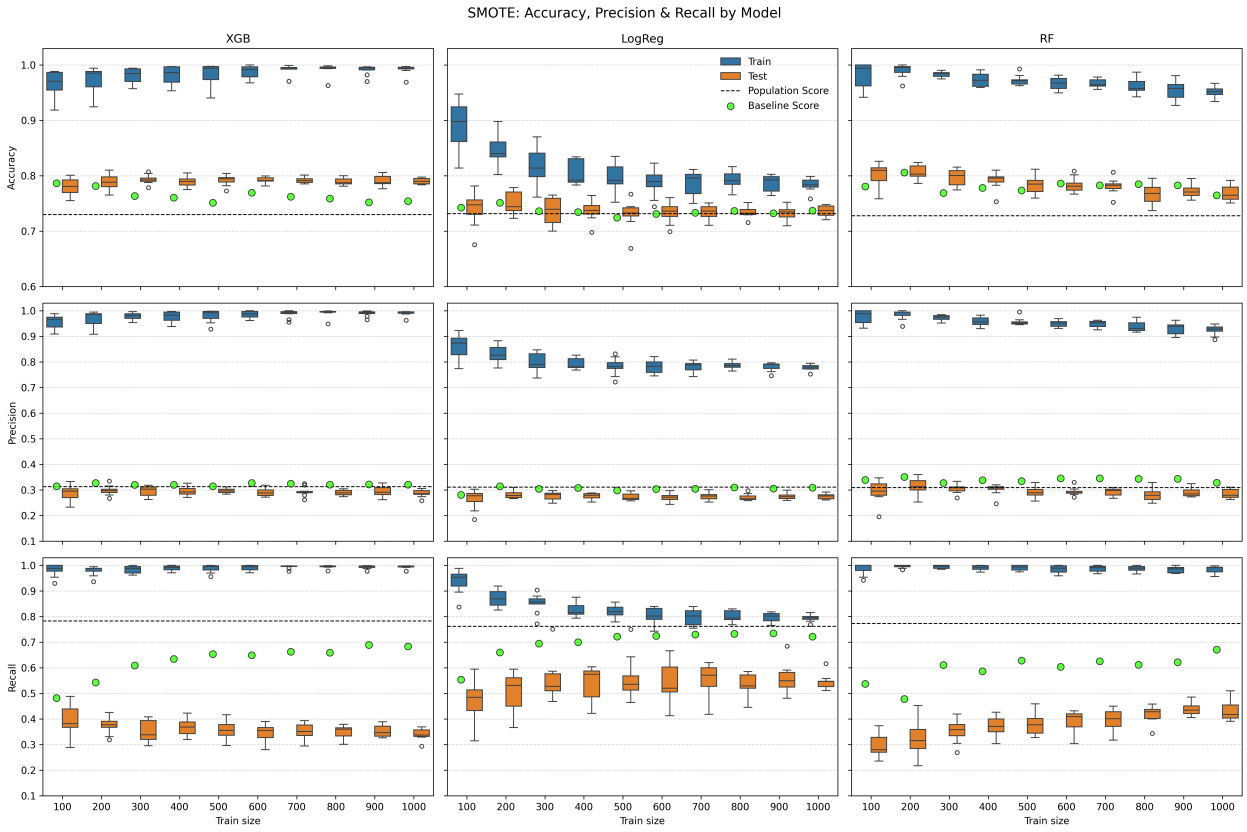


Figuur 5: Boxplots voor de evaluatiemetrieken bekomen met de PCA datasets

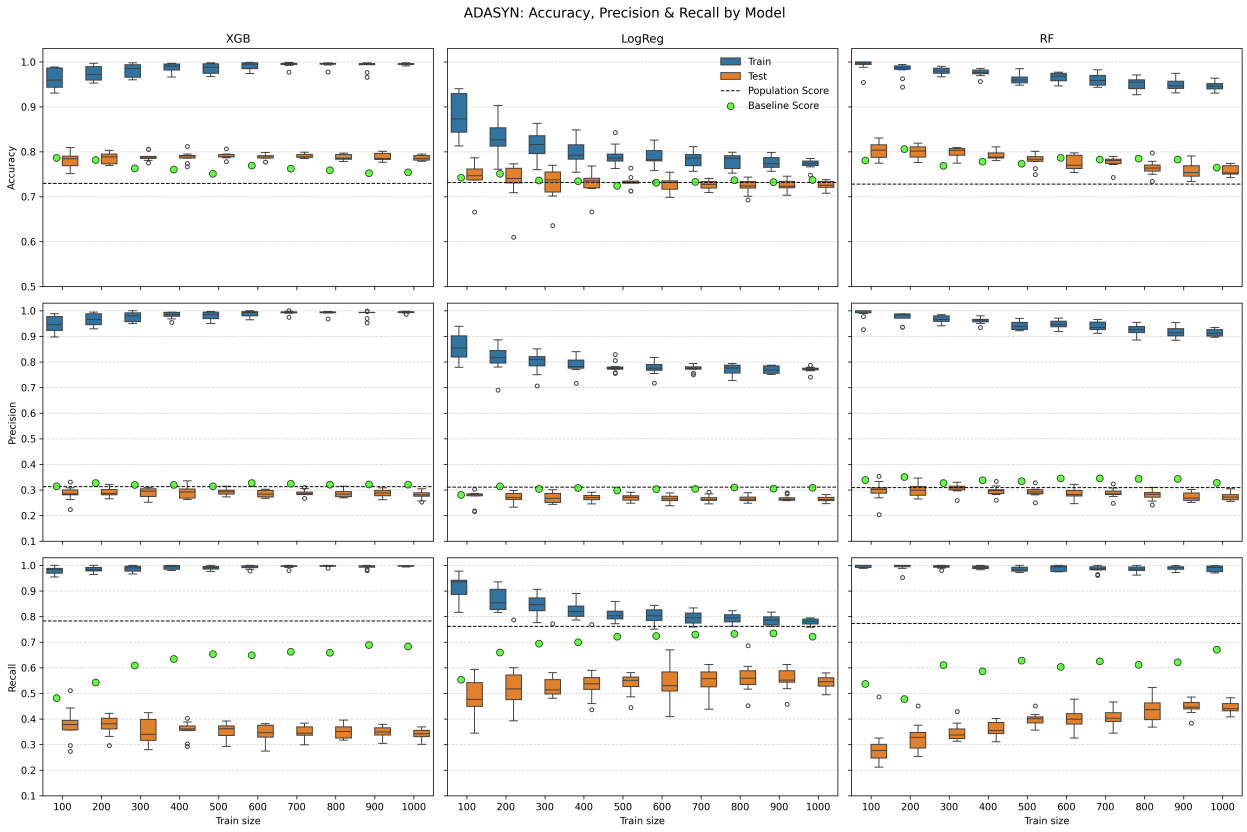
Figuur 5 toont de resultaten van de verkleinde diabetesdatasets, waarbij de trainingsets eerst een dimensiereductie ondergingen via PCA. Wat hierbij opvalt, is dat de precision van alle drie de ML-modellen hoger liggen dan bij de baseline en zelfs de populatiemodellen. De recall blijft grotendeels vergelijkbaar met die van de baseline maar ligt wel lager dan bij de populatiemodellen. Enkel bij het RF ligt de recall gemiddeld hoger ten opzichte van de baseline. Dat betekent dat het aantal vals positieve classificaties daalt, zonder dat dit ten koste gaat van het vermogen om echte positieve gevallen correct te identificeren. Verder ligt de accuracy van zowel XGB als RF gemiddeld hoger dan zowel de baseline als de populatiescores.

Figuur 6 geeft de resultaten weer voor de verkleinde datasets die met het SMOTE-algoritme zijn uitgebreid. Zo ontstaat in de trainingsdata een gelijke klassenverdeling van 50:50. Uit de scores blijkt dat alle drie de modellen sterk overfitten bij elke trainingsgrootte: de train- en testscores liggen relatief ver uit elkaar. Alleen bij de LogReg worden de train- en testscores iets gelijkmatiger naarmate de trainingsgrootte toeneemt. Desondanks behaalt RF en XGB een hogere accuracy dan LogReg. De accuracy ligt bij XGB zelfs gemiddeld hoger dan bij de baseline en het populatiemodel. LogReg presteert echter beter op recall dan de andere modellen. Deze recall komt echter niet in de buurt van de score van de baseline of van de populatie.

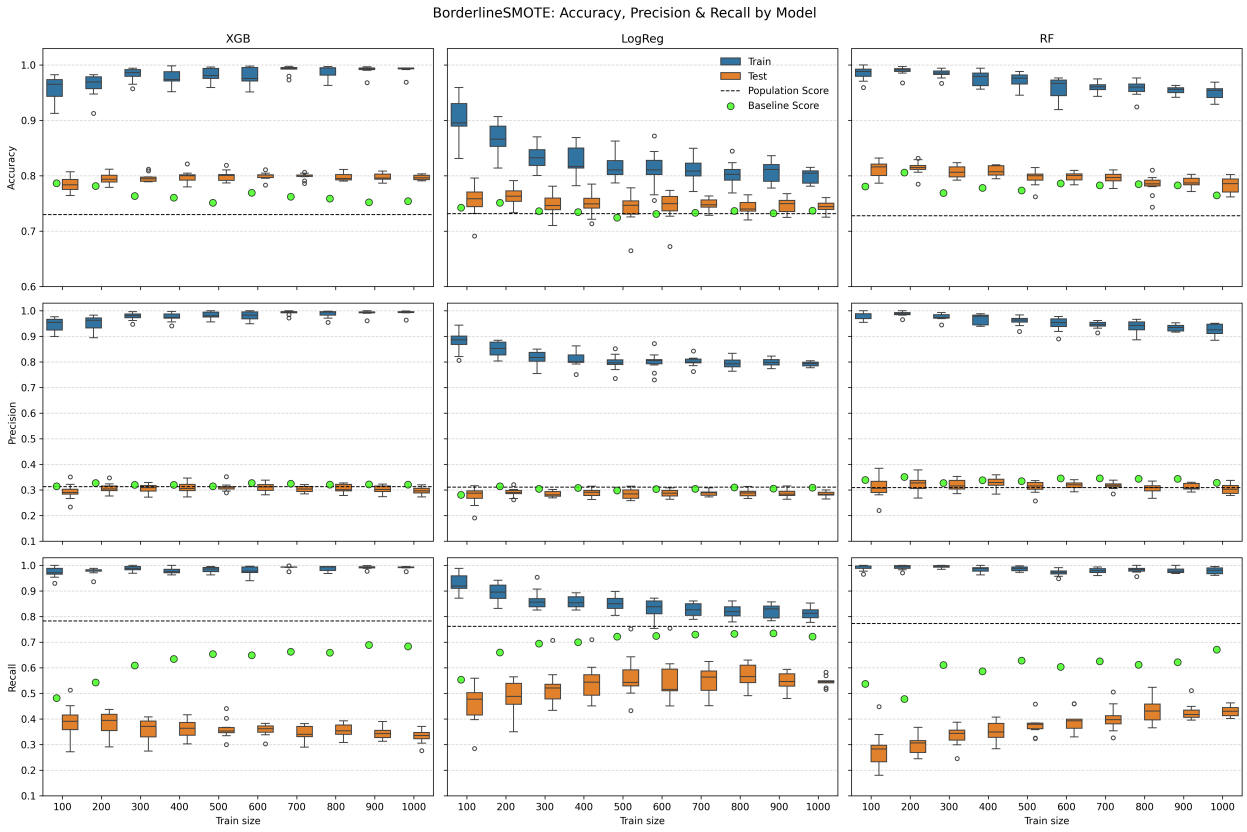
Figuren 7 - 9 kennen een gelijkaardige trend aan figuur 6. Over het algemeen volgen de waardes van deze figuren de waardes die bij de SMOTE-datasets werden geobserveerd. De testaccuracy ligt bij XGB en RF telkens hoger dan de baseline en het respectievelijke populatiemodel. De precision scores zijn telkens vergelijkbaar met die van de baseline en de recall scores zijn altijd lager dan de baseline.



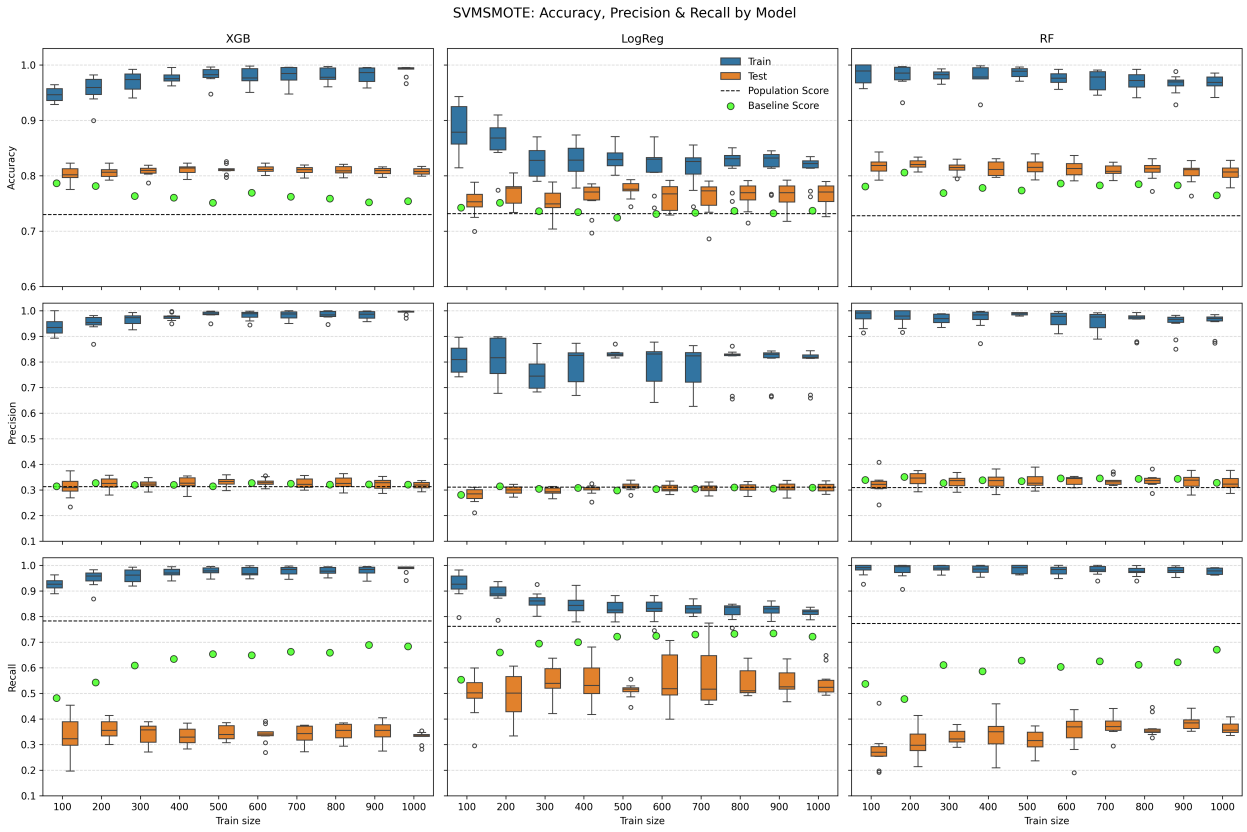
Figuur 6: Boxplots voor de evaluatiemetrieken bekomen met de SMOTE-datasets



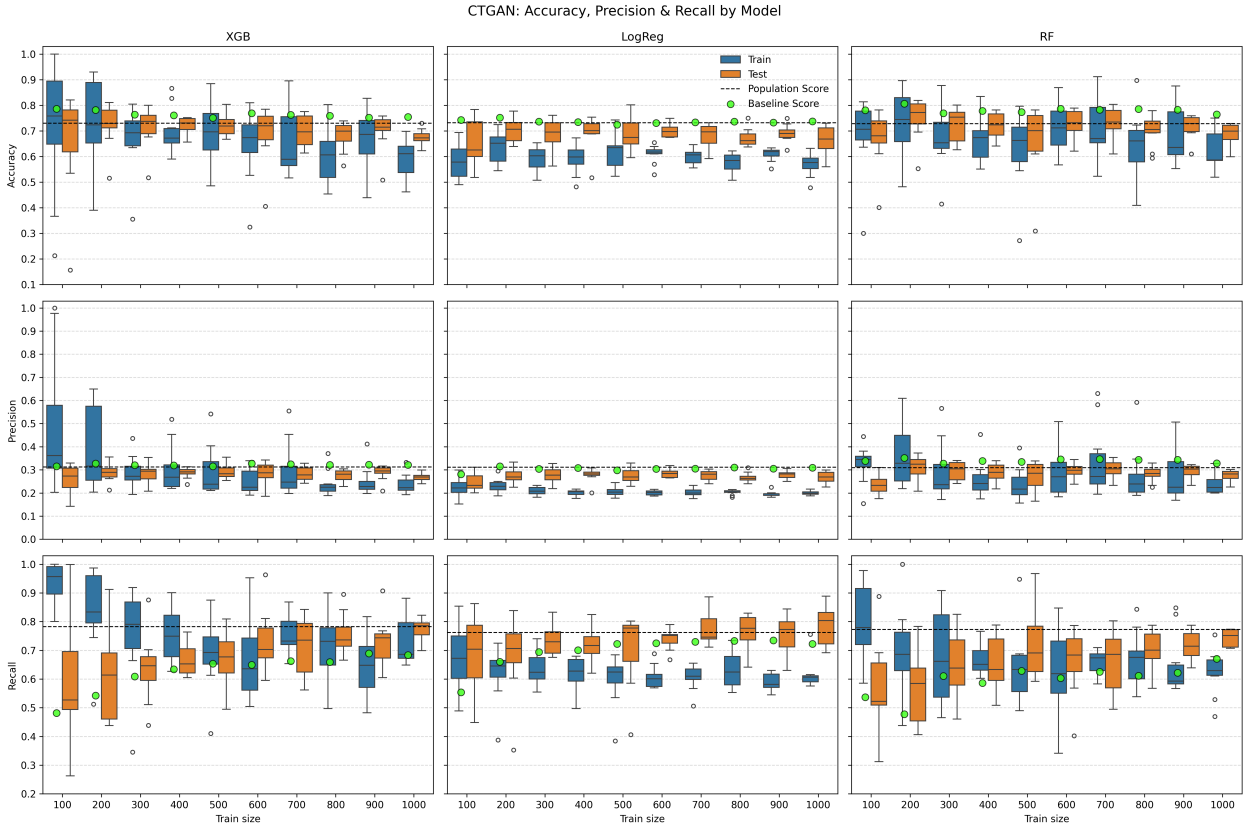
Figuur 7: Boxplots voor de evaluatiemetrieken bekomen op met ADASYN-datasets



Figuur 8: Boxplots voor de evaluatiemetrieken bekomen met de BorderlineSMOTE-datasets

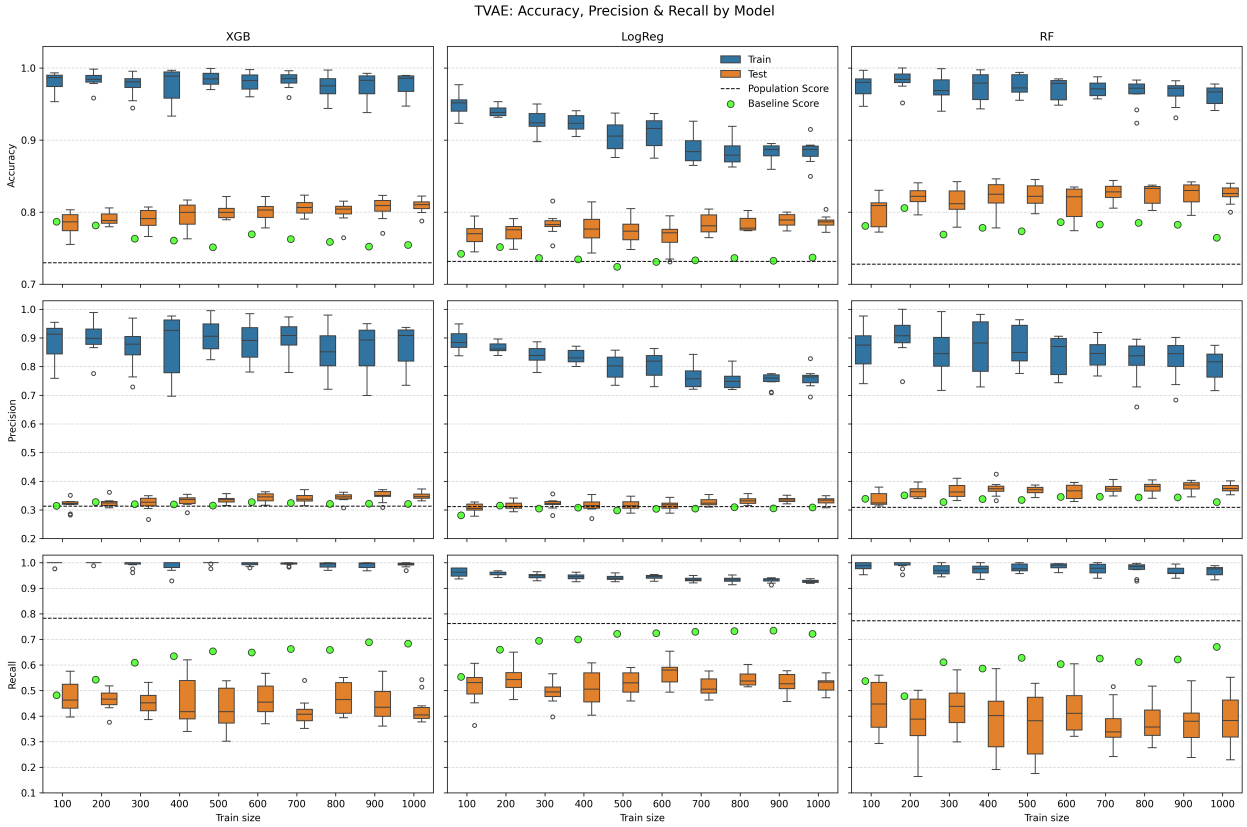


Figuur 9: Boxplots voor de evaluatiemetrieken bekomen op de SVMSMOTE-datasets



Figuur 10: Boxplots voor de evaluatiemetrieken bekomen op de CTGAN-datasets

In figuur 10 staan de resultaten voor de datasets verrijkt met synthetische data van het CTGAN. Opvallend is dat bij het LogReg-model bijna alle testwaarden van de drie metrieken hoger liggen dan de bijbehorende trainingswaarden. Hetzelfde patroon wordt geobserveerd bij de testmetrieken van de RF- en XGB-modellen zodra de trainingsgrootte groter wordt dan 600. Daarnaast zijn de testscores voor recall bij XGB en RF veel variabeler en minder stabiel over de verschillende steekproeven voor elke trainingsgrootte. De testscores voor de accuracy- en precisionmetriek liggen over alle modellen relatief dicht bij de baseline. De mediaan van de testscores voor de recallmetriek ligt daarentegen over het algemeen iets hoger dan die van de recallmetriek bij de baseline.

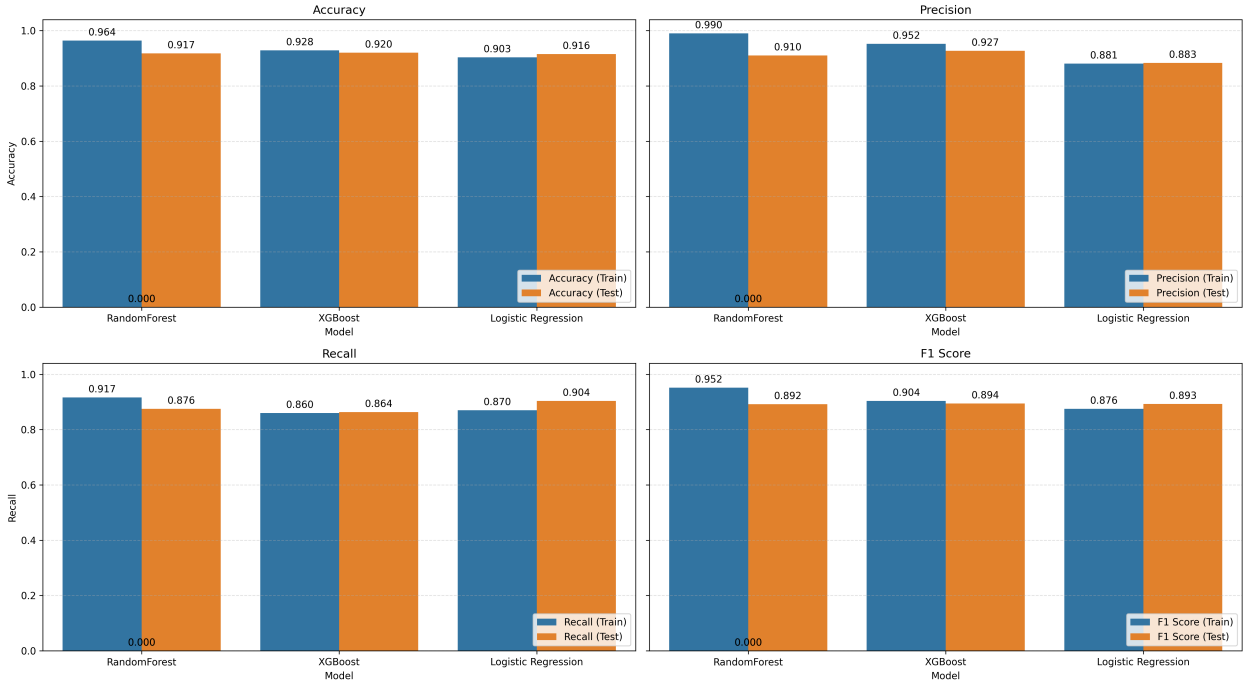


Figuur 11: Boxplots voor de evaluatie metrieken bekomen op de TVAE datasets

In figuur 11 staan de resultaten met de datasets verrijkt met het TVAE-model. Ook hier worden de modellen relatief sterk overfit op de trainingsdata. Opvallend is dat de accuraatheid hoger ligt dan bij de baseline en zelfs de populatiemodellen. Het model is in staat om meer classificaties juist te voorspellen dan wanneer de dataset niet verrijkt zou zijn. Ook de precision scores bij het RF-model liggen marginaal hoger dan de baseline en het populatiemodel. Voor LogReg en XGB liggen deze relatief dicht bij de baseline. De recallscores liggen voor alle modellen lager dan de baseline.

4.1.2 Student Dropout dataset

Deze sectie toont de resultaten voor de Student Dropout dataset. Figuur 12 toont de prestaties van de modellen die zijn getraind op 70% van de volledige dataset. Deze populatiescores dienen als globaal referentiepunt om te beoordelen of het small-data scenario vergelijkbare resultaten kan bereiken wanneer modellen slechts beperkte gegevens zien. Zowel de train- als testscores zijn weergegeven, zodat er nagegaan kan worden of een van de modellen overfit op de trainingsdata.

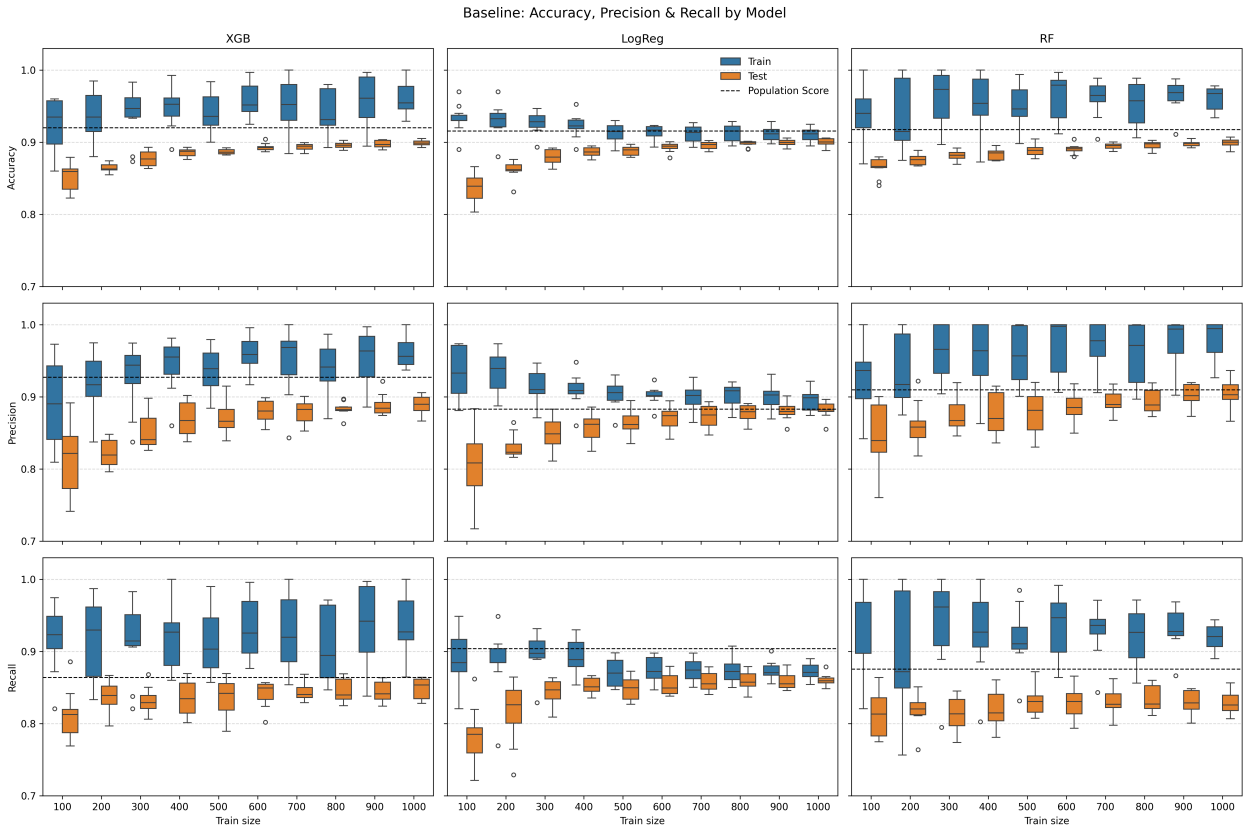


Figuur 12: Evaluatiemetrieken voor de ML-modellen getraind op de volledige studenten dataset

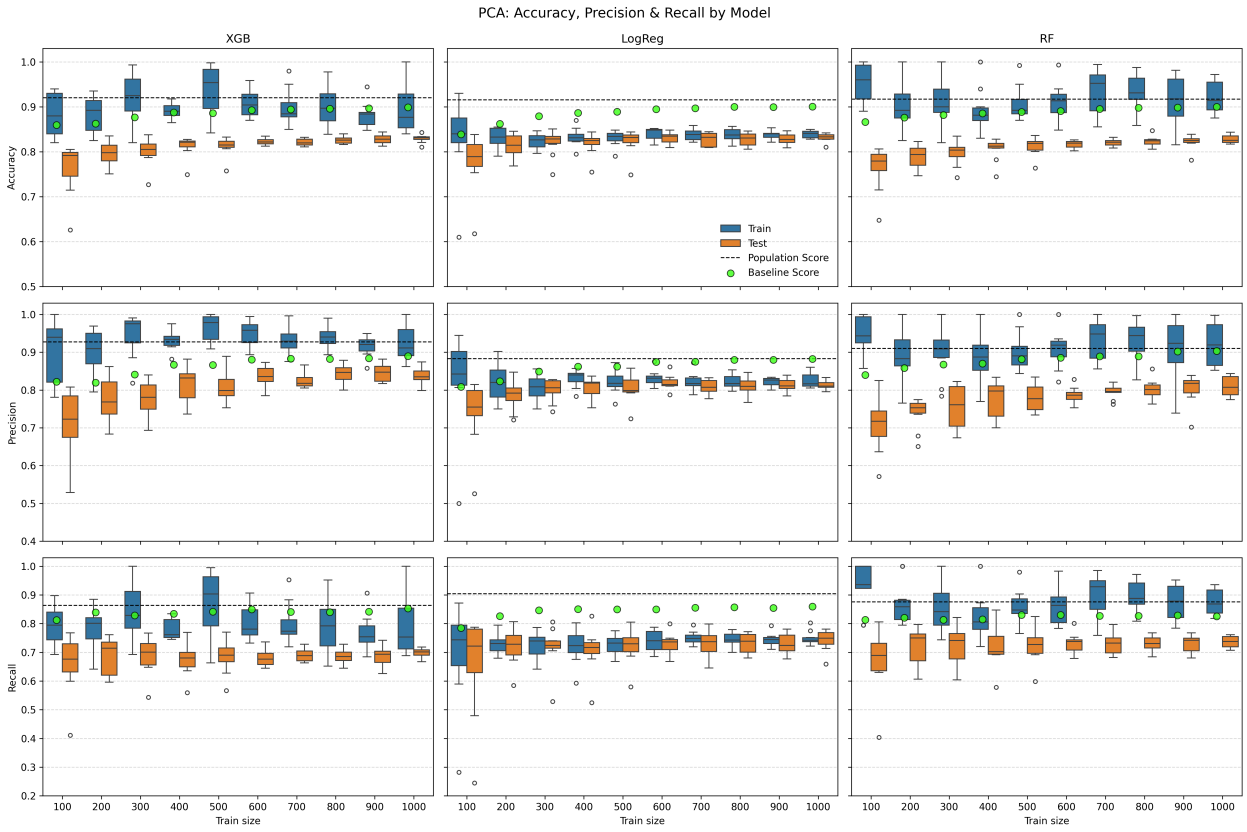
Figuur 12 toont dat bij de populatiemodellen relatief weinig overfitting optreedt aangezien de train- en testscores relatief dicht bij elkaar liggen. De testscores van alle metrieken liggen vrij hoog voor elk model. XGB scoort het hoogst voor de accuracy, precision en F1-score. LogReg doet het beter in termen van recall.

Figuren 13–20 tonen de resultaten van de drie ML-modellen die getraind zijn op de gesamplede, verkleinde subsets van de volledige studentendataset. De uitkomsten worden opnieuw in boxplotvorm weergegeven, aangezien voor elke trainingsgrootte tien steekproeven beschikbaar zijn. De figuren tonen de baselinescores, de scores na toepassing van PCA en de scores van datasets die verrijkt werden met de aangegeven augmentatietechnieken.

Figuur 13 toont de resultaten van de baselinedatasets, waarop geen transformatie of data-augmentatie is toegepast. Uit de figuur blijkt dat alle ML-modellen bij elke trainingsgrootte sterk overfitten ondanks de tijdens het tunen ingestelde regularisatiehyperparameters. De scores van alle metrieken liggen vooral bij trainingsgroottes tussen de 100 en de 500 iets lager dan die van de populatiemodellen. Pas bij grotere trainingsgroottes zijn de testscores meer vergelijkbaar met die van de populatiemodellen. De accuracyscores over de modellen heen zijn vanaf een trainingsgrootte van 300 ongeveer gelijk, ondanks het sterke overfitten bij zowel het XGB- als het RF-model. Bij trainingsgroottes van 100 en 200 presteert het RF-model net iets beter in termen van accuracy dan de andere twee modellen. Ook de recall scores liggen gemiddeld bij alle drie de modellen wat lager dan bij hun equivalente populatievariant. Wat betreft de precision scores ligt deze alleen bij XGB voor alle trainingsgroottes wat lager. Vanaf een trainingsgrootte van 700 presteren LogReg en RF in termen van precision ongeveer even goed als, of beter dan, hun equivalente populatievariant.

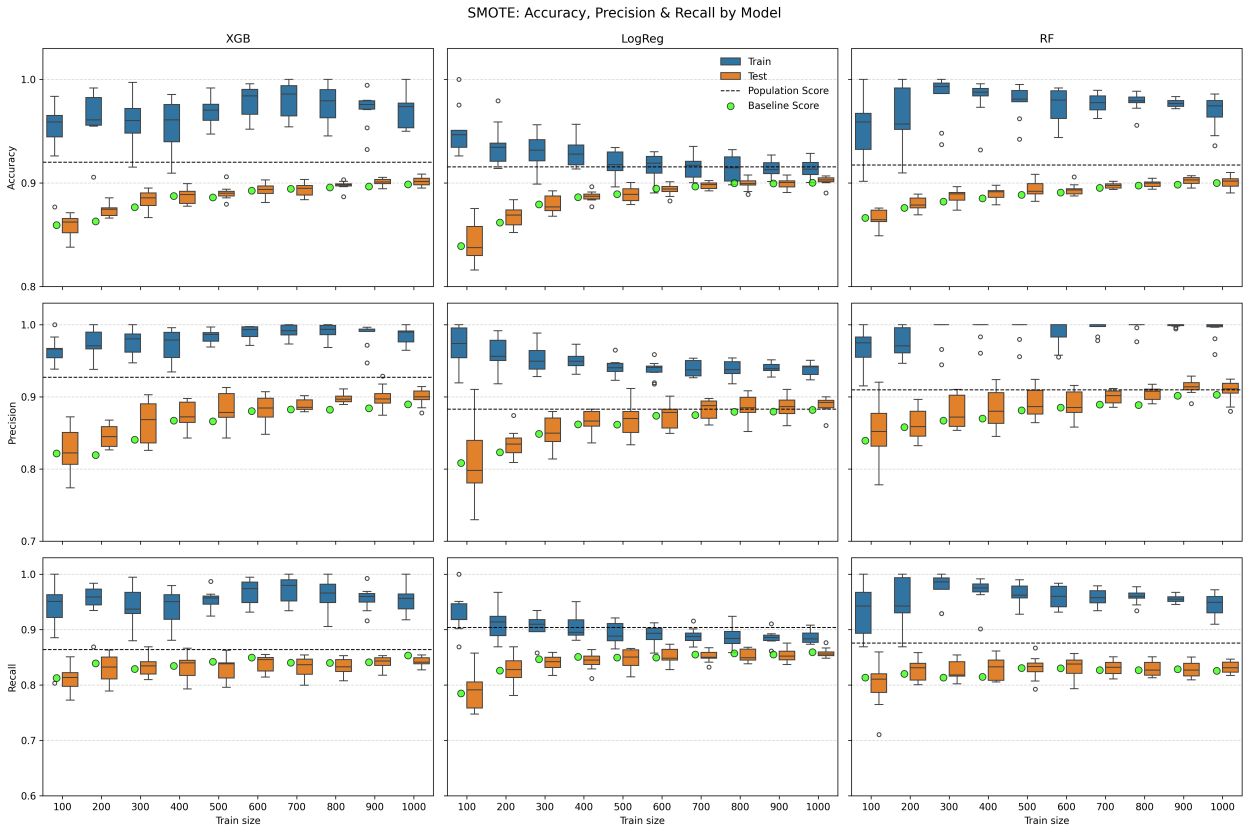


Figuur 13: Boxplots voor de evaluatie metrieken bekomen op de baseline datasets



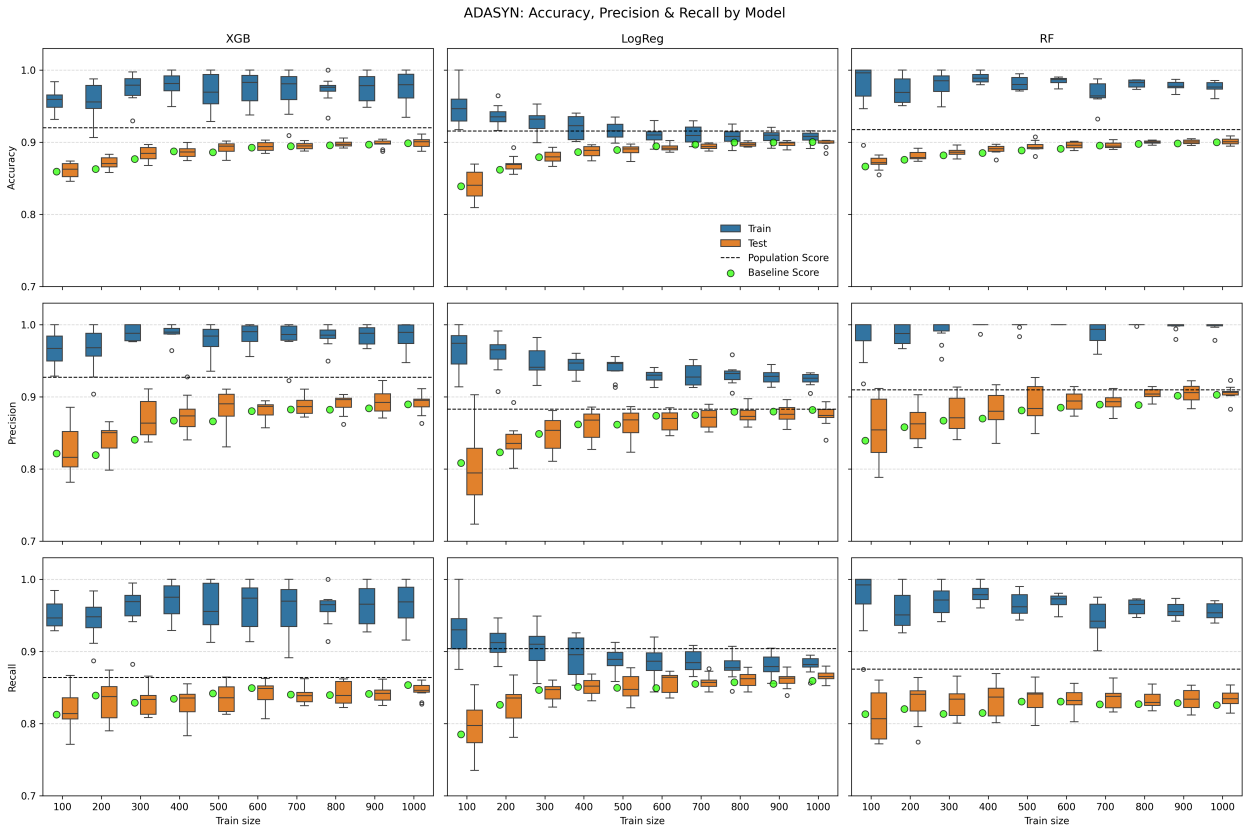
Figuur 14: Boxplots voor de evaluatiemetrieken bekomen met de PCA-datasets

Figuur 14 laat de resultaten zien voor de datasets waarop een PCA-transformatie is toegepast. Opvallend is dat het LogReg-model hier nauwelijks overfit: zelfs bij een trainingsgrootte van 100 liggen train- en testscores dicht bij elkaar. Verder zijn de trainingscores van XGB en RF relatief variabel ten opzichte van LogReg. De trainingscores hebben bij LogReg namelijk een minder brede box dan bij XGB en RF voor alle metrieken. Verder zijn de PCA-testscores over het algemeen bij elk model en elke metriek lager dan de baseline.

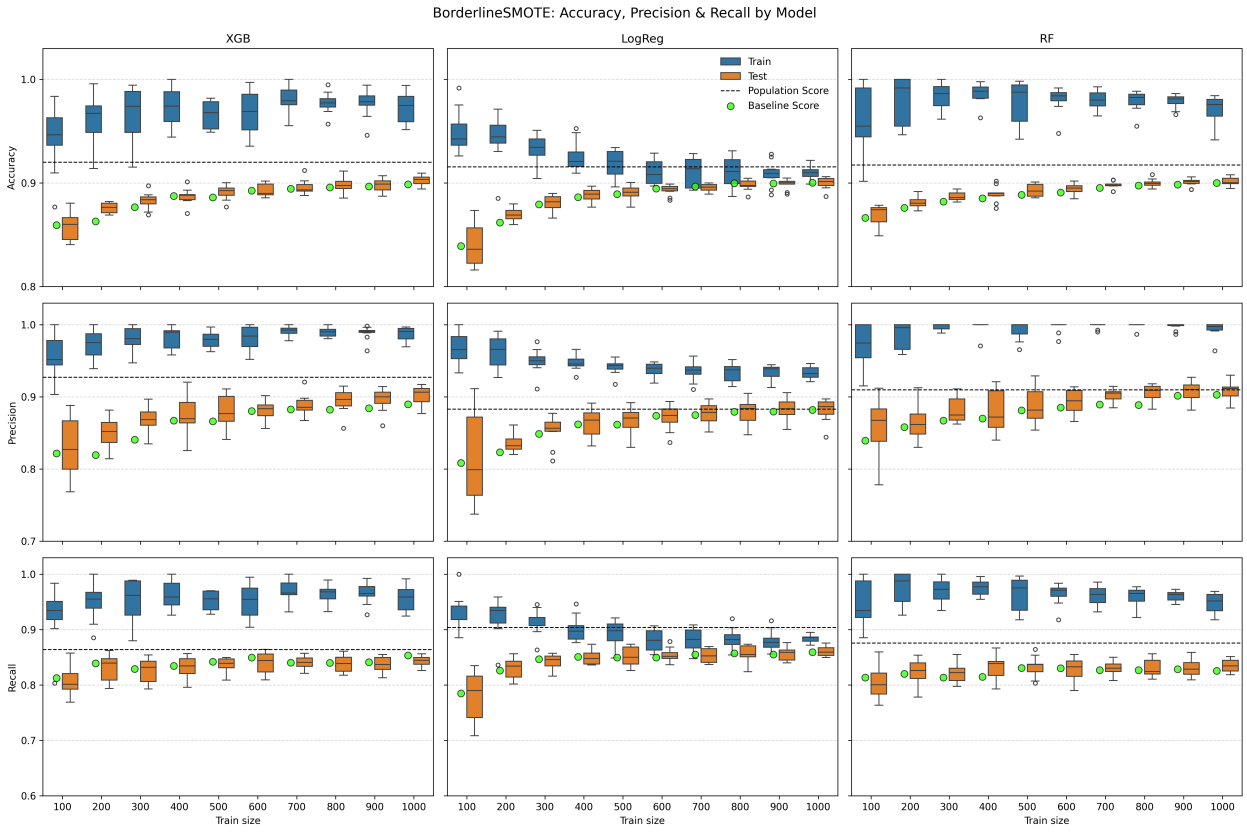


Figuur 15: Boxplots voor de evaluatiemetrieken bekomen op de SMOTE-datasets

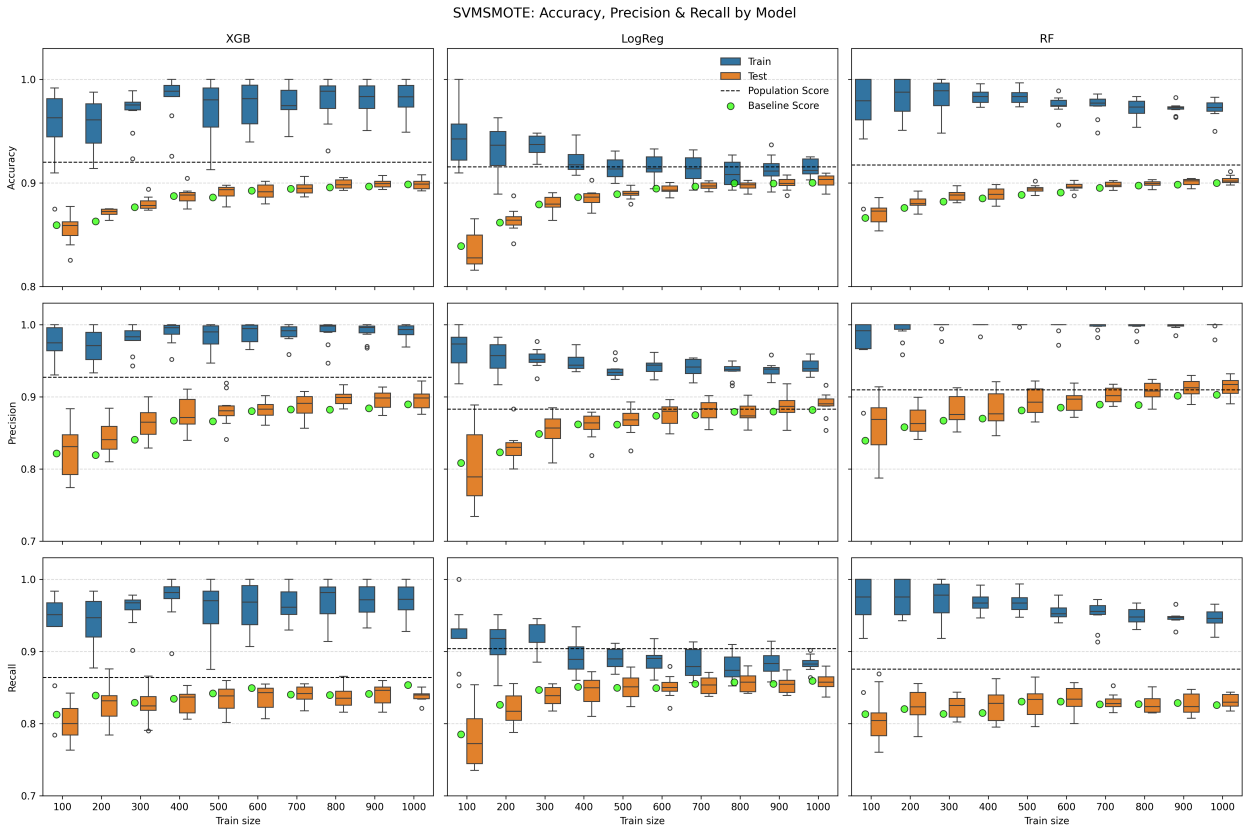
Figuren 15 - 18 zijn de resultaten voor de verkleinde datasets die verrijkt zijn met de respectievelijk aangegeven oversamplingmethode. Over het algemeen kennen al deze figuren een gelijkaardig verloop wat betreft hun resultaten. De ML-modellen zijn sterk overfit op de data. Hoewel dat naarmate de trainingsgrootte groter wordt, de mate van overfitting bij LogReg minder sterk wordt. Bij alle vier de oversamplingmethoden zijn de precision scores bij XGB en RF gelijkaardig of zelfs groter dan de baseline bij elke trainingsgrootte. Voor LogReg zijn de precision scores ongeveer gelijk aan de baseline. Vanaf trainingsgrootte 800 doen de vier oversamplingmethodes het bij LogReg en RF wel marginaal beter dan de baseline en evenaren ze zelfs de populatie precision score. Wat betreft de accuracy en recall behalen de vier oversamplingmethoden vergelijkbare resultaten als de baseline voor elk ML-model.



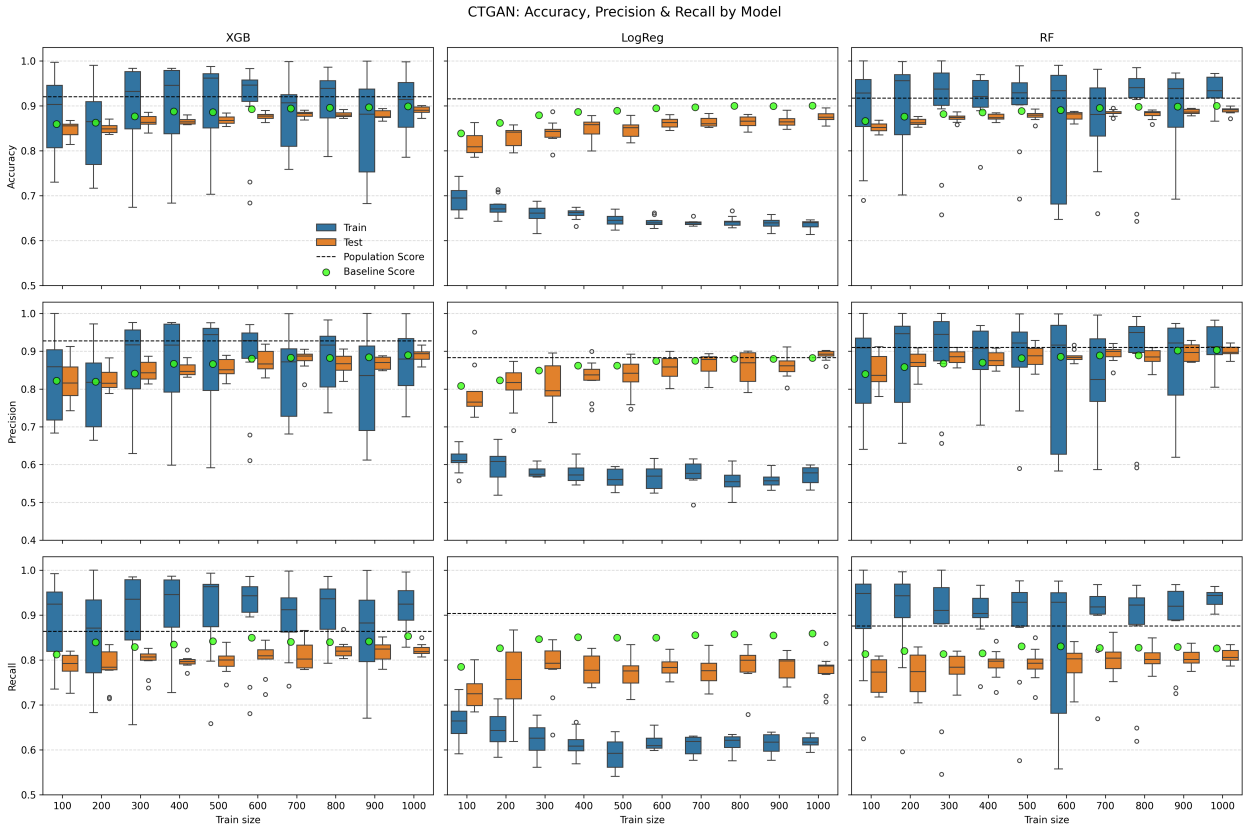
Figuur 16: Boxplots voor de evaluatiemetrieken bekomen op de ADASYN-datasets



Figuur 17: Boxplots voor de evaluatiemetrieken bekomen op de BorderlineSMOTE-datasets

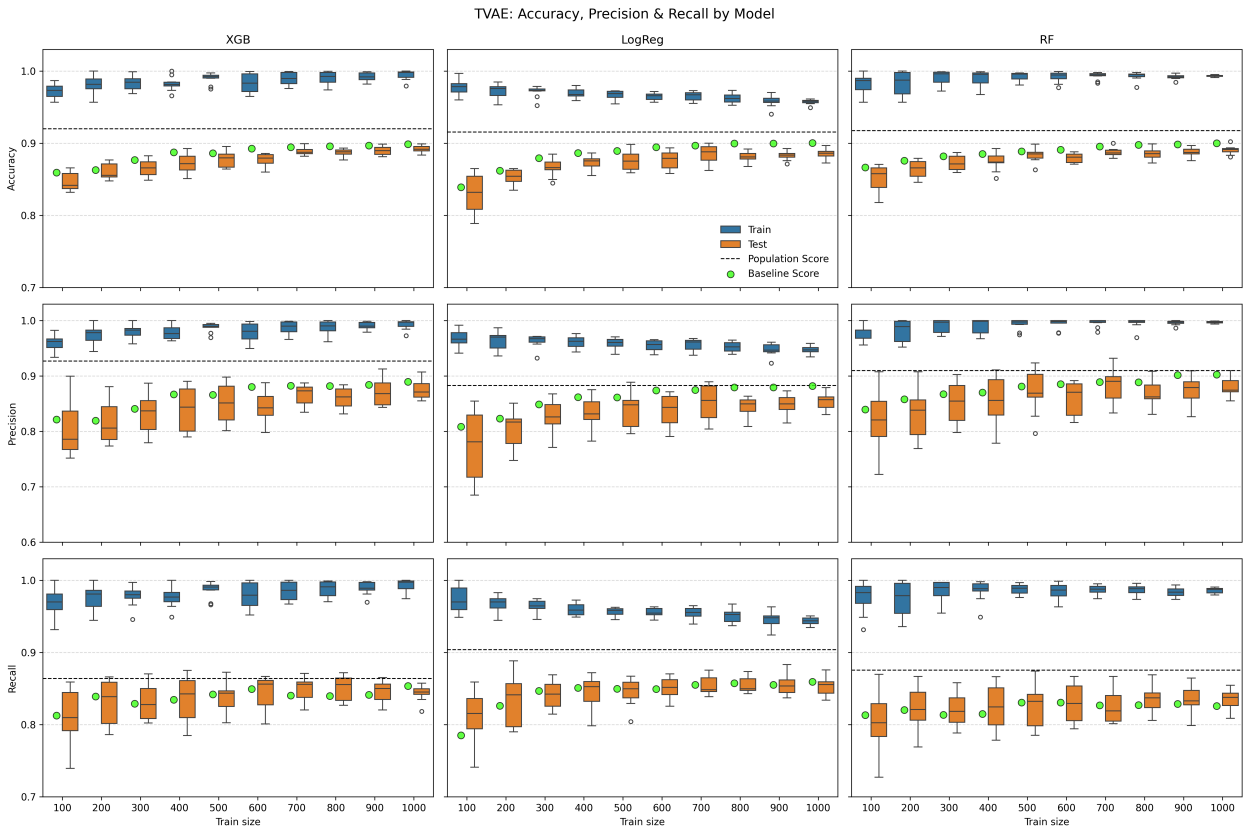


Figuur 18: Boxplots voor de evaluatiemetrieken bekomen op de SVMSMOTE-datasets



Figuur 19: Boxplots voor de evaluatiemetrieken bekomen op de CTGAN-datasets

In figuur 19 zijn de resultaten van de CTGAN-verrijkte datasets weergegeven. In deze figuur is te zien dat de trainingscores van XGB en RF relatief variabel zijn. De trainingsscores van LogReg zijn stabieler. Opvallend is dat de testscores van alle metrieken bij LogReg hoger zijn dan de trainingsscores. De scores voor accuracy en recall blijven gemiddeld altijd lager dan de baseline, terwijl alleen de precision scores van het LogReg-model in de buurt komen van de baseline. De andere twee modellen halen bij de recall score eveneens niet het niveau van hun respectievelijke baseline. Voor accuracy liggen zowel XGB als RF relatief dicht bij de baseline.



Figuur 20: Boxplots voor de evaluatiemetrieken bekomen op de TVAE-datasets

In figuur 20 zijn de resultaten van de datasets verrijkt met TVAE-gesynthetiseerde data weergegeven. De train- en testscores liggen voor elk ML-model vrij ver uiteen, wat wijst op overfitting van de modellen op de trainingsdata. De testscores van de recallmetriek zijn bij alle drie de modellen gemiddeld vergelijkbaar met, en soms zelfs hoger dan, die van de baseline. Het XGB-model behaalt bij trainingsgroottes groter dan 600 gemiddeld vaak resultaten die vergelijkbaar zijn met het respectievelijke populatiemodel in termen van recall. Wat betreft precision en accuracy presteren alle drie de modellen ongeveer gelijk aan tot marginaal slechter dan hun respectievelijke baseline. De precision scores van LogReg en RF liggen vanaf een trainingsgrootte van 600 iets lager, maar nog steeds vergelijkbaar met die van hun respectievelijke populatiemodel. De precision score voor XGB ligt nog relatief ver onder de score van het populatiemodel.

4.2 Statistische analyse van effectiviteit augmentatie technieken

In deze sectie zal voor beide datasets nagegaan worden of er significante verschillen zijn tussen de F1-scores die bekomen zijn door de drie verschillende ML-modellen. Om dat te testen wordt de methodologie gevolgd die in sectie 3.6 beschreven staat. In tabel 4 is een snapshot terug te vinden van hoe de finale dataset eruit zag waarop de statistische analyse is uitgevoerd. Deze tabel is voor beide datasets gemaakt. In figuur 2 is schematisch weergegeven hoe deze tabel tot stand gekomen is.

4.2.1 Diabetes Dataset

Uit tabel 5 kan vastgesteld worden dat de resultaten van de Friedman-hypothesetest voor de drie verschillende ML-modellen significant is. De p-waarden liggen namelijk ver onder de grenswaarde van 0.05. Dat betekent dat er wel degelijk een verschil zit tussen de verdelingen van de F1-scores van de verschillende datasets die de verschillende augmentatiemethodes/PCA vertegenwoordigen.

Model	Trainingsgrootte	Methode	Sample nummer	F1-Score
XGBoost	100	Baseline	1	0.409
XGBoost	100	Baseline	2	0.350
XGBoost	100	Baseline	3	0.363
...	...	...	...	...
Random Forest	1000	TVAE	8	0.342
Random Forest	1000	TVAE	9	0.316
Random Forest	1000	TVAE	10	0.415

Tabel 4: Structuur van de finale tabel waarop de statistische analyse van de F1-scores gebeurd is. Dimensie finale tabel (2400, 5).

Model	Chi-kwadraat	P-waarde
RF	515.36	3.99e-107
XGB	580.43	3.98e-121
LogReg	535.27	2.08e-111

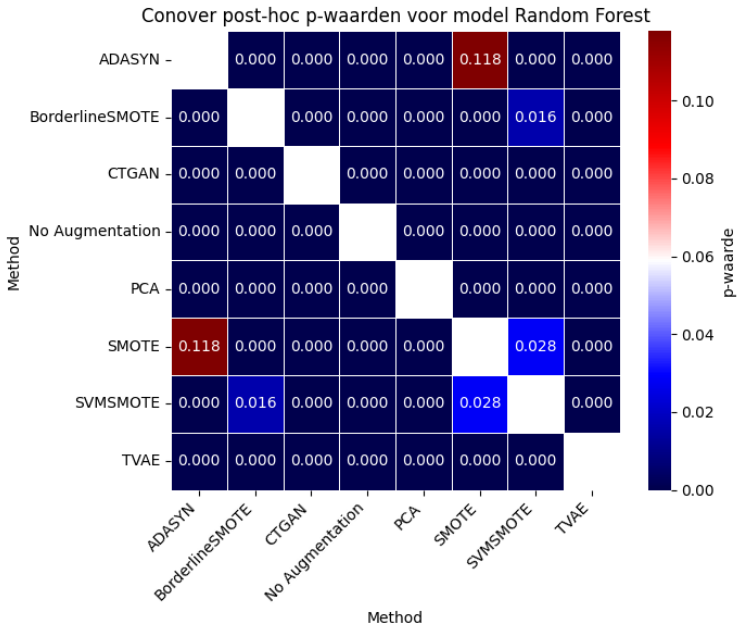
Tabel 5: Friedman-hypothesetest voor de verschillende ML-modellen (CDC diabetesdataset).

Om te kijken tussen welke datasets de verschillen precies zitten is de post-hoc Conovertest uitgevoerd. De resultaten hiervan zijn terug te vinden in figuren 21 - 23. In deze figuren is het duidelijk te zien dat de verdeling van de F1-scores bijna overal ten opzichte van elkaar significant verschillend zijn. Hierbij zijn echter enkele uitzonderingen: BorderlineSMOTE t.o.v. TVAE en SMOTE t.o.v. SVMSMOTE bij het RF model. Verder is er ook geen verschil tussen de verdeling van de F1-scores tussen BorderlineSMOTE-datasets en SVMSMOTE-datasets bij het XGB-model. In deze studie ligt de nadruk vooral op de verschillen tussen de datasets verrijkt met augmentatietechnieken of PCA en de baseline dataset (No Augmentation).
Voor elke augmentatiemethode/PCA die een significant verschil vertoonde (p-waarde kleiner dan 0.05) met de baseline datasets is er ook een Wilcoxon signed-rank test met een Holm-Bonferroni-correctie uitgevoerd om te kijken in welke richting dit verschil ligt. De resultaten van deze testen zijn terug te vinden in tabellen 6 - 8.

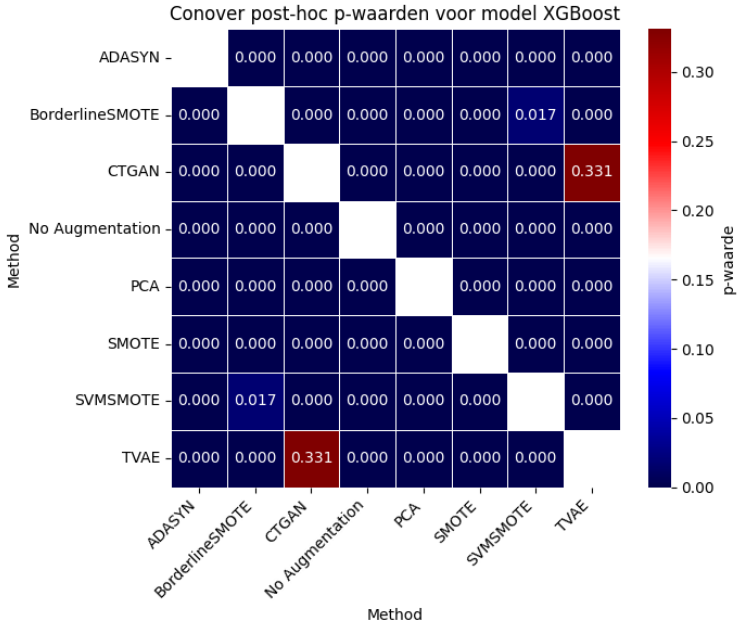
Random Forest - Wilcoxon Test				
Methode	**W-statistiek**	**P-waarde**	**Holm P-waarde**	**Significant?**
ADASYN	0.0	1.0	1.0	Nee
BorderlineSMOTE	0.0	1.0	1.0	Nee
CTGAN	282	1.0	1.0	Nee
PCA	5050	1.95e-18	1.36e-17	Ja
SMOTE	0.0	1.0	1.0	Nee
SVMSMOTE	8.0	1.0	1.0	Nee
TVAE	147	1.0	1.0	Nee

Tabel 6: Wilcoxon-test voor RF-model die aangeeft of de verdeling van de F1-scores hoger ligt dan de baseline (No Augmentation) voor de CDC diabetes Dataset. Volgens de tabel is dat enkel het geval voor de PCA methode.

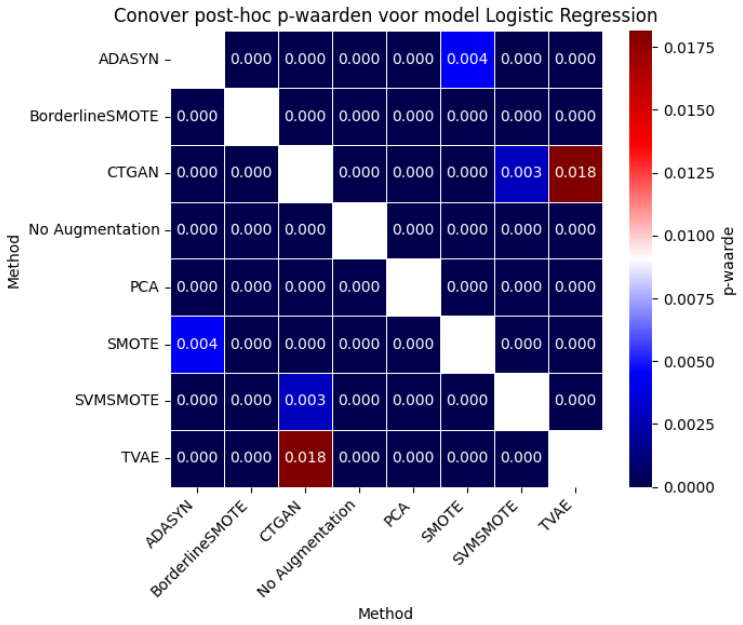
Uit tabellen 6 - 8 blijkt dat de F1-scores van de datasets waar vooraf PCA is op toegepast, statistisch significant hoger zijn dan die van de baseline. Voor elk model, getraind op de PCA datasets, is de p-waarde van de Wilcoxon-test lager dan 5%. De overige data-augmentatietechnieken tonen een p-waarde van 1.0. Dat geeft aan dat deze methoden er niet in geslaagd zijn om de ML-modellen, getraind op de verrijkte data, statistisch significante verbeteringen in F1-scores te laten behalen.



Figuur 21: Post-hoc Conovertest resultaten voor RF op cdc diabetes dataset. P-waarden worden getoond.



Figuur 22: Post-hoc Conovertest resultaten voor XGB op cdc diabetes dataset. P-waarden worden getoond.



Figuur 23: Post-hoc Conovertest resultaten voor LogReg op cdc diabetes dataset. P-waarden worden getoond.

XGBoost - Wilcoxon Test				
Methode	**W-statistiek**	**P-waarde**	**Holm P-waarde**	**Significant?**
ADASYN	0.0	1.0	1.0	Nee
BorderlineSMOTE	2.0	1.0	1.0	Nee
CTGAN	396	1.0	1.0	Nee
PCA	5050	1.95e-18	1.36e-17	Ja
SMOTE	1.0	1.0	1.0	Nee
SVMSMOTE	9.0	1.0	1.0	Nee
TVAE	218	1.0	1.0	Nee

Tabel 7: Wilcoxon-test voor XGB model die aangeeft of de verdeling van de F1-scores hoger ligt dan de baseline (No Augmentation) voor de CDC diabetes Dataset. Volgens de tabel is dat enkel het geval voor de PCA methode.

Logistic Regression - Wilcoxon Test				
Methode	**W-statistiek**	**P-waarde**	**Holm P-waarde**	**Significant?**
ADASYN	10.0	1.0	1.0	Nee
BorderlineSMOTE	3.0	1.0	1.0	Nee
CTGAN	106.0	1.0	1.0	Nee
PCA	5050	1.95e-18	1.36e-17	Ja
SMOTE	3.0	1.0	1.0	Nee
SVMSMOTE	13.0	1.0	1.0	Nee
TVAE	294.0	1.0	1.0	Nee

Tabel 8: Wilcoxon-test voor LogReg die aangeeft of de verdeling van de F1-scores hoger ligt dan de baseline (No Augmentation) voor de CDC diabetes Dataset. Volgens de tabel is dat enkel het geval voor de PCA methode.

CDC diabetes - F1-score (%)			
	Logistic Regression	**Random Forest**	**XGBoost**
Baseline	42.22	42.90	41.90
PCA	**62.74**	**62.96**	**62.58**
SMOTE	36.16	32.93	32.16
ADASYN	35.69	32.72	31.69
BorderlineSMOTE	37.06	33.80	32.84
SVMSMOTE	38.33	33.41	33.21
CTGAN	39.41	39.32	39.20
TVAE	37.79	37.15	38.24

Tabel 9: Deze tabel geeft een globaal gemiddelde van F1-score weer over alle samples van alle training groottes van de verkleinde datasets gesampled uit de CDC diabetesdataset.

Tabel 9 toont de globale gemiddelde F1-scores over alle steekproeven en trainingsgroottes van de diabetes-datasets. In deze tabel is per techniek en model de gemiddelde F1-score weergegeven. Vetgedrukt zijn de F1-scores die hoger zijn dan de baseline. Alleen de PCA-methode scoort beter dan de baseline, met een verbetering van ongeveer 20 procentpunten. De gemiddelde F1-scores van alle overige augmentatiemethoden liggen lager dan die van de baseline.

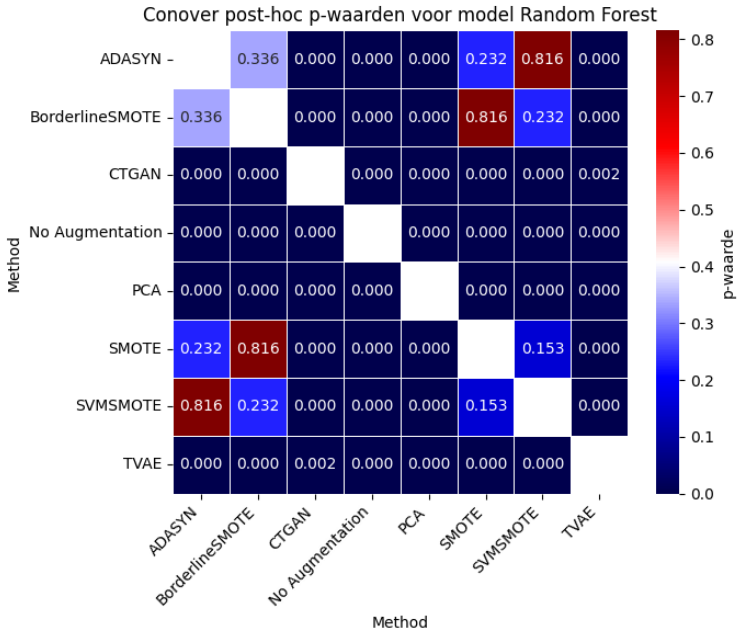
4.2.2 Student Dropout dataset

In tabel 10 zijn de resultaten terug te vinden voor de Friedman-hypothesetest op de drie ML-modellen. Deze tabel toont dat ook hier een verschil zit tussen de verschillende verdelingen van F1-scores per dataset. De p-waarden zijn hier namelijk allemaal beduidend lager dan 5%.

Model	Chi-kwadraat	P-waarde
RF	530.19	2.58e-110
XGB	448.05	1.17e-92
LogReg	452.09	1.58e-93

Tabel 10: Friedman-hypothesetest voor de verschillende ML-modellen (Student Dropout Dataset).

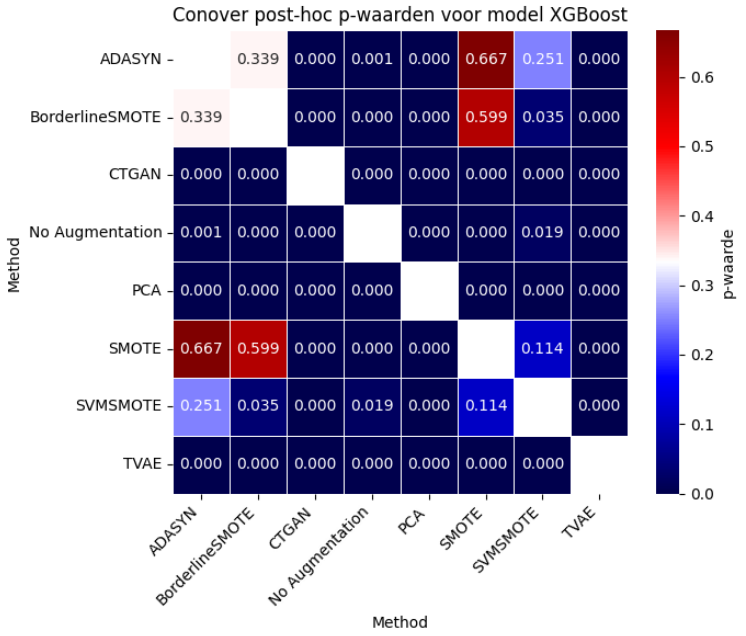
In figuren 24 - 26 zijn de resultaten terug te vinden van de post-hoc Conovertest om te kijken tussen welke methoden de verschillen lagen. De p-waarden in deze figuren geven aan of er al dan niet een significant verschil is in verdeling van de F1-scores van methode 1 t.o.v. methode 2. Indien de p-waarde tussen twee methoden kleiner is dan 0.05, is er een significant verschil tussen de F1-scores van deze technieken.



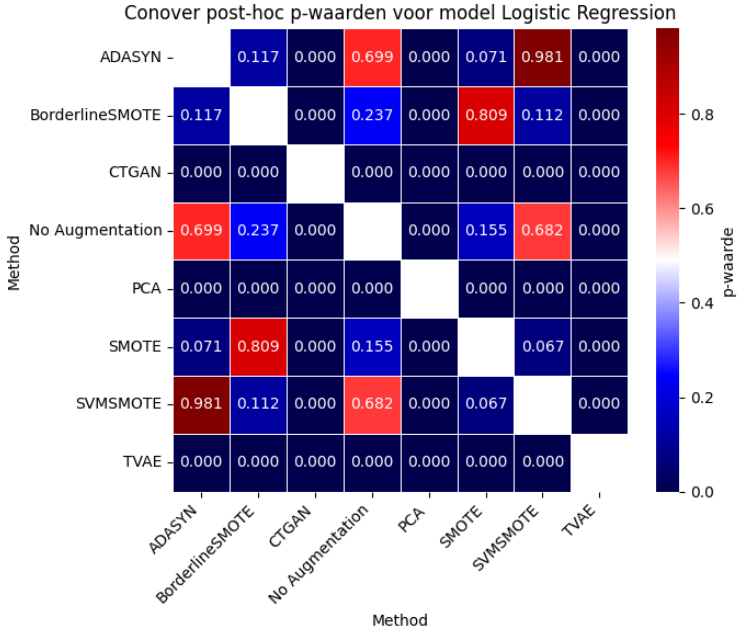
Figuur 24: Post-hoc Conovertest resultaten voor RF op student dropout dataset. P-waarden worden getoond.

Ook hier ligt de nadruk voornamelijk bij de methodes die verschillen ten opzichte van de baseline. Daarom is er per model voor elke methode die significant verschilde ten opzichte van de baseline een Wilcoxon signed-rank test, gecorrigeerd door de Holm-Bonferroni correctie, gebeurd. De resultaten van deze tests zijn terug te vinden in tabellen 11 - 13

Figuur 26 toont aan dat er bij de verkleinde datasets voor LogReg alleen een significant verschil bestaat tussen CTGAN, PCA en TVAE ten opzichte van de baseline. De overige methoden leverden geen significant



Figuur 25: Post-hoc Conovertest resultaten voor XGB op student dropout dataset. P-waarden worden getoond.



Figuur 26: Post-hoc Conovertest resultaten voor LogReg op student dropout dataset. P-waarden worden getoond.

XGBoost - Wilcoxon Test				
Methode	**W-statistiek**	**P-waarde**	**Holm P-waarde**	**Significant?**
ADASYN	3524.0	0.0003	0.0019	Ja
BorderlineSMOTE	3531.0	0.0003	0.0019	Ja
CTGAN	164.0	1.0	1.0	Nee
PCA	0.0	1.0	1.0	Nee
SMOTE	3472.0	0.0006	0.0028	Ja
SVMSMOTE	3194.0	0.0107	0.0429	Ja
TVAE	523.0	1.0	1.0	Nee

Tabel 11: Wilcoxon-test voor XGB die aangeeft of de verdeling van de F1-scores hoger ligt dan de baseline (No Augmentation) voor de student dropout dataset. Dat is volgens de tabel het geval voor de vier oversamplingsmethodes

Logistic Regression - Wilcoxon Test				
Methode	**W-statistiek**	**P-waarde**	**Holm P-waarde**	**Significant?**
CTGAN	2.0	1.0	1.0	Nee
PCA	0.0	1.0	1.0	Nee
TVAE	483.0	1.0	1.0	Nee

Tabel 12: Wilcoxon-test voor LogReg die aangeeft of de verdeling van de F1-scores hoger ligt dan de baseline (No Augmentation) voor de student dropout dataset. Volgens de tabel is dat niet het geval voor de CTGAN-, PCA- en TVAE-methodes

Random Forest - Wilcoxon Test				
Methode	**W-statistiek**	**P-waarde**	**Holm P-waarde**	**Significant?**
ADASYN	4525.0	<0.0001	<0.0001	Ja
BorderlineSMOTE	4383.0	<0.0001	<0.0001	Ja
CTGAN	83.0	1.0	1.0	Nee
PCA	0.0	1.0	1.0	Nee
SMOTE	4356.0	<0.0001	<0.0001	Ja
SVMSMOTE	4403.0	<0.0001	<0.0001	Ja
TVAE	219.0	1.0	1.0	Nee

Tabel 13: Wilcoxon-test voor RF die aangeeft of de verdeling van de F1-scores hoger ligt dan de baseline (No Augmentation) voor de student dropout dataset. Dat is volgens de tabel het geval voor de vier oversamplingsmethodes

verschillende distributies van F1-scores op. Daarom is bij de LogReg-modellen enkel een Wilcoxon-test uitgevoerd tussen de baseline en de augmentatiemethoden die significante verschillen in F1-scores vertoonden. Uit tabel 12 kan worden geconcludeerd dat voor het LogReg-model geen enkele augmentatie- of transformatietechniek statistisch significante verhogingen in de distributies van F1-scores weet te realiseren. De gecorrigeerde p-waarden liggen voor alle drie de methoden ruim boven de 5%. Tabel 11 laat zien dat voor het XGB-model de oversamplingmethoden ADASYN, BorderlineSMOTE, SMOTE en SVMSMOTE wel in staat zijn om hogere distributies van F1-scores te bereiken ten opzichte van de baseline. De gecorrigeerde p-waarden van de Wilcoxon-test liggen telkens onder de 5%. Daarnaast liggen ook hier de gecorrigeerde p-waarden voor de CTGAN-, TVAE- en PCA-methode ver boven de 5%. Eveneens toont tabel 13 aan dat er voor het RF-model een statistisch significante verhoging in verdeling van F1-scores bestaat tussen de datasets verrijkt met SMOTE, ADASYN, BorderlineSMOTE en SVMSMOTE, en de baseline datasets. Ook voor deze vier oversamplingmethodes liggen de gecorrigeerde p-waarden van de Wilcoxon-test ver onder de 5%. Voor de overige drie methoden liggen de gecorrigeerde p-waarden ver boven de 5%.

Student Dropout Dataset - F1-score (%)			
	Logistic Regression	**Random Forest**	**XGBoost**
Baseline	85.05	85.16	85.04
PCA	75.53	74.56	73.88
SMOTE	**85.24**	**85.61**	**85.36**
ADASYN	**85.17**	**85.73**	**85.37**
BorderlineSMOTE	**85.24**	**85.65**	**85.40**
SVMSMOTE	**85.08**	**85.70**	**85.23**
CTGAN	80.29	83.30	82.93
TVAE	83.70	84.13	84.08

Tabel 14: Deze tabel geeft een globaal gemiddelde van F1-score weer over alle samples van alle training groottes van de verkleinde datasets gesampled uit de student dataset.

Tabel 14 geeft de globale gemiddeldes weer van de F1-scores over alle steekproeven van alle trainingsgroottes van de studentendatasets weer. In de tabel is duidelijk te zien wat de gemiddelde F1-score per techniek en model is over alle steekproeven heen. In het vet staan alle F1-scores die groter zijn dan de baseline. Uit de resultaten blijkt tevens dat het globale gemiddelde van de vier oversamplingmethoden gering hoger ligt dan dat van de baseline. Daarentegen presterevn CTGAN en TVAE bij elk model enkele procentpunten slechter dan de baseline en PCA scoort zelfs gemiddeld 10 procentpunten lager dan de baseline.

5 Bespreking

In bovenstaande experimenten is onderzocht of ML-modellen getraind op kleine datasets een vergelijkbare performantie kunnen behalen als modellen op grote datasets. Daarnaast is er onderzocht of state-of-the-art augmentatietechnieken waardevol zijn om kleine datasets te verrijken en zo de prestaties van de ML-modellen te verbeteren. In de onderstaande secties worden de verkregen resultaten van het experiment besproken. Sectie 5.1 vergelijkt de prestaties van verschillende ML-modellen, getraind op de verkleinde datasets die als baseline dienen voor het small-data scenario, met die van hun respectievelijke populatiemodellen. Sectie 5.2 bespreekt het effect van PCA op de performantie van de gebruikte ML-modellen ten opzichte van de baseline. Vervolgens behandelt sectie 5.3 de resultaten van datasets waarop oversamplingtechnieken toegepast zijn en vergelijkt deze met de baseline. In sectie 5.4 worden de resultaten besproken van datasets die zijn verrijkt met de zuivere augmentatiemethoden CTGAN en TVAE, wederom in vergelijking met de baseline. Tot slot wordt in sectie 5.5 een antwoord gegeven op de geformuleerde onderzoeksvragen.

5.1 Populatiemodellen & baseline

De resultaten van de studentendataset komen overeen met de verwachting dat ML-modellen minder goed presteren wanneer ze op kleinere datasets zijn getraind en daardoor slechts aan een beperkt deel van de

datadistributie zijn blootgesteld (figuur 13). Bij de diabetesdataset zijn de resultaten voor XGB en RF echter in tegenspraak met deze verwachting, gegeven dat zij voor bepaalde metrieken beter presteren (figuur 4). Bij RF liggen zowel accuracy als precision hoger dan bij de baseline. Bij XGB is enkel de accuracy hoger, terwijl de andere twee metrieken vergelijkbaar of lager scoren dan bij het populatiemodel. Dat kan betekenen dat deze modellen conservatiever zijn geworden bij training op kleine datasets. Concreet houdt dit in dat de modellen waarschijnlijk vaker instanties classificeren als niet-diabetespatiënt, en enkel bij hoge zekerheid een positief label (diabetes) toekennen. Dat wordt ondersteund door de lagere recall scores. Aangezien de testset meer niet-diabetespatiënten dan diabetespatiënten bevat, kan dat verklaren waarom de accuracy verhoogd is. Een conservatief model leidt tevens tot minder vals positieve voorspellingen, waardoor ook de precision hoger is.

Verder blijkt op basis van de train- en testaccuracies van de baseline voor zowel de diabetes- als studentendataset dat het LogReg-model minder sterk overfit naarmate de trainingsgrootte toeneemt. XGB en RF overfitten bij alle trainingsgroottes relatief sterk op de trainingsdata. Dat kan tevens ook een reden zijn waarom de ML-modellen bij de diabetesdataset veel conservatiever zijn geworden in het maken van hun voorspellingen. Hoewel er voor XGB en RF hyperparameteroptimalisatie heeft plaatsgevonden, kan dat erop wijzen dat deze modellen nog te veel representatiecapaciteit hebben voor deze grootorde aan datasets, ondanks het toevoegen van diverse regularisatiehyperparameters tijdens het tuningproces. Gegeven dat LogReg minder overfitting vertoont, suggereert dat de representatiecapaciteit van dit model het meest geschikt is voor deze specifieke taak met deze grootorde aan beschikbare trainingdata. Het verschil in de mate van overfitting kan ook samenhangen met het type model. Logistische regressie is namelijk een parametrisch ML-model wat zorgt voor een inherent regularisatie-effect, terwijl bij non-parametrische modellen alle ruis een direct effect heeft op het leergedrag van het ML-model.

5.2 PCA

Figuur 5 toont dat na dimensiereductie (PCA) de accuracy scores voor XGB en RF relatief hoog zijn, en voor LogReg iets lager, maar vergelijkbaar met de baseline. Wat verder zeer opvallend is, is de mate waarin de precision scores zijn toegenomen bij het gebruik van PCA op de diabetesdataset. De precision scores zijn toegenomen van schommelend rond de 30% naar schommelend rond de 60%. Dat is een stijging van ongeveer 30 procentpunten. Dat is tevens ook een toename ten opzichte van de baseline en de populatiemodellen. Dat suggereert dat de oorspronkelijke, hoog-dimensionele data ertoe leidt dat de ML-modellen sneller positieve (diabetes) voorspellingen maken. Dimensiereductie leidt er dus toe dat er minder vals positieve voorspellingen gemaakt worden. Ook is de recall voor de kleinere trainingsgroottes (100-300) bij de PCA-methode verbeterd en voor de grotere trainingsgroottes hetzelfde gebleven. Dat ondersteunt de hypothese dat de grote hoeveelheid features in deze dataset zorgde voor ruis, met als gevolg veel meer vals positieve voorspellingen. Door de dimensies te reduceren konden de ML-modellen zelfs met relatief weinig data en een grote klasse-imbalans een grotere precision en accuracy halen zonder te moeten inboeten op recall. Die bevindingen worden ook ondersteund door de statistische analyse van de F1-scores. Voor de verkleinde diabetesdatasets blijkt dat bij alle drie de ML-modellen alleen PCA een effectieve methode is om de distributie van F1-scores hoger te krijgen dan die van de baseline. Tabel 9 toont dat het gaat om een grote gemiddelde stijging van ongeveer 20 procentpunten.

Figuur 14 toont echter dat het reduceren van de dimensies bij de verkleinde studentendatasets geen effectieve manier blijkt te zijn om de performantie te verhogen. Alle ML-modellen scoren veel lager op alle evaluatiemetrieken bij de PCA-datasets. Dat betekent dat de features die hier zijn gebruikt wel allemaal predictieve waarde met zich meedroegen en dat de hogere dimensionaliteit weinig ruis introduceert voor de ML-modellen. Verder kan er worden afgeleid dat, aangezien de scores lager liggen dan de baseline en populatiemodellen, er mogelijk zelfs sprake is van underfitting in plaats van overfitting.

Dat suggereert dat zowel de hoeveelheid beschikbare data als het aantal gebruikte features belangrijke factoren zijn voor het optimaliseren van de performantie van ML-modellen wanneer er weinig trainingsdata beschikbaar is. Deze hypothese wordt ondersteund door de studie van Kim et al. (2024) [23], waarin het Compact Data Learning-framework werd toegepast en de trainingsets zowel in aantal samples als in aantal features werden gereduceerd en geoptimaliseerd. In het experiment uitgevoerd in deze studie blijkt PCA effectief voor een dataset met relatief veel features (21) en een grote klasse-imbalans (diabetesdataset). Anderzijds heeft PCA mogelijk geleid tot underfitting bij een dataset met eveneens veel features (36), maar

met een gematigde klasse-imbalans (studentendataset). PCA kan bijgevolg leiden tot goede prestaties, zelfs bij beperkte datahoeveelheden, maar het effect ervan is sterk afhankelijk van de specifieke use case.

5.3 Oversamplingmethoden

Uit figuren 6 - 9 blijkt dat bij het toepassen van oversamplingtechnieken, zoals de verschillende SMOTE-varianten en ADASYN, de ensemblemethoden RF en XGB veel sterker overfitten op de trainingsdata. Toch zijn de accuracy scores van deze modellen bij gebruik van oversamplingtechnieken vergelijkbaar met de baseline en tevens ook hoger dan die van de populatiemodellen. Daarnaast valt op dat LogReg, ondanks het feit dat het minder goed presteert dan de baseline, consistent beter scoort op recall ten opzichte van XGB en RF. Een mogelijke reden hiervoor is dat LogReg een parametrisch model is en daardoor minder sterk op de trainingsdata overfit dan XGB en RF. Alle ML-modellen scoren echter wel lager dan de baseline op de recallmetriek. Met betrekking tot precision behalen alle oversamplingtechnieken voor ieder model scores die vergelijkbaar zijn met de baseline. Het feit dat de recall scores lager liggen dan de baseline heeft vermoedelijk te maken met de beperkte effectiviteit van deze augmentatiemethoden op de verkleinde datasets. Daarbij dient te worden opgemerkt dat beide datasets oorspronkelijk een klasse-imbalans vertoonden, waarbij de diabetesdataset een sterke imbalans kende. Hierdoor kan een hoge accuracyscore soms eenvoudig bekomen worden door voornamelijk de meerderheidsklasse te voorspellen. Verder geldt voor beide datasets dat een hoge recall- of precision score op zichzelf weinig inzicht biedt in de performantie van het model, aangezien beide metrieken afzonderlijk eenvoudig te optimaliseren zijn ten koste van de andere. Een hoge recall kan bijvoorbeeld bereikt worden door enkel positieve instanties te voorspellen, terwijl een hoge precision gerealiseerd kan worden door gebruik te maken van een extreem conservatief model dat alleen positieve voorspellingen maakt wanneer het zeer zeker is.

Figuren 15 - 18 tonen dat de recall scores voor het XGB-model voor de oversamplingtechnieken relatief dicht bij de baseline liggen. De precision scores van de oversamplingmethoden voor RF en XGB liggen over het algemeen gemiddeld marginaal hoger dan de respectievelijke baseline. Dat suggereert dat de vier oversamplingmethoden een positief effect hebben gehad op de precisionscore, waardoor de modellen door het oversamplen ervoor zorgden dat er meer van hun geclassificeerde vroegtijdige schoolverlaters daadwerkelijk schoolverlaters waren. Zowel XGB als RF lijken niet direct meer conservatief aangezien de recall scores voor beide modellen min of meer gelijk gebleven zijn aan de baseline. Dat kan impliceren dat de oversamplingtechnieken bij XGB en RF de trainingsets van de verkleinde studentendatasets voldoende verrijkt hebben om effectief minder vals positieve schoolverlaters te voorspellen. Figuren 15 - 18 tonen dat de effectieve stijging in performantie in absolute cijfers eerder gering is. Dat wordt ondersteund met de statistische analyse die op de verkleinde studentendatasets uitgevoerd is. Voor het RF-model hebben de oversamplingtechnieken SMOTE, ADASYN, BorderlineSMOTE en SVMSMOTE statistisch significante hogere F1-scores opgeleverd ten opzichte van de baseline. Deze methoden hebben ook gezorgd voor een statistisch significant hogere F1-score bij het XGB-model. Tabel 14 toont echter dat de stijging in F1-score zeer gering is (maar wel significant), met een gemiddelde toename van minder dan 1 procentpunt.

In deze studie zijn oversamplingtechnieken toegepast op twee datasets met klasse-imbalans. Bij de studentendataset, die een gematigde klasse-imbalans vertoont, blijken de oversamplingtechnieken een beperkt maar positief effect te hebben. Bij de diabetesdataset, die een sterkere klasse-imbalans vertoont, blijken de oversamplingtechnieken geen effect te hebben, wat enigszins verrassend is gegeven de mate van de klasse-imbalans in deze dataset. Een mogelijke verklaring hiervoor is dat de oversamplingalgoritmen niet voldoende in staat waren om kwalitatieve synthetische instanties van de minderheidsklasse te genereren. Een andere mogelijke verklaring ligt bij het eerder genoemde hoge aantal features. De hoge dimensionaliteit zorgde vermoedelijk voor extra ruis in de data, welke vervolgens ook aanwezig bleef in de oversamplede datasets en misschien ook wel werd overgenomen door de oversamplingalgoritmes.

5.4 TDA methoden

Figuur 10 toont dat bij verrijking van de verkleinde diabetesdatasets met CTGAN-synthetische data de recall scores van alle drie de ML-modellen gemiddeld hoger liggen dan hun respectievelijke baseline, hoewel de breedte van de boxplots aangeeft dat er veel variatie aanwezig is in de voorspellingen. Voor accuracy en precision vertonen de modellen gemiddeld vergelijkbare resultaten met de baseline. Uit figuur 19 blijkt dat

bij toepassing van CTGAN op de studentendatasets, de accuracy- en recall scores gemiddeld lager uitvallen dan de baseline, terwijl de precision scores vergelijkbare niveaus bereiken. Voor beide datasets valt verder op dat bij LogReg, en vanaf bepaalde trainingsgroottes ook bij XGB en RF, de testscores hoger liggen dan de trainingsscores. Dat is zeer opmerkelijk omdat het impliceert dat de ML-modellen de geziene trainingsdata minder goed kunnen voorspellen dan de ongeziene testdata. Een mogelijke verklaring hiervoor is dat CTGAN complexe instanties heeft toegevoegd aan de trainingssets, waardoor deze instanties binnen één trainingset moeilijker te voorspellen waren. Omdat de testset uitsluitend uit reële data bestaat, die mogelijks minder complex is, kon het model beter generaliseren. Desondanks het schijnbaar groter generalisatievermogen, tonen de resultaten aan dat de ML-modellen, getraind met CTGAN-verrijkte datasets, in termen van accuracy en precision gemiddeld slechter presteerden dan de baseline.

Figuur 11 laat zien dat bij toepassing van de augmentatiemethode TVAE de accuracy scores voor de diabetesdataset relatief hoog zijn ten opzichte van de baseline, en dat de precision score bij het RF-model marginale verbetering vertoont. De recall scores daarentegen liggen voor alle modellen onder de baseline. Deze individuele metriekwaarden zeggen ook hier op zichzelf weinig, aangezien bij een ongebalanceerde dataset zoals de diabetesdataset vooral de trade-off tussen recall en precision van belang is (zie sectie 5.3). Omdat de precision scores vergelijkbaar of soms hoger liggen, terwijl de recall scores ver onder de baseline blijven, heeft TVAE er waarschijnlijk voornamelijk voor gezorgd dat de ML-modellen conservatiever werden in het toekennen van een positief (diabetes) label. Figuur 20 toont dat de scores van alle evaluatiemetrieken bij de TVAE-verrijkte studentendatasets gemiddeld ongeveer gelijk zijn aan of iets lager liggen dan de baseline. De relatief gematigde klasse-imbalans lijkt in deze dataset ervoor te zorgen dat de recall scores niet sterk afwijken van de baseline, wat suggereert dat de modellen niet conservatiever geworden zijn zoals bij de diabetesdatasets. Desondanks bereiken de ML-modellen niet het prestatieniveau van de baseline, wat impliceert dat TVAE er niet in is geslaagd om via synthetische data een groter deel van de datadistributie bloot te stellen aan de modellen.

Bovenstaande bevindingen worden ook ondersteund door de statistische analyse die uitgevoerd is op de CTGAN- en TVAE-verrijkte studenten- en diabetesdatasets. Voor alle drie de ML-modellen bij zowel de studenten- als de diabetesdatasets werd bevonden dat CTGAN en TVAE gemiddeld lagere F1-scores opleverde dan de baseline.

Over het algemeen blijkt dat de pure TDA-methoden geen directe positieve impact hebben op de prestaties van de ML-modellen in een small-data scenario. CTGAN lijkt complexe synthetische instanties aan de trainingsdata toe te voegen die eerder dienen als extra ruis dan als een uitbreiding van de onderliggende datadistributie. Ook TVAE vertoont niet direct een positief effect op de resultaten van de ML-modellen, hoewel het bij de studentendataset de performantie ook niet negatief beïnvloedt. Dat kan betekenen dat TVAE mogelijk beter presteert wanneer er sprake is van een meer gebalanceerde dataset. Daarnaast werd voor TVAE ook een alternatieve implementatie toegepast voor het genereren van synthetische data (zie sectie 3.4), wat mogelijk ook een invloed heeft op de kwaliteit van de gegenereerde samples en daardoor op de prestaties van de modellen. Ook werd voor zowel TVAE als CTGAN gebruikgemaakt van standaard hyperparameterinstellingen, wat mogelijk heeft geleid tot een suboptimale training van beide methodes op de beschikbare trainingdata. Dat kan ertoe geleid hebben dat de onderliggende neurale netwerken de datadistributie onvoldoende hebben kunnen leren, en daardoor synthetische data van lage kwaliteit genereerden.

5.5 Antwoord op de onderzoeksvragen

Wat betreft OV1 kan er gesteld worden dat de metrieken van kleine datasets over het algemeen iets lager zijn dan hun equivalente populatiemetrieken. Dat is een verwachte uitkomst. Het verschil met de populatiemodellen voor beide datasets is echter wel zeer gering. Daarom kan er gesteld worden dat ook de resultaten gelijkaardig zijn aan de populatiemodellen. Qua accuraatheid scoren de ensemblemodellen RF en XGB vaak iets beter dan het LogReg model. Het LogReg model doet het over het algemeen iets beter bij de precision en recall scores. Verder wordt ook vastgesteld dat de populatiemodellen hun performantie pas volledig benaderd wordt wanneer de trainingsset minstens 500 datapunten heeft. Daaronder is er toch nog wel een merkbaar verschil tussen performantie van de populatiemodellen en de performantie van het small-data scenario. Er bestaat dan ook geen eenduidig ‘beste model’ voor elk small-data scenario, de optimale modelkeuze hangt sterk af van het doel van de ML-taak.

Wat betreft OV2 kan er gesteld worden dat niet alle augmentatietechnieken effectief zijn bij elk ML-model. Een observatie die consistent is met de studie van Wanigasekar et al. (2018) [63]. De combinatie augmentatietechniek en ML-model is wel degelijk van belang. Bij de diabetesdataset verbeterde enkel de dimensiereductietechniek PCA de performantie. Dat kan ook kloppen gegeven de hoeveelheid features (zie tabel 1) die de diabetesdataset bevat. De augmentatietechnieken bij deze dataset zijn er niet in geslaagd om de F1-scores statistisch significant te verbeteren. Voor de studentendataset zijn er enkel bij het XGB-model en RF-model augmentatiemethodes, zijnde de vier oversamplingmethodes, geïdentificeerd die de performantie van de modellen bevorderd hebben. De bevorderingen waren echter vrij gering, amper 1 procentpunt. Over het algemeen kan er gesteld worden dat in deze studie de datasets verrijkt met de verschillende augmentatiemethodes er amper tot niet in geslaagd zijn om de performantie van de ML-modellen verbeteren ten opzichte van de datasets die niet verrijkt zijn met synthetische data.

6 Conclusie

Het onderzoek in deze paper heeft op basis van twee benchmarkdatasets geprobeerd de effectiviteit van ML-modellen te meten in een small-data scenario. Er zijn tevens verschillende manieren onderzocht om de beperkte trainingsdata te verrijken om zo de ML-modellen toch bloot te kunnen stellen aan een groter deel van de onderliggende datadistributie. Zo zijn er verschillende onderzoeksstromen geïdentificeerd die veelbelovend zijn op het gebied van small-data scenario's.

Verder is er uit de bevindingen van het experiment gebleken dat op de diabetesdataset waarbij er inherent een grote klasse-imbalans is, de augmentatiemethoden niet effectief waren om de F1-scores van de geteste ML-modellen (XGB, RF en LogReg) te verbeteren. Door eerst een Principal Component Analysis uit te voeren op de trainingsets verbeterde de F1-scores van de diabetesdataset sterk. Hierin was een toename van bijna 20 procentpunten waargenomen. Ook blijft er bij vrijwel alle trainingsgroottes een relatief hoge graad van overfitting aanhouden. Dat vermindert enigszins wanneer de trainingsgroottes vergroten of wanneer er een model gekozen wordt met inherent minder representatiecapaciteit zoals een parametrisch model, in deze studie logistische regressie.

Uit de resultaten blijkt eveneens dat in de studentendataset, met een matige klasse-imbalans, geen enkele augmentatietechniek of dimensiereductiemethode erin slaagt om voor het LogReg-model de F1-scores boven de baseline te krijgen. Voor RF en XGB werd echter bij de met SMOTE, BorderlineSMOTE, SVMSMOTE en ADASYN verrijkte datasets wel een statistisch significant verschil in F1-scores bevonden, maar de effectgrootte blijft met een toename van ongeveer 1 procentpunt zeer beperkt.

Op basis van deze resultaten kan er geconcludeerd worden dat kleine datasets wel gelijkaardige resultaten kunnen behalen als ML-modellen getraind op grote datasets (OV1). Daarnaast slagen de augmentatiemethoden er over het algemeen amper tot niet in om een toename in performantie te veroorzaken bij de ML-modellen (OV2). Indien er wel een significante toename in F1-score na augmentatie werd waargenomen, bedroeg deze gemiddeld slechts 1 procentpunt. Een dimensiereductie met behulp van PCA was wel effectief om de F1-scores van ML-modellen voor de diabetesdataset te verhogen met ongeveer 20 procentpunten.

Er kan dus gesteld worden dat de combinatie van het ML-model en de toegepaste datatransformaties, of het nu gaat om data-augmentatie, dimensiereductie of een andere transformatie, zeer belangrijk is [63]. Deze combinatie zou voor elke use case apart getest moeten worden om te kijken welke de beste resultaten oplevert.

7 Limitaties en toekomstig werk

Deze studie geeft inzicht in de effectiviteit van ML-modellen bij beperkte trainingsdata. Daarbij zijn verschillende technieken, zoals oversampling en tabulaire data-augmentatie, geëvalueerd op hun vermogen om datasets van verschillende groottes te verrijken. Deze studie kent echter wel enkele beperkingen en biedt verschillende mogelijkheden voor vervolgonderzoek.

Zo is de evaluatie uitgevoerd op een beperkt aantal datasets. Toekomstig onderzoek kan daarom een grotere variëteit aan datasets omvatten om te onderzoeken wat de performantie is van de gebruikte ML-modellen en technieken op die specifieke datasets. Bovendien zijn in dit onderzoek alleen binaire classificatiedatasets gebruikt. Verder onderzoek naar regressiedatasets kan van toegevoegde waarde zijn om te bepalen of kleine en eventueel geaugmenteerde datasets in staat zijn om accurate voorspellingen te maken van numerieke

waarden. Daarnaast kan het ook waardevol zijn om te onderzoeken hoe performant small-data scenario's zijn bij multi-label en multi-class classificatie taken.

Verder wordt er in de literatuur, bijvoorbeeld [5, 10] nog verschillende andere augmentatietechnieken beschreven, maar die worden vaak niet onderling vergeleken op dezelfde datasets. Toekomstige studies kunnen daarom overwegen om meerdere technieken te testen, zodat de sterke en zwakke punten van elke techniek gezamenlijk beoordeeld kunnen worden. In deze studie bleef het aantal geanalyseerde methoden namelijk maar beperkt tot acht, een uitbreiding hiervan kan nieuwe inzichten opleveren. Daarnaast zijn in deze studie de augmentatietechnieken enkel onderling getest. Zo is bijvoorbeeld de combinatie tussen het reduceren van de dimensies door middel van een dimensiereductiealgoritme zoals PCA in combinatie met een augmentatietechniek niet getest. Ook de combinatie van meerdere augmentatietechnieken zou dus verder onderzocht kunnen worden.

Daarnaast is er in de literatuur sprake van het feit dat grote Large Language Models (LLM's) als tabulaire datageneratoren of zelfs als classifiers/regressors kunnen dienen [3, 7]. Toekomstig werk kan onderzoeken hoe goed LLM's datasets kunnen verrijken en of ze accurate classificaties of regressievoorspellingen kunnen doen. Hierbij moet wel rekening gehouden worden met mogelijke data-lekkage: commerciële LLM's zoals ChatGPT zijn getraind op enorme hoeveelheden data die op het internet beschikbaar is. Daardoor is het mogelijk dat bepaalde openbare benchmarkdatasets al in hun trainingsdata omvat zit. Dat zou betekenen dat de generaliseerbaarheid van zo een LLM als classifier niet gegarandeerd kan worden, tenzij de data waarop het commerciële LLM getraind is openbaar is. Een mogelijkheid om dit toch te testen is door het gebruiken van een privé dataset die gelijkaardig is aan één van de vele benchmarkdatasets.

Verder is in deze studie bij alle augmentatiemethoden gebruikgemaakt van standaard hyperparameters. In vervolgonderzoek kan het optimaliseren van die hyperparameters per methode en dataset bijdragen aan betere resultaten. Verder lag de focus in deze studie op tree-based ensemblemodellen, toekomstige studies zouden kunnen onderzoeken hoe andere soorten ML-modellen (bijvoorbeeld neurale netwerken of support vector machines) presteren op kleine en geaugmenteerde tabulaire datasets.

Referenties

[1] Atzeni, D., Ramjattan, R., Figliè, R., Baldi, G., & Mazzei, D. (2023). Data-Driven Insights through Industrial Retrofitting: An Anonymized Dataset with Machine Learning Use Cases. *Sensors*, *23*(13), 6078. https://doi.org/10.3390/s23136078

[2] Bettoni, A., Matteri, D., Montini, E., Gładysz, B., & Carpanzano, E. (2021). An AI adoption model for SMEs: a conceptual framework. *IFAC-PapersOnLine*, *54*(1), 702–708. https://doi.org/10.1016/j.ifacol.2021.08.082

[3] Borisov, V., Seßler, K., Leemann, T., Pawelczyk, M., & Kasneci, G. (2023, april). Language Models are Realistic Tabular Data Generators. https://doi.org/10.48550/arXiv.2210.06280

[4] Chaudhary, P. S., Khurana, M. R., & Ayalasomayajula, M. (2024). Real-World Applications of Data Analytics, Big Data, and Machine Learning. In *Data Analytics and Machine Learning* (pp. 237–263). Springer, Singapore. Verkregen maart 18, 2025, van https://link.springer.com/chapter/10.1007/978-981-97-0448-4_12

[5] Cui, L., Li, H., Chen, K., Shou, L., & Chen, G. (2024, juli). Tabular Data Augmentation for Machine Learning: Progress and Prospects of Embracing Generative AI. https://doi.org/10.48550/arXiv.2407.21523

[6] Dohmatob, E., Feng, Y., Subramonian, A., & Kempe, J. (2024, oktober). Strong Model Collapse. https://doi.org/10.48550/arXiv.2410.04840

[7] Dong, H., & Wang, Z. (2024). Large Language Models for Tabular Data: Progresses and Future Directions. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2997–3000. https://doi.org/10.1145/3626772.3661384

[8] Feng, S., Zhou, H., & Dong, H. (2019). Using deep neural network with small dataset to predict material defects. *Materials & Design*, *162*, 300–310. https://doi.org/10.1016/j.matdes.2018.11.060

[9] Feng, X., & Kim, S.-K. (2024). Novel Machine Learning Based Credit Card Fraud Detection Systems. *Mathematics*, *12*(12), 1869. https://doi.org/10.3390/math12121869

[10] Fonseca, J., & Bacao, F. (2023). Tabular and latent space synthetic data generation: a literature review. *Journal of Big Data*, *10*(1), 1–37. https://doi.org/10.1186/s40537-023-00792-7

[11] Gharoun, H., Momenifar, F., Chen, F., & Gandomi, A. (2024). Meta-learning Approaches for Few-Shot Learning: A Survey of Recent Advances. *ACM Comput. Surv.*, *56*(12), 294:1–294:41. https://doi.org/10.1145/3659943

[12] Gui, J., Chen, T., Zhang, J., Cao, Q., Sun, Z., Luo, H., & Tao, D. (2024). A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(12), 9052–9071. https://doi.org/10.1109/TPAMI.2024.3415112

[13] Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2023). A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 3313–3332. https://doi.org/10.1109/TKDE.2021.3130191

[14] Guo, K., Chen, J., Qiu, T., Guo, S., Luo, T., Chen, T., & Ren, S. (2023). MedGAN: An adaptive GAN approach for medical image generation. *Computers in Biology and Medicine*, *163*, 107119. https://doi.org/10.1016/j.compbiomed.2023.107119

[15] Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2022). Meta-Learning in Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5149–5169. https://doi.org/10.1109/TPAMI.2021.3079209

[16] Hutter, F., Hoos, H., & Leyton-Brown, K. (2014). An Efficient Approach for Assessing Hyperparameter Importance. *Proceedings of the 31st International Conference on Machine Learning*, 754–762. Verkregen november 30, 2024, van https://proceedings.mlr.press/v32/hutter14.html

[17] Iyelolu, T. V., Agu, E. E., Idemudia, C., & Ijomah, T. I. (2024). Driving SME innovation with AI solutions: overcoming adoption barriers and future growth opportunities. *International Journal of Science and Technology Research Archive*, *7*(1), 036–054. https://doi.org/10.53771/ijstra.2024.7.1.0055

[18] Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2021). A Survey on Contrastive Self-Supervised Learning. *Technologies*, *9*(1), 2. https://doi.org/10.3390/technologies9010002

[19] Kadam, S., & Vaidya, V. (2020). Review and Analysis of Zero, One and Few Shot Learning Approaches. In A. Abraham, A. K. Cherukuri, P. Melin & N. Gandhi (Red.), *Intelligent Systems Design and Applications* (pp. 100–112). Springer International Publishing. https://doi.org/10.1007/978-3-030-16657-1_10

[20] Käppel, M., Schönig, S., & Jablonski, S. (2021). Leveraging Small Sample Learning for Business Process Management. *Information and Software Technology*, *132*, 106472. https://doi.org/10.1016/j.infsof.2020.106472

[21] Kedi, W. E., Ejimuda, C., Idemudia, C., & Ijomah, T. I. (2024). AI Chatbot integration in SME marketing platforms: Improving customer interaction and service efficiency. *International Journal of Management & Entrepreneurship Research*, *6*(7), 2332–2341. https://doi.org/10.51594/ijmer.v6i7.1327

[22] Kim, D.-K., Ryu, D., Lee, Y., & Choi, D.-H. (2024). Generative models for tabular data: A review. *Journal of Mechanical Science and Technology*, *38*(9), 4989–5005. https://doi.org/10.1007/s12206-024-0835-0

[23] Kim, S.-K. ((2024). Compact Data Learning for Machine Learning Classifications. *Axioms*, *13*(3), 137. https://doi.org/10.3390/axioms13030137

[24] Kim, S.-K. A. (2024). Compact Data Learning For Time-Series Forecasting Systems [ISSN: 2832-9848]. *2024 6th International Conference on Electrical, Control and Instrumentation Engineering (ICECIE)*, 1–5. https://doi.org/10.1109/ICECIE63774.2024.10815647

[25] Kling, N., Haugk, S., & Gebauer, H. (2025). Towards a Data-Driven Organisation: Making Data a Strategic Knowledge Asset in SMEs. *Journal of the Knowledge Economy*, 1–19. https://doi.org/10.1007/s13132-025-02631-x

[26] Lewy, D., & Mańdziuk, J. (2023). An overview of mixing augmentation methods and augmentation strategies. *Artificial Intelligence Review*, *56*(3), 2111–2169. https://doi.org/10.1007/s10462-022-10227-z

[27] L'Heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. M. (2017). Machine Learning With Big Data: Challenges and Approaches. *IEEE Access*, *5*, 7776–7797. https://doi.org/10.1109/ACCESS.2017.2696365

[28] Liu, D., He, Z., Chen, D., & Lv, J. (2020). A Network Framework for Small-Sample Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(10), 4049–4062. https://doi.org/10.1109/TNNLS.2019.2951803

[29] Liu, P., Wang, X., Xiang, C., & Meng, W. (2020). A Survey of Text Data Augmentation. *2020 International Conference on Computer Communication and Network Security (CCNS)*, 191–195. https://doi.org/10.1109/CCNS50731.2020.00049

[30] Machado, P., Fernandes, B., & Novais, P. (2022). Benchmarking Data Augmentation Techniques for Tabular Data. In H. Yin, D. Camacho & P. Tino (Red.), *Intelligent Data Engineering and Automated Learning – IDEAL 2022* (pp. 104–112). Springer International Publishing. https://doi.org/10.1007/978-3-031-21753-1_11

[31] Manousakas, D., & Aydöre, S. (2023, juni). On the Usefulness of Synthetic Tabular Data Generation. https://doi.org/10.48550/arXiv.2306.15636

[32] Martins, M. V., Tolledo, D., Machado, J., Baptista, L. M. T., & Realinho, V. (2021). Early Prediction of student's Performance in Higher Education: A Case Study. In Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira

& A. M. Ramalho Correia (Red.), *Trends and Applications in Information Systems and Technologies* (pp. 166–175). Springer International Publishing. https://doi.org/10.1007/978-3-030-72657-7_16

[33] Mohamed, M., & Weber, P. (2020). Trends of digitalization and adoption of big data & analytics among UK SMEs: Analysis and lessons drawn from a case study of 53 SMEs. *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 1–6. https://doi.org/10.1109/ICE/ITMC49519.2020.9198545

[34] Mohib, M., Khan, F. K., Burari, E. R. E., & Ali, S. (2025). The Challenges and Limitations of Artificial Intelligence Adoption in Small and Medium-Sized Enterprises [Number: 1]. *Review Journal of Social Psychology & Social Works*, *3*(1), 292–303. https://doi.org/10.71145/rjsp.v3i1.99

[35] Muminova, E., Ashurov, M., Akhunova, S., & Turgunov, M. (2024). AI in Small and Medium Enterprises: Assessing the Barriers, Benefits, and Socioeconomic Impacts. *2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, *1*, 1–6. https://doi.org/10.1109/ICKECS61492.2024.10616816

[36] Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

[37] Nanga, S., Bawah, A. T., Acquaye, B. A., Billa, M.-I., Baeta, F. D., Odai, N. A., Obeng, S. K., & Nsiah, A. D. (2021). Review of Dimension Reduction Methods [Number: 3 Publisher: Scientific Research Publishing]. *Journal of Data Analysis and Information Processing*, *9*(3), 189–231. https://doi.org/10.4236/jdaip.2021.93013

[38] Niu, S., Liu, Y., Wang, J., & Song, H. (2020). A Decade Survey of Transfer Learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, *1*(2), 151–166. https://doi.org/10.1109/TAI.2021.3054609

[39] Nyíri, T. B., & Kiss, A. (2023). What can we learn from Small Data. *INFOCOMMUNICATIONS JOURNAL*, *15*(SI), 27–34. https://doi.org/10.36244/ICJ.2023.5.5

[40] Ouali, Y., Hudelot, C., & Tami, M. (2020, juni). An Overview of Deep Semi-Supervised Learning. Verkregen maart 19, 2025, van https://arxiv.org/abs/2006.05278v2

[41] Ouyang, B., Song, Y., Li, Y., Wu, F., Yu, H., Wang, Y., Yin, Z., Luo, X., Sant, G., & Bauchy, M. (2021). Using machine learning to predict concrete's strength: learning from small datasets. *Engineering Research Express*, *3*(1), 015022. https://doi.org/10.1088/2631-8695/abe344

[42] Panigrahi, S., Nanda, A., & Swarnkar, T. (2021). A Survey on Transfer Learning [ISSN: 2190-3026]. In *Intelligent and Cloud Computing* (pp. 781–789). Springer, Singapore. https://doi.org/10.1007/978-981-15-5971-6_83

[43] Parnami, A., & Lee, M. (2022, maart). Learning from Few Examples: A Summary of Approaches to Few-Shot Learning. https://doi.org/10.48550/arXiv.2203.04291

[44] Patki, N., Wedge, R., & Veeramachaneni, K. (2016). The Synthetic Data Vault. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 399–410. https://doi.org/10.1109/DSAA.2016.49

[45] Rani, V., Nabi, S. T., Kumar, M., Mittal, A., & Kumar, K. (2023). Self-supervised Learning: A Succinct Review. *Archives of Computational Methods in Engineering*, *30*(4), 2761–2775. https://doi.org/10.1007/s11831-023-09884-2

[46] Rather, I. H., Kumar, S., & Gandomi, A. H. (2024). Breaking the data barrier: a review of deep learning techniques for democratizing AI with small datasets. *Artificial Intelligence Review*, *57*(9), 226. https://doi.org/10.1007/s10462-024-10859-3

[47] Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–15. https://doi.org/10.1145/3411764.3445518

[48] Shaikhina, T., Lowe, D., Daga, S., Briggs, D., Higgins, R., & Khovanova, N. (2015). Machine Learning for Predictive Modelling based on Small Data in Biomedical Engineering. *IFAC-PapersOnLine*, *48*(20), 469–474. https://doi.org/10.1016/j.ifacol.2015.10.185

[49] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

[50] Shu, J., Xu, Z., & Meng, D. (2018). Small Sample Learning in Big Data Era. *CoRR*, *abs/1808.04572*. http://arxiv.org/abs/1808.04572

[51] Sivarajah, U., Kumar, S., Kumar, V., Chatterjee, S., & Li, J. (2024). A study on big data analytics and innovation: From technological and business cycle perspectives. *Technological Forecasting and Social Change*, *202*, 123328. https://doi.org/10.1016/j.techfore.2024.123328

[52] Song, Y., Wang, T., Cai, P., Mondal, S. K., & Sahoo, J. P. (2023). A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities. *ACM Comput. Surv.*, *55*(13s), 271:1–271:40. https://doi.org/10.1145/3582688

[53] Sutojo, T., Rustad, S., Akrom, M., Syukur, A., Shidik, G. F., & Dipojono, H. K. (2023). A machine learning approach for corrosion small datasets. *npj Materials Degradation*, *7*(1), 1–10. https://doi.org/10.1038/s41529-023-00336-7

[54] Tao, Q., Yu, J., Mu, X., Jia, X., Shi, R., Yao, Z., Wang, C., Zhang, H., & Liu, X. (2025). Machine learning strategies for small sample size in materials science. *Science China Materials*, *68*(2), 387–405. https://doi.org/10.1007/s40843-024-3204-5

[55] Tran, N.-T., Tran, V.-H., Nguyen, N.-B., Nguyen, T.-K., & Cheung, N.-M. (2021). On Data Augmentation for GAN Training. *IEEE Transactions on Image Processing*, *30*, 1882–1897. https://doi.org/10.1109/TIP.2021.3049346

[56] Uddin, S., & Lu, H. (2024). Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data. *PLOS ONE*, *19*(4), e0301541. https://doi.org/10.1371/journal.pone.0301541

[57] Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PLOS ONE*, *14*(11), e0224365. https://doi.org/10.1371/journal.pone.0224365

[58] Vajjhala, N. R. (2024). Exploratory Review of Applications of Machine Learning for Small- and Medium-Sized Enterprises (SMEs). In S. Bhattacharyya, J. S. Banerjee & M. Köppen (Red.), *Human-Centric Smart Computing* (pp. 261–270). Springer Nature. https://doi.org/10.1007/978-981-99-7711-6_21

[59] van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6

[60] Wang, A. X., Chukova, S. S., Simpson, C. R., & Nguyen, B. P. (2024). Challenges and opportunities of generative models on tabular data. *Applied Soft Computing*, *166*, 112223. https://doi.org/10.1016/j.asoc.2024.112223

[61] Wang, J. (2024). A Comprehensive Survey on Meta-Learning: Applications, Advances, and Challenges. https://doi.org/10.36227/techrxiv.172840352.29141687/v1

[62] Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.*, *53*(3), 63:1–63:34. https://doi.org/10.1145/3386252

[63] Wanigasekara, C., Swain, A., Nguang, S. K., & Prusty, B. G. (2018). Improved Learning from Small Data Sets Through Effective Combination of Machine Learning Tools with VSG Techniques [ISSN: 2161-4407]. *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–6. https://doi.org/10.1109/IJCNN.2018.8489759

[64] Wen, J., Su, A., Wang, X., Xu, H., Ma, J., Chen, K., Ge, X., Xu, Z., & Lv, Z. (2025). Virtual sample generation for small sample learning: A survey, recent developments and future prospects. *Neurocomputing*, *615*, 128934. https://doi.org/10.1016/j.neucom.2024.128934

[65] Xie, L., Lin, K., Wang, S., Wang, F., & Zhou, J. (2018, februari). Differentially Private Generative Adversarial Network. https://doi.org/10.48550/arXiv.1802.06739

[66] Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling Tabular data using Conditional GAN. *Advances in Neural Information Processing Systems*, *32*. https://proceedings.neurips.cc/paper/2019/hash/254ed7d2de3b23ab10936522dd547b78-Abstract.html

[67] Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2023, november). Image Data Augmentation for Deep Learning: A Survey. https://doi.org/10.48550/arXiv.2204.08610

[68] Yurshin, V. G., & Stanovov, V. V. (2023). Artificial Neural Network Architecture Tuning Algorithm. *European Proceedings of Computers and Technology*, *Hybrid Methods of Modeling and Optimization in Complex Systems*. https://doi.org/10.15405/epct.23021.29

[69] Zhang, S., & Balog, K. (2020). Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Trans. Intell. Syst. Technol.*, *11*(2), 13:1–13:35. https://doi.org/10.1145/3372117

[70] Zhao, J., Kong, L., & Lv, J. (2025). An Overview of Deep Neural Networks for Few-Shot Learning. *Big Data Mining and Analytics*, *8*(1), 145–188. https://doi.org/10.26599/BDMA.2024.9020049

[71] Zhou, X., & Belkin, M. (2014, januari). Chapter 22 - Semi-Supervised Learning. In P. S. R. Diniz, J. A. K. Suykens, R. Chellappa & S. Theodoridis (Red.), *Academic Press Library in Signal Processing* (pp. 1239–1269, Deel 1). Elsevier. Verkregen maart 21, 2025, van https://www.sciencedirect.com/science/article/pii/B978012396502800022X

[72] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, *109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555