UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Sales forecasting algoritmen: een benchmarking-studie

Brent Vranken

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

PROMOTOR :

dr. Frank VANHOENSHOVEN

2021
2022

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

Sales forecasting algoritmen: een benchmarking-studie

Brent Vranken

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

PROMOTOR :

dr. Frank VANHOENSHOVEN

Voorwoord

Wat volgt is een benchmarking studie over 'Sales Forecasting' algoritmen. Deze studie bekijkt verschillende voorspellingsmethoden om te bepalen welke het meest bruikbaar zijn in verschillende omstandigheden en op verschillende datasets. Dit artikel is geschreven in het kader van mijn masterproef die gelinkt is aan het afronden van mijn master opleiding Handelsingenieur in de beleidsinformatica (BI) aan de Universiteit van Hasselt (UHasselt).

Dit onderzoek werd mede mogelijk gemaakt en werd voornamelijk uitgevoerd in opdracht voor het bedrijf Acumen Consulting BV in Leuven. Zij hebben enkele tijdreeks-datasets aangeleverd waar voorspellingsmodellen op konden worden toegepast. Zowel de promotor als ikzelf bedanken Acumen en zijn medewerkers, die geholpen hebben tijdens het onderzoek en die feedback hebben voorzien. Ook wil ik mijn promotor Frank Vanhoenshoven bedanken voor de begeleiding, feedback en inzichten doorheen het hele proces.

Sales Forecasting Algoritmes: een benchmarking-studie

Vranken Brent[1746853]

Universiteit van Hasselt, België Brent.vranken@student.uhasselt.be
https://www.uhasselt.be/

Abstract. Bedrijven willen zo veel mogelijk ondersteuning krijgen bij het nemen van beslissingen om het meeste waarde uit hun beslissingen te kunnen halen. Dit kan ondersteuning zijn op gebied van ervaring en kennis (subjectief) van experten, maar ook op basis van rekenkundige / statistisch onderbouwde beargumenteringen (objectief). De retailsector heeft door een stijging van online verkoop en doordat het zich bevindt in een sterk competitieve omgeving, nood aan goede beslissingsondersteuning. Er zijn veel beslissingen die genomen moeten worden rond: Marketing, Sales, Finance, ... In retail wordt er voor objectieve standpunten vaak gebruik gemaakt van tijdreeks verkoopsvoorspellingen, die voorspellen hoeveel verkopen er zullen zijn binnen een bepaalde tijd. Huidige voorspellingsmethodes hebben allemaal hun voor- en nadelen en ze presteren beter in verschillende situaties. Machine Learning (ML) modellen willen tot een optimalisatie komen van een bepaalde loss-functie, terwijl statistische methodes op basis van een theorie en vergelijking vasthouden aan bepaalde wiskundige formules. Er zijn verschillende packages en libraries te vinden, die voorspellingsmethodes aanbieden, maar de vraag is welke methodes het best passen in een specifieke situatie. Uit dit onderzoek bleek er dat AutoETS, Theta en Recurrent Neural Networks over verschillende datasets heen goede voorspellingen produceerden. Verder werd er een indicatie gevonden dat voorspellingen op meer geaggregeerde data betere voorspellingen leveren dan op lager aggregatieniveau. Ook de invloed van de coronacrisis op de data en de bijbehorende voorspellingen werden onderzocht en het bleek dat de prestaties veel afweken per methode en per dataset die gebruikt werd.

Keywords: Sales Forecasting, Python, Data Science, Decision support, Benchmarking, Machine learning, Econometrics, Time-series

1 Introductie

Context. In elke sector van de maatschappij worden er voorspellingen uitgevoerd. Zo worden er bijvoorbeeld voorspellingen gedaan over de weersituatie, maar ook hoeveel mensen geïnteresseerd zijn in een bepaald evenement of product. Voorspellingen kunnen gebeuren op een kwalitatieve of kwantitatieve manier [1, 2]. Kwalitatieve voorspellingsmethodes maken gebruik van menselijke beoordelingen, die vooral gebaseerd zijn op intuïtie, ervaring, persoonlijke beoordel-

ing en emoties [3, 4]. Voor dit artikel werd er gewerkt met kwantitatieve methodes, die ook zelf nog onderverdeeld worden in tijdreeks- en causale modellen [1, 2]. Kwantitatieve modellen hebben de eigenschap dat de voorspellingsmodellen volledig gefocust zijn op historische data.

In de retailsector is het belangrijk om zo accuraat mogelijke voorspellingen te verkrijgen, omdat ze een grote impact kunnen hebben op een organisatie en zijn winstgevendheid [5]. De voorspellingen dienen als objectieve argumentering bij het nemen van financiële, operationele en strategische beslissingen [6]. Demand Forecasting of Sales Forecasting omvatten predictieve analyses, die ervoor zorgen dat klantvraag / verkopen begrepen en voorspeld kunnen worden [7]. Bij het voorspellen van verkopen wordt er voornamelijk gebruik gemaakt van tijdreeksdata [5, 2]. Langetermijnvoorspellingen ($\geq$ 1j) kunnen een invloed hebben op financieel management en financiële investeringen [8] zoals: het al dan niet uitvoeren van overnames, de allocatie van middelen, ... Anderzijds kunnen kortetermijnvoorspellingen een invloed hebben op bijvoorbeeld voorraden en de daarbij horende kost [9].

Voor het uitvoeren van tijdreeksvoorspellingen, kan er gebruik gemaakt worden van statistische of *Machine Learning* (ML) methodes [6, 7]. Beide hebben hun voor- en nadelen, en beschikken over verschillende eigenschappen. Zo zijn er de statistische methodes zoals Autoregressive Integrated Moving Average (ARIMA) en Exponential Smoothing (ES), die eenvoudig te interpreteren zijn en die betrouwbare voorspellingen geven [10, 11]. Voor complexere tijdreeksen, waar meer non-lineariteit voorkomt, is het beter om te kiezen voor ML methodes. Deze kunnen non-lineariteit en verborgen patronen in de data identificeren, wat meer potentieel geeft op accuratere voorspellingen [7], maar anderzijds ook meer kans geeft op overfitten. Overfitting wilt zeggen dat het model te sterk getraind is op de geobserveerde data waardoor deze niet meer representatief is voor niet-geobserveerde data [12].

Probleem. Er zijn doorheen de laatste decennia zeer veel tijdreeksvoorspellingsmethodes ontwikkeld en verbeterd [7]. Deze zijn niet altijd even toepasbaar, noch interpreteerbaar. Er is nood aan verduidelijking over de verschillende voorspellingsmethodes en wanneer deze het best commercieel worden toegepast.

Verder is er te weinig duidelijkheid over de prestaties van de voorspellingsmodellen. Prestaties van tijdreeksvoorspellingsmodellen kunnen door veel factoren beïnvloed worden [13]. Een model dat goed presteert op één dataset is geen garantie op goede voorspellingen op een andere dataset, aangezien de prestatie / accuraatheid vaak afhangt van de eigenschappen en bruikbaarheid van de dataset [14]. Al te vaak worden in onderzoeken gebruik gemaakt van slechts één of te weinig *accuracy measures*, waardoor er te weinig inzicht gekregen kan worden in de voorspellingsmethodes [13].

Bijdrage. In deze studie werden verschillende voorspellingsmethodes, uit verschillende Python open-source libraries, met elkaar vergeleken op basis van twee bron-datasets, afkomstig van kleding retailwinkels. De voorspellingsmethodes werden op beide bron-datasets toegepast op verschillende aggregatieniveaus: maand/week en per winkeltype/algemeen. Er werden zowel statistische als ML

methodes toegepast om te zien welke het best presteren en gebruikt kunnen worden in real-life toepassingen volgens verschillende accuracy measures, nl. Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) en Root Mean Squared Error (RMSE).

Verder werden de modellen gepaard - met behulp van een t-test - vergeleken met een voorafbepaalde benchmark (Naïeve methode). Deze benchmark werd gekozen op basis van zijn eenvoudige toepas- en interpreteerbaarheid.

Voor de verschillende winkeltypes werden de gemiddelde absolute procentuele fouten (MAPEs) vergeleken om een zicht te krijgen op de prestaties van de verschillende voorspellingsmethodes op de verschillende winkeltypes. Ook werd er gekeken naar de verschillen in prestaties wanneer er op verschillend aggregatieniveau, zoals wekelijks of maandelijks maar ook per winkeltype of algemeen met alle winkeltypes samen, voorspellingen gemaakt werden.

Als aanvulling op deze studie werd er ook gekeken naar de invloed van Covid-19 - met bijhorende beperkende maatregelen - op de voorspellingsaccuraatheid van verkoopcijfers. Dit kon meer duidelijkheid scheppen over de robuustheid van een bepaalde voorspellingsmethode of bepaalde dataset.

Validatie. De verschillende voorspellingsmethodes werden met elkaar vergeleken met behulp van de Wilcoxon Signed-Rank test [15]. Deze gepaarde t-test geeft aan wanneer twee verdelingen significant verschillen van elkaar. Deze werd toegepast op vijf procent significantie niveau. Indien de verschillende toegepaste voorspellingsmethodes vergeleken werden met de benchmark methode, kon er uitsluitsel gevonden worden over de al dan niet beter presterende methodes. Verder werden ook de *accuracy measures*: Mean Absolute Error, Mean Absolute Percentage Error en Root Mean Squared Error gebruikt om betere en slechtere voorspellingsmodellen te bepalen per dataset.

Structuur Het vervolg van de paper is als volgt gestructureerd: Sectie twee beschrijft belangrijke begrippen en voorspellingsmethodes uit de literatuur. In sectie drie wordt de methodologie besproken, waarbij er bepaalde keuzes, die gemaakt zijn in het onderzoek, ondersteund worden. Verder in sectie vier worden de resultaten besproken van verschillende experimenten. Tot slot wordt de paper samengevat met een conclusie en korte beschrijving van de limitaties van het onderzoek.

2 Literatuurstudie

In onderstaande subsecties wordt er kort ingegaan op de hoofdaspecten van verschillende toegepaste voorspellingsmodellen. Deze informatie is verkregen uit de huidige literatuur. Om niet te zeer in detail te treden over de modellen zelf en de onderliggende logica, werd er enkel rekening gehouden met de belangrijkste en meest beïnvloedende parameters en concepten.

Het onderzoek focust op het vergelijken van verschillende voorspellingsmethodes op verschillende soorten data. Deze focus zal vooral gericht zijn op *accuracy measures*, die onderling vergeleken worden, als ook de residuals/errors

die gemaakt worden bij voorspellingen. Door gelimiteerde tijd voor het uitvoeren van het onderzoek en de vereiste expertise bij het aanpassen/finetunen van modelparameters, werd er, waar mogelijk, eerder gekozen voor een automatische methode. Een eigenschap van deze automatische methode is dat deze zelf gaat zoeken naar de optimale parameterwaarden die het best passen bij de gegeven dataset en overfitting voorkomen door het gebruik van informatie criteria [16]. In volgende subsecties zullen de toegepaste voorspellingsmodellen verder uitgelegd worden met bijhorende cruciale aspecten.

2.1 Stationariteit

Binnen het onderwerp van tijdreeksanalyse/-voorspellingen is stationariteit een zeer belangrijk begrip omdat veel modellen hier assumpties of testen over doen. Een stationaire tijdreeks is een tijdreeks waarvan de eigenschappen zoals gemiddeldes, variantie, covariantie en/of standaardafwijkingen niet afhangen van het moment (tijd) dat de reeks geobserveerd wordt [17]. Dit impliceert dat tijdreeksen met trend en/of seizoensgebondenheid (*seasonality*) niet stationair zijn. Er bestaan echter enkele transformaties om een tijdreeks stationair te maken, nl. differentiëren en logaritmes nemen. Differentiëren [1] zorgt ervoor dat het gemiddelde van de tijdreeks stabiliseert, wat ook betekent dat de trend verdwijnt/vermindert [18]. Het logaritme[2] nemen zorgt ervoor dat de variantie van de tijdreeks gestabiliseerd wordt. Om op een objectieve manier een zicht te krijgen op de trend, seizoensgebondenheid en de stationariteit kan er gebruik gemaakt worden van een unit root test zoals bijvoorbeeld Augmented Dickey-Fuller (ADF) [19]. Dit zijn aspecten die ook verder toegelicht worden in Sectie 3.1.2.

2.2 Trends & Seizoenen

Trends en seizoenen zijn twee componenten die kunnen voorkomen in tijdreeksdata [6]. Een trend geeft aan op welke manier een tijdreeks evolueert/beweegt doorheen de tijd. Zo kan er bijvoorbeeld een negatieve trend geassocieerd worden met een 'dalende' tijdreeks. Seizoenen geven aan dat er bepaalde patronen/gedrag in de data zich herhalen na binnen bepaalde tijdsperiodes [20]. Beide kunnen additief of multiplicatief voorkomen. Additief betekent dat de betreffende component (trend of seizoen) wordt toegevoegd/opgeteld met andere individuele componenten in een tijdsreeks. De trend- en seizoenscomponent kunnen aan de hand van bepaalde functies, zoals "seasonal decompose" in Python, achterhaald worden [21]. Onderstaande formule geeft zowel additieve trend (T) als seizoenen (S) aan, die worden opgeteld met de error (E), wat een component voorstelt die niet verklaard wordt door trend of seizoen. y stelt de tijdreeks voor op tijdstip t.

$$y_t = T_t + S_t + E_t$$

[1] Verschil = observatie(t) - observatie(t-1), waarbij t een bepaalde tijdsperiode voorstelt waarop gegevens verzameld werden.

[2] Transformatie = log(x), waar x de tijdreeksdata voorstelt.

Een additieve trend in een tijdreeks impliceert dat er een lineaire trend aanwezig is in de tijdreeks. Een additieve seizoensgebondenheid betekent dat de frequentie/periode (breedte) en de amplitude (hoogte) van seizoenen hetzelfde blijven.

Multiplicatief betekent echter dat de betreffende component wordt vermenigvuldigd met de ander componenten zoals aangegeven in onderstaande formule. Met een multiplicatieve trend wordt er bedoeld dat een non-lineaire/gebogen trend aanwezig is in de tijdreeks [18]. Een multiplicatieve seizoensgebondenheid betekent dat de frequentie en/of de amplitude van seizoenen stijgt of daalt doorheen de tijdreeks.

$$y_t = T_t * S_t * E_t$$

De trends kunnen ook worden aangepakt door een zogenaamde 'damped' trend toe te voegen aan de vergelijking. Deze techniek zorgt ervoor dat de trend zal uitdoven (verminderen in waarde) totdat deze uiteindelijk geen invloed meer heeft op de voorspellingen [22]. Dit wil zeggen dat de trendcomponent gelijk zal worden aan nul. Dit is nodig omdat Holt [23] besloot dat wanneer er een constante trend werd toegevoegd aan een voorspelling, deze leidde tot overvoorspellingen (te hoog of te laag) op lange termijn.

Seizoenen kunnen in verschillende ingewikkelde vormen voorkomen in tijdreeksen zoals: meerdere seizoensperioden, hoge frequentie seizoenen, non-integer (gehele) seizoenen en duale-kalender effecten op seizoenen [24]. Meerdere seizoensperioden betekent dat er op een tijdreeks zowel wekelijkse, maandelijks als dagelijkse seizoenspatronen voorkomen. Er wordt gesproken van hoge frequentie seizoenen wanneer deze op op tijdreeksen voorkomen met hogere frequentie dan maandelijkse data (wekelijks, dagelijkse, ...) [25]. Deze kunnen vaak zeer complex zijn om te bevatten en vereisen soms transformaties. Non-integere seizoenen komen voor wanneer een seizoensperiode niet bevat kan worden in een geheel getal, bijvoorbeeld 52.179 weken [24]. Dan zijn er ook nog duale-kalender effecten op seizoenen in tijdreeksdata. Deze effecten zijn te zien in de tijdreeks door de verschillen in kalender-eigenschappen waardoor feestdagen en culturele/religieuze activiteiten verschillen [26].

2.3 Statistische methodes

De statistische voorspellingsmethodes baseren zich op de geschiedenis van de afhankelijke variabele om voorspellingen te produceren [27]. Deze modellen worden beschreven aan de hand van wiskundige formules. Ze zijn eerder eenvoudig te interpreteren en hebben doorgaans een goede accuraatheid [28]. Deze methoden kunnen trend en seizoenen verwerken, maar het bepalen van non-lineaire functies is beperkt tot onmogelijk.

2.3.1 Naïve Seasonal & Drift De naïeve methode wordt gekenmerkt door een seizoen en een trend component. Deze twee componenten worden wel op een

eenvoudige manier voorgesteld. In het onderzoek werd er gebruik gemaakt van *naïve seasonal* met drift. Een gewoon naïef model neemt voor voorspellingen, telkens de laatste observatie als voorspelde waarde [18]. In een *naïve seasonal* gaat dit niet de laatste waarde zijn, maar de x-aantal (parameter K) laatste waarden. *Een naïve seasonal* met K=1 geeft dus een voorspelling die gelijk is aan de laatste waarde uit de training set. De toevoeging van een drift geeft dan weer de trend aan van de voorspellingen. Deze drift is gelijk aan de gemiddelde verandering over alle verleden data. Dit wordt getoond in onderstaande formule waarbij de voorspelde waarde op voorspellingshorizon h gelijk is aan de waarde k tijdsperiode (in het verleden) plus de trend uit de trainingsdata met trainingslengte T [29].

$$\hat{y}_{t+h} = y_{t-k} + h\left(\frac{y_t - y_1}{T-1}\right)$$

2.3.2 Autoregressive Integrated Moving Average Autoregressive Integrated Moving Average (ARIMA) heeft vooral focus op autocorrelaties tussen datapunten en probeert aan de hand van autocorrelaties de data te beschrijven [30]. Dit wil zeggen dat het model aan de hand van de historische data en de relaties tussen punten, voorspellingen zal proberen maken. ARIMA bestaat uit Autoregressief (AR) Integrated (I) Moving Average (MA) [30]. Het autoregressieve (AR(p)) gedeelte gaat de *variable of interest* voorspellen aan de hand van een lineaire combinatie van verleden waarden (p) van de verklarende variabele, in geval van 'Sales Forecasting' zijn dit dan de verkopen. De orde van parameter p bepaalt hoeveel verleden waarden er mee opgenomen worden in het model, dus p=1 zal enkel de eerste verleden waarde meetellen [3]. Integrated (I) gaat ervoor zorgen dat de tijdreeks stationair wordt door het meten van de tijdsverschillen (differences) van observaties op een verschillend tijdsmoment. Concreet gaat het model dus datapunten differentiëren, wat betekent dat het de directe voorganger van datapunten zal aftrekken om de data stationair te maken ($\hat{y}_t - y_{t-1} = (y_{t-1} - y_{t-2}) + \mu$) [18, 31] . Het Moving Average (MA) gedeelte zorgt ervoor dat er rekening gehouden wordt met de afhankelijkheid tussen observaties en de error [30], wat beschreven wordt als parameter q die de grootte van de *moving average window* aangeeft. Het gaat dus ervoor zorgen dat het model zichzelf aanpast op basis van de vorige voorspellingen en de richting van de error door het toevoegen van een aparte constante (αe_{t-1}, met e_{t-1} gelijk aan $y_{t-1} - \hat{y}_{t-1}$). Autocorrelatie functie (ACF) kan helpen in het bepalen van de lineaire afhankelijkheid tussen observaties die p waarden van elkaar liggen. De partiële autocorrelatie functie (PACF) kan ook helpen bepalen hoeveel autoregressieve termen p nodig zijn.

2.3.3 Exponential Smoothing & ETS *Exponential smoothing* (ES) methodes voorspellen de toekomst door middel van gewogen gemiddeldes te nemen

[3] $\hat{y}_t = y_{t-1} + \mu$, waarbij μ de constante term is die gelijk is aan de lange-termijn drift/trend, die niet verklaard wordt door voorgaande observaties.

van verleden observaties, waarbij nieuwe observaties meer gewicht/invloed (α) krijgen dan oudere observaties. De originele ES methode laat enkel toe voor punt-voorspellingen (één voorspelling in toekomst).

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + ...$$

In de meest recente literatuur wordt er bijna enkel nog gesproken over *Holt-Winters Exponential Smoothing* (HWES) [18, 20, 32, 33]. Dit is een extensie van de originele *Exponential Smoothing* (ES), waarbij er rekening gehouden wordt met de effecten van trends en seizoensgebondenheid. Niveau, trends en seizoenen worden gezien als aparte componenten.

Verder zijn er ook de '*Error, Trend, and Seasonality*' (ETS) modellen [18, 34, 35]. Via een *state space framework*, met assumpties over de verdeling van de errors, worden alle verschillende ES modellen geclassificeerd volgens error, trend en seizoen (niet aanwezig, additief of multiplicatief), wat de berekening mogelijk maakt van: voorspellingsintervallen (en niet enkel punt-voorspellingen), aannemelijkheid (*probability*) en model selectie criteria [34]. Hierdoor gaat het ook het beste ES model kunnen bepalen voor de gekozen data door elk mogelijk model te evalueren.

2.3.4 Theta Volgens Hyndman en Billah [36] is de Theta methode gelijkaardig aan de *Simple Exponential Smoothing* (SES) methode met drift en zouden dit ook equivalente voorspellingen moeten opleveren. Zij omschrijven de Theta methode ook als een SES met drift, waarbij de SES zelfs betere voorspellingen kan maken wanneer de parameters geoptimaliseerd worden met behulp van '*maximum likelihood approach*'. Dit is een methode in de statistiek die ervoor zorgt dat er voor parameters een optimale waarde gekozen wordt en waarbij de aannemelijksheidsfunctie (*probability function*) wordt geoptimaliseerd [37]. De belangrijkste parameter bij de Theta methode is de theta (θ) parameter, die de richtingscoëfficiënt/helling ("slope") van de trend bepaalt en kan versterken of verminderen [38]. Voor meer geaggregeerde data zoals op maandelijks of jaarlijks niveau, wordt er eerder een grotere waarde verwacht voor theta. Omgekeerd wordt er voor minder geaggregeerde data een lagere waarde voor theta verwacht omdat seizoensgebondenheid hier vaker sterker aanwezig is [38]. Het model voorspelt twee richtingen/lijnen met behulp van ES en combineert deze daarna om tot een finale voorspelling te komen.

2.3.5 (T)BATS De methodes BATS en TBATS zijn ontworpen om complexe seizoensgebondenheid te modelleren in voorspellingen [24]. Dit kan gaan over meerdere seizoensperioden in een dataset, hoge frequentie seizoenen, seizoenen die niet omschreven kunnen worden als een getal en duale-kalender effecten zoals besproken in Sectie 2.2. In het geval van retail zal het eerder betrekking hebben op de meerdere seizoensperioden zoals de maandelijkse seizoenen maar dan ook nog seizoenen die per jaar terugkomen. Ook op de hoge frequentie van seizoenen kan dit model inspelen. De verschillende componenten van (T)BATS zijn:

(Trigonometric ES), Box-Cox transformatie, ARMA residuals, Trend en Seizoen. De trigonometrische component laat toe om non-integer en hoge seizoensfrequenties op te nemen in het model. Dit wilt zeggen dat seizoenen niet binnen een standaard periode/gehele frequentie (dagen, weken, maanden,...) moeten liggen en dat complexe seizoenen beter behandeld kunnen worden [24]. De Box-Cox transformatie [39] zorgt ervoor dat een niet-normaal verdeling omgezet wordt naar een normaal verdeling, om dan lineaire bewerkingen toe te laten op non-lineaire data. Het verschil tussen BATS en TBATS is dus enkel de trigonometric component die deels bepaalt hoe seizoenen behandelt worden. BATS doet dit op een meer traditionele manier waar enkel gehele getallen als periodes per seizoen genomen kan worden en veel parameters aangepast moeten worden om complexe seizoenen te behandelen [24].

2.3.6 Prophet De Facebook Prophet methode is ontstaan met de gedachten van twee grote problemen tijdens het voorspellen van tijdreeksen. Ten eerste is het moeilijk om automatische voorspellingsmodellen aan te passen en deze zijn ook vaak te inflexibel in een bedrijfscontext [40]. Ten tweede is het vaak zo dat de analist verantwoordelijk voor het uitvoeren van de voorspellingen veel kennis heeft van het product of de dienst, maar te weinig training heeft in tijdreeksvoorspellingen. Dus als hoofdgedachte wil de methode goede voorspellingen kunnen maken, die voor een groot aantal mensen beschikbaar en te gebruiken zijn. Dit gaat het proberen te bereiken door veel stappen/parameters zelf te bepalen om zo gebruikersfouten te voorkomen. De methode zelf is een ontleedbare tijdsreeks [41] die hoofdzakelijk bestaat uit: trend, seizoensgebondenheid en feestdagen/gebeurtenissen [40]. In onderstaande formule wordt Prophet simpel omschreven als een non-lineair regressiemodel met: g(t) als beschrijving van de lineaire trend ('groei term'), s(t) die verschillende seizoenspatronen beschrijft, h(t) die feestdageffecten omvat en ε_t die de errorterm is [18]. Een voordeel van deze methode is ook zijn eenvoudige interpreteerbaarheid, ten opzichte van complexere modellen, in de verdeling van de regressie.

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t$$

2.4 Machine Learning methodes

Machine Learnig (ML) methodes laten toe om direct en autonoom te leren van de data, waardoor er meer potentieel is voor accuratere voorspellingen, maar er ook een gevaar op overfitting is. Deze modellen zijn flexibeler om zelf verborgen en non-lineaire patronen in de data te identificeren waardoor de aanpasbaarheid verhoogt en er een robuuster systeem is dan de voorgaande methodes, die vast leren op basis van hun voorafbepaalde formules [7]. In volgende subsecties worden de toegepaste ML methodes verder uitgelegd.

2.4.1 NeuralProphet De NeuralProphet methode is afgeleid van de Prophet methode en probeert dezelfde waarden aan te houden om interpreteerbare modellen te maken die toch accuraat zijn [42]. Het breidt Prophet uit door neurale

netwerk modules toe te voegen aan de gekende tijdreekscomponenten om ook non-lineaire dynamieken te kunnen bevatten [42]. Ook kan NeuralProphet altijd aangepast worden aan nieuwe deep learning innovaties [42]. De uiteindelijke componenten zijn: trend (g(t)), seizoenen (s(t)), feestdagen/gebeurtenissen (h(t)), regressie van externe variabelen die gekend zijn in de toekomst (F(t)), autoregressie op verleden observaties (A(t)) en regressie voor externe variabelen die gekend zijn uit het verleden (L(t)). Er worden neurale netwerken gebruikt om componenten te optimaliseren en voornamelijk bij het autoregressieve gedeelte (AR-net) [42] waar Feed-Forward Neural Networks voor gebruikt worden.

$$y_t = g(t) + s(t) + h(t) + F(t) + A(t) + L(t)$$

2.4.2 Neurale Netwerken Neurale netwerken (NN), ookwel Artificiële neurale netwerken (ANN) of recurrent neural networks (RNN) zijn verschillende types voorspellingsmodellen, die proberen om het denkproces van mensen te modelleren om zo patronen/relaties in de data te ontdekken [43, 44]. Dit zijn non-lineaire modellen, waardoor deze soms moeilijk zijn om te begrijpen of uit te leggen, daarom worden ze soms ook wel 'black-box' modellen genoemd. ANNs komen voor in de vorm van: Feed-Forward Neural Networks (FFNN) waarbij telkens een waarde voorwaarts wordt doorgegeven naar de neuronen en in de vorm van RNNs die ook 'feedback loops' ondergaan waarbij verleden output ook in rekening worden gebracht, zie appendix. FFNN hebben geen geheugen en weten enkel de huidige input en trainingscomponent. RNN hebben wel geheugen en nemen buiten de input ook de verleden output mee in rekening. Nog een eigenschap van RNNs is dat deze een geheugen hebben en dus de 'state' kan onthouden van voorgaande neuronen [45] waardoor deze geschikt is voor tijdreeksvoorspellingen. Het model gaat dus niet enkel rekening houden met nieuwe input maar ook met voorgaande output [46]. De manier waarop het model zaken onthoudt kan veranderen, namelijk via Long Short-Term Memory (LSTM) en Gated Recurrent Unit (GRU).

LSTMs [47] hebben geheugen eenheden, voorgesteld als poorten, voor neurale netwerken. Ze hebben drie poorten die de inhoud van het geheugen managen. Deze poorten kunnen voorgesteld worden als logaritmische functies van gewogen sommen, waar de gewichten geleerd kunnen worden door backpropagation [46]. Er is een vergeet-poort, die conditioneel beslist welke informatie weg te gooien. Er is de input-poort, die beslist welke waarden van de input gebruikt worden om geheugen te vernieuwen. En dan is er nog de output-poort, die bepaalt welke output het verder verstuurd op basis van de input en het geheugen. Het kan met andere woorden leren wat het moet leren, onthouden wat het moet onthouden en doorgeven wat het moet doorgeven. GRUs [48] zijn eigenlijk vereenvoudigde LSTMs en vervullen dezelfde rol in het neurale netwerk. Het enige verschil is dat GRU maar twee poorten heeft. Voor een verdieping in de vergelijking van LSTMs en GRUs zie de paper van Chung et al. [49].

2.4.3 Random Forest Random Forests (RF) [50] zijn 'Ensemble' ML algoritmes, die voorspellingen genereren door eerst een groot aantal willekeurige beslissingsbomen te maken en daarna deze resultaten samen te voegen. Het wordt benoemd als 'Ensemble' omdat het eigenlijk voorspellingen uit meerderen modellen (beslissingsbomen) gaat verwerken, die dan moeten leiden tot betere voorspellingen dan enkel één model zou doen. Het gebruikt de 'bagging' techniek, waarbij het de data willekeurig gaat opdelen in subsets, die daarna teruggeplaatst worden in de populatie [50, 51], waarop het dan beslissingsbomen op toepast.

2.4.4 Gradient Boosting Een laatste ML model dat getest werd is Gradient Boosting. Het is een ML techniek die voorspellingen maakt in de vorm van zwakke voorspellingsmodellen, zoals beslissingsbomen [52]. Het idee is om herhaaldelijk de patronen in de residuals/fouten te gebruiken om het model betere voorspellingen te laten maken. Dit gebeurd op een additieve manier, dus nieuwe leermodellen worden steeds toegevoegd waar $f_t(X)$ het volledige model is en g(X) de nieuwe beslissingsboom:

$$f_t(X) = f_{f-1}(X) + g(X)$$

2.5 Accuracy Measures

Om uiteindelijk de verschillende modellen met elkaar te vergelijken, zijn er bepaalde nauwkeurigheidsmaatstaven (*accuracy measures*) nodig om de prestaties van voorspellingsmodellen aan te geven. Deze maatstaven geven noodzakelijke en beslissende feedback om al dan niet voor een bepaalde voorspellingsmethode te kiezen of om aanpassingen te doen [5, 53]. In gevallen van machine learning modellen worden deze ook benoemd in de loss-functie, die een model probeert te minimaliseren. Het is afgeraden om de prestatie van een voorspellingsmodel enkel te baseren op één maatstaf, want prestaties kunnen wijzigen naargelang de maatstaf die gekozen wordt [53, 54]. Er is in de literatuur geen eenzelfde accuracy maatstaf te vinden die voor alle onderzoeken als doorslaggevend gezien wordt [5]. Er zijn wel al onderzoeken gedaan naar het opstellen van frameworks zoals het Multi-Criteria Decision Analysis (MCDA) maar deze worden niet vaak toegepast in bedrijfsomgevingen [13, 53]. In deze frameworks wordt er een robuuste analyse opgesteld die verschillende voorspellingsmethodes zou kunnen vergelijken.

Er kunnen drie prestatiecriteria onderscheiden worden. Zo zijn er de *goodness-of-fit*, *biasedness*, en de *correct sign* [55, 56]. De *goodness-of-fit* bekijkt hoe goed de voorspellingen de echte observaties benaderen. Dit kan bepaald worden met behulp van *squared errors* of *absolute errors*. Volgens Hyndman Athanasopoulos [18] zijn de MAE en RMSE de meest toegepaste in realiteit. Over deze maatstaven zal in de volgende subsecties verder uitgeweid worden. Biasedness gaat eerder over de systematische over- of onderschatting van voorspellingen [13], wat bijvoorbeeld gemeten kan worden met Proportion of Tests Supporting Unbiasedness (PTSU). Als laatste is er nog de correct sign, die aangeeft of een

voorspelling consistent is met het teken van de echte observaties [55]. Dus wanneer een voorspelling stijgt of daalt zou de echte observaties dit ook moeten doen. Een maatstaf die hier bijvoorbeeld voor gebruikt kan worden is de *Proportion of Correct Direction Change Predictions* (PCDCP). Voor verkoopdata, die seizoensgebondenheid bevat, zal dit minder belangrijk zijn want de richting van de verkopen zal vaak veranderen en irrelevant zijn als maatstaf van een voorspellingsmethode.

In de volgende subsecties wordt er een opsomming gemaakt van de algemeen toegepaste maatstaven, die gebruikt kunnen worden voor voorspellingaccuraatheid. Deze werden opgedeeld in vier categorieën zoals Hyndman en Koehler [57] hebben voorgesteld: *scale-dependent*, percentage, relatief, en geschaald. In dit onderzoek worden enkel de eerste twee gebruikt omdat deze vaker worden toegepast in realiteit en deze zijn eenvoudiger te interpreteren [58]. Om de modellen te vergelijken werd er in dit onderzoek gebruik gemaakt van volgende maatstaven: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) en de Root Mean Squared Error (RMSE).

2.5.1 Absolute Error Zoals hierboven al vermeld, kan een *goodness-of-fit* getest worden aan de hand van absolute errors. De meest toegepaste voorbeelden van deze maatstaven zijn Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), en Mean Absolute Deviation (MAD) [13, 59, 3, 57, 60, 18]. Een nadeel van de MAE en MAD is dat deze schaalafhankelijk zijn, deze kan dus niet gebruikt worden om verschillende datasets met elkaar te vergelijken [61]. In tegenstelling tot percentage errors want deze zijn wel schaalonafhankelijk en deze kunnen dus wel gebruikt worden als accuracy measure over verschillende datasets. Het nadeel van percentage errors is dan weer wel dat deze een andere straf geven aan positieve errors dan aan negatieve errors. [61, 57]. Nog nadelen van deze percentage errors zijn dat deze oneindig zijn wanneer een echte observatie nul is en dat de distributie scheef is wanneer geobserveerde waardes dichtbij nul zijn [58]. Voor het onderzoek werd er voor de absolute errors MAE en MAPE gekozen.

- MAE: $\frac{1}{n}\sum_{t=1}^{n}|e_t|$
- MAPE: $\frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{e_t}{y_t}\right|$

In bovenstaande formules worden deze weergegeven, waarbij e_t het verschil aangeeft tussen de geobserveerde waarde en de voorspelde waarde op tijdstip t. De MAE geeft dus aan hoeveel eenheden je voorspelling gemiddeld verschilt met de geobserveerde waarden. De MAPE daarentegen met het percentage verschil van de voorspelling in relatie met de geobserveerde waarde.

2.5.2 Squared errors Dan zijn er nog de squared errors of de gekwadrateerde fouten waarbij er een transformatie wordt gedaan. Er wordt dan in de plaats van de absolute waarde, gebruik gemaakt van de tweedemacht van de error. Dit heeft

als eigenschap dat de maatstaf gevoeliger is voor grote errors (groter dan 1 of -1) dan absolute errors [13]. Enkele voorbeelden van deze *squared errors* zijn: Mean Squared Error (MSE), Mean Squared Percentage Error (MSPE) en Root Mean Squared Error (RMSE). In ons onderzoek zal er gebruik gemaakt worden van de RMSE. Zoals bij de absolute errors betekent e_t het verschil tussen de geobserveerde waarde en de voorspelde waarde op tijdstip t.

– RMSE: $\sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2}$

2.5.3 Statistische vergelijkingsmethode Buiten de eerder conventionele *accuracy measures* om de accuraatheid van de voorspellingen te meten, zijn er ook nog andere methodes om modellen met elkaar te vergelijken. Er zijn mogelijkheden tot F- en t-testen om methodes met elkaar te vergelijken. In dit onderzoek werd er gebruik gemaakt van verschillende t-testen om modellen met elkaar te vergelijken. Uiteindelijk kon er dan een significant verschil of significante best/slecht presterend model gekozen worden. Een voorbeeld van deze gepaarde t-test vergelijking tussen voorspellingsmethodes, is te vinden in het werk van Kandananond [62]. Een t-test is een statistische test, die de gemiddeldes van twee groepen vergelijkt door te bekijken of de lijst van observaties tot dezelfde verdeling horen of niet [63]. Er zijn twee types t-testen: onafhankelijke en gepaarde. De onafhankelijke wordt toegepast wanneer de twee groepen observaties onafhankelijk zijn van elkaar. De gepaarde wordt toegepast wanneer de twee groepen wel afhankelijk zijn [63].

3 Methodologie

In volgende subsecties wordt er dieper ingegaan op de voorafgaande stappen die leidden tot de resultaten. Eerst wordt de data besproken en welke voorbereidende stappen er ondernomen werden om tot bruikbare data te komen. Ook wordt de analyse van de tijdreeks besproken, die ons inzichten geeft over enkele eigenschappen van de data. Een volgende aspect zijn dan de accuracy measures, die meer informatie geven over de prestaties van verschillende voorspellingsmodellen. Ook zal er per voorspellingsmodel besproken worden met welke libraries deze toegepast werden en welke parameters er aangepast werden.

3.1 Data

Er werd in dit onderzoek gebruik gemaakt van twee verschillende retail databronnen, waarvan beide betrekking hebben op kledingdata. Vooraleer er met de data voorspellingen gedaan konden worden, werd deze grondig geanalyseerd op bijvoorbeeld trends, seizoenen, autocorrelaties, uitschieters, etc. Elke vorm van voorbereiding of analyse werd uitgevoerd op verschillende aggregatieniveaus van de data. Zo werd er op wekelijkse en maandelijkse data geanalyseerd, maar ook nog per winkeltype. Het winkeltype kan bijvoorbeeld online of periferie zijn. Een

dataset bevat data over een periode van zeven jaar en de andere dataset bevat een periode van vijf jaar. In beide gevallen gaat het hier over verkoopsdata, die opgeslagen werd als kassalijnen. Een overzicht van de data kan gevonden worden in Tabel 1. Voor elke aangegeven dataset (om het even welk aggregatieniveau) werd er ook een dataset gemaakt waarvan de Covid-19 data uit lockdownmaanden maart en april (2020) aangepast werden naar het rollende gemiddelde zoals aangegeven in de volgende subsectie 3.1.1 Voorbereiding. Voor dataset drie en vier werden ook de maanden november en december (2020) vervangen aangezien deze zwaar door de coronacrisis werden beïnvloed (uitzonderlijke daling aantal verkopen).

Data	Dataset1	Dataset1a	Dataset2	Dataset2a	Dataset3	Dataset4
Granulariteit	Wekelijks	Wekelijks + Winkeltype	Maandelijks	Maandelijks + Winkeltype	Wekelijks	Maandelijks
Start Datum	2014-01-05	2014-01-05	2014-01-31	2014-01-31	2016-01-03	2016-01-31
Eind Datum	2021-10-10	2021-10-10	2021-10-31	2021-10-31	2021-10-10	2021-10-31
Totale observaties	406	406 per winkeltype	94	94 per winkeltype	302	70
Train Observaties	354	354	82	82	250	58
Test Observaties	52	52	12	12	52	12
Seizoenen	4 - 9 - 13	(4) - (6) - 9 -13	3 - 6 -9 - 12	(3) - 6 - 9 - 12	4 - 9 - 13	6 - 12 - 18

Tabel 1: Dataset 1(a) en 2(a) zijn afkomstig van dezelfde databron, idem voor dataset 3 en 4. Seizoenen geven aan op welke *lags* er seizoenen voorkomen, (getallen) tussen haakjes zijn niet overal als seizoen aangeduid bij de verschillende winkeltypes.

3.1.1 Voorbereiding Tijdens de voorbereiding van de data werden er visualisaties gemaakt om een beeld te krijgen op de verdeling en de inhoud van de beschikbare tijdreeksdata. Een voorbeeld hiervan zijn visualisaties met rollende gemiddeldes en met rollende standaardafwijking, zie Sectie 7.2. Deze twee maatstaven gaven een eerste zicht op de seizoensgebondenheid en trends in de data. Aangezien veel voorspellingsmethodes afhangen en beïnvloed worden door seizoenen en trends, waren dit al enkele belangrijke inzichten die achteraf verwerkt konden worden in de modellen.

Uit de grafieken, getoond in Sectie 7.2, werden snel enkele belangrijke zaken duidelijk. Zo was er het effect van Covid-19, en bijhorende lockdowns/beperkingen, op de verkopen goed te zien in de maanden maart en april van 2020. De daling in verkopen voor desbetreffende maanden zijn eenvoudig te verklaren door de start en opkomst van Covid-19 in België, die door een lockdown[4] ongetwijfeld invloed heeft gehad op het aankoopgedrag van consumenten. Een probleem dat zich stelde voor de laatste observatie (wekelijks als maandelijkse data) van de datasets was dat deze onvolledige data bevatte door onderbroken dataverzameling. Verder is er voor de periode van november (2020) tot april (2021) te zien dat het aantal verkopen dichtbij nul zou liggen voor zowel off- als online verkopen. Dit wees op een probleem bij de dataverzameling, wat een sterke invloed kon

[4] https://www.hln.be/binnenland/overzicht-van-de-lockdown-op-18-maart-tot-de-recentste-coronamaatregelen-dit-is-het-parcours-dat-belgie-tot-nu-heeft-doorlopen af470e8f/

hebben op de voorspelbaarheid van de tijdreeks. Daardoor werd er ook gekozen om deze uitzonderlijke datapunten om te zetten naar meer representatieve datapunten. Voor de desbetreffende maanden werden de waarden vervangen door het rollend gemiddelde over een periode van 32 weken (wekelijkse data) en negen maanden (maandelijkse data). Deze perioden werden gekozen aangezien deze overeenkomen met de seizoenspatronen die ontdekt werden. Ook doordat het rollend gemiddelde berekend werd vanuit een centrum-opzicht (x-aantal voor en x-aantal na gekozen datapunt), moest er rekening gehouden worden met de periode van zes maanden waarvoor geen data beschikbaar was.

3.1.2 Analyse Verder in het onderzoek werden er diepere analyses uitgevoerd om een gedetailleerder zicht te krijgen op de data. Zo werd er een Augmented Dickey-Fuller (ADF) test uitgevoerd op de data. Deze test bekijkt of de tijdreeks stationair is [19], wat een vereiste is voor verschillende tijdreeksmodellen. Uit de test volgde dat de wekelijkse datasets wel stationair en de maandelijkse data niet stationair zijn. Dit zou kunnen komen doordat wekelijkse data meer cyclisch gedrag heeft dan maandelijkse data. Stationairiteit kon in verschillende voorspellingsmethodes meegegeven worden als parameter. Om nog extra kennis te verkrijgen over seizoenen en trends werden er nog boxplots opgesteld, zie Figuur 4 & 5 in de appendix. Deze tonen aan dat de maanden februari en maart het minder goed doen, maar de maanden januari, april en juli doen het beduidend beter. Deze laatste observatie zet zich door in de maandelijkse data. In de maandelijkse data is een negatieve trend te zien, die in op het andere aggregatieniveau minder aanwezig was. Als extra test voor seizoensgebondenheid werd er de functie 'check-seasonality' uit het 'Darts' package [29] gebruikt. Deze test gaat eigenlijk net zoals de auto-correlation function (ACF) test kijken op welke *lags* er correlatie is en bekijkt dit dan op een bepaald significantieniveau, nl. vijf procent.

Vervolgens werden ook manueel de autocorrelaties (ACF) en partiële autocorrelaties (PACF) geanalyseerd. In tegenstelling tot gewone correlaties waarbij de relatie tussen twee of meer variabelen worden bekeken, wordt er bij autocorrelaties gekeken naar de correlatie tussen twee observaties op verschillende punten in een tijdreeks [64]. Via de ACF kunnen er onderliggende patronen en eigenschappen van de tijdreeks achterhaald worden. De waarden van de autocorrelaties kunnen ook wijzigen naargelang het aggregatieniveau van de data. Aan de hand van autocorrelaties kunnen verschillende aspecten zoals trends, seizoensgebondenheid en stationariteit achterhaald worden. Het verschil tussen ACF en PACF is dat de PACF enkel de correlatie tussen twee observaties toont, die eerdere observaties niet verklaarden [64]. Als extra validatie-stap om de verschillende trends en seizoenen per dataset te bepalen, werden er lineaire regressies opgesteld. De maanden, jaren en een sequentie nummer werden meegegeven aan de regressie. Deze kregen door de regressie een gewicht toegewezen, waaruit dan afgeleid kon worden in welke maten er seizoensgebondenheid aanwezig was voor de maanden. Ook kon er uit het gewicht van de sequenties afgeleid worden in welke mate er een trend aanwezig was in de data. Dit werd enkel toegepast op de

maandelijkse aggregatieniveaus. Gedetailleerde waardes van deze regressie zijn te vinden in Tabel 9 met bijhorende visualisaties in Figuur 4 & 5. Uit de analyses is gebleken dat wekelijkse data uit beide datasets sterk gecorreleerd zijn met *lags* 4, 9, 13,... Voor de wekelijkse data per winkeltype varieerden 'Online' en 'Shop' enkel omdat zij op *lag* 4 geen seizoensgebondenheid hadden maar wel op 6 of 9. De maandelijkse data is eerder gecorreleerd met *lags* 3, 6, 9,... Ook voor de maandelijkse data waren er enkele winkeltypes die op andere lags seizoensgebondenheid vertoonde. Dit kunnen belangrijke gegevens zijn wanneer er in de voorspellingsmodellen seizoenen aangegeven moeten worden. In Tabel 1 worden de verschillende seizoenen ook aangegeven per dataset. Ook wordt aangegeven dat er enkele seizoenen anders plaatsvonden bij verschillende winkeltypes.

3.2 Accuracy Measures

Om verschillende voorspellingsmethodes met elkaar te vergelijken, werd er gebruik gemaakt van de accuracy measures besproken in Sectie 2.5. Deze *accuracy metrics* werden telkens berekend op een test set met een voorspellingshorizon van één jaar. Hierbij konden we de MAE en de RMSE enkel vergelijken bij methodes die op dezelfde dataset (dezelfde schaal) getraind waren. Voor vergelijkingen met andere datasets werd de MAPE gebruikt. Twee factoren die nog werden toegevoegd aan de analyse als extra maatstaf waren: de duur en het geheugengebruik [5]. De 'Duur' geeft aan hoeveel seconden de voorspellingsmethode nodig had om het model te trainen plus de tijd om voorspellingen van één jaar te maken. Het geheugengebruik, uitgedrukt in bytes, werd bekeken (via tracemalloc) door het voor zowel training als voorspellingen te berekenen. Bij de bespreking van de resultaten, werden enkel bij de percentages en duur decimalen meegegeven, bij de andere metrieken werden decimale getallen afgerond naar het dichtstbijzijnde geheel getal.

Verder werden er ook vergelijkingen gemaakt via de MAPE tussen de verschillende soorten datasets. Voor de datasets die opgedeeld werden per winkeltype, kon de Theta methode niet gebruikt worden voor de originele data omdat deze negatieve waarden bevatten en dus niet bruikbaar waren voor het theta model. Er werd ook bekeken of door Covid-19 beïnvloede datapunten op hun beurt een invloed hadden op de voorspellingsaccuraatheid. Een ander aspect was dan om te kijken naar de verschillende aggregatieniveau's. Er werd gekeken naar het aggregatieniveau per winkeltype en of deze beter presteren met verschillende voorspellingsmethodes dan een algemeen model voor alle winkeltypes samen. Deze vergelijkingen werden bekomen door een t-test uit te voeren op de percentage errors ($e_t = \frac{y_t - \hat{y}_t}{y_t} * 100$) van de verschillende voorspellingsmodellen.

Om verschillende modellen met de benchmark te vergelijken, werd er gekozen voor gepaarde t-testen, omdat er gewerkt werd in experimentele omgevingen

[5] Deze kunnen verschillen per uitvoering van het model, maar ze geven een indicatie aan rond welke waarde deze schommelen. Sommige modellen werden in Pycharm uitgevoerd en sommigen in Jupyter Notebook met Anaconda omgeving. Dit kan een invloed hebben op deze twee maatstaven.

waarbij verschillende voorspellingsmethodes werden toegepast op identieke data. Deze test heeft wel assumpties zoals normaliteit, gelijke variantie en onafhankelijkheid. Omdat er voor sommige voorspellingen enkel een zeer kleine verzameling voorspelde datapunten waren (onder 20 observaties), is er gekozen voor een alternatieve t-test. Er werd gekozen voor de non-parametric t-test 'Wilcoxon Signed-Rank' die geen assumpties heeft over de onderliggende verdelingen zoals de assumptie van normaalverdeling [15]. In sommige werken [65] wordt er aangegeven dat er een minimum grootte van 15 tot 20 datapunten moet zijn per verzameling. In ons onderzoek worden deels voorspellingen gedaan met enkel 12 voorspelde datapunten, hier nemen we de assumptie aan dat dit voldoende is om toch betrouwbare testen uit te voeren. De nulhypothese werd verworpen op vijf procent significantie niveau, wat betekent dat er een significant verschil is tussen twee verdelingen. Eenmaal er een significant verschil gevonden werd in de verdelingen van de errors, werden de gemiddeldes van de absolute waardes van de errors met elkaar vergeleken om te kunnen bepalen welk model beter of slechter presteerde.

3.3 Model

In onderstaande subsecties wordt telkens een korte beschrijving gegeven over de verschillende toegepaste voorspellingsmodellen. Deze beschrijvingen bevatten uitleg over de Python packages waarin deze modellen terug te vinden zijn alsook welke hyperparameters er aangepast werden. Alle gebruikte packages zijn terug te vinden in PYPI (Python Package Index) [66]. Voor methodes waarbij enkel één of twee hyperparameters gekozen konden worden en deze invloed hadden op de resultaten, werd er een simpele optimalisatie gezocht van deze parameters om de MAPE, MAE en RMSE te minimaliseren. Deze optimalisatie zocht voor verschillende waarden van de parameters naar de beste accuracy measures. Voor dezelfde voorspellingsmethodes uit verschillende packages, werden identieke waarden toegekend voor de hyperparameters. In Tabel 2 worden de verschillende voorspellingsmethodes weergegeven met bijhorende: aangepaste hyperparameters, packages en het al dan niet automatisch schatten van (hyper)parameters. De kolom automatisch geeft met ja (J) of nee (N) aan of het model volledig automatisch zijn optimale parameters kan bepalen. Na het uitvoeren van de eerste testen met alle verschillende voorspellingsmethodes, werd duidelijk dat verschillende voorspellingsmethodes identieke voorspellingen produceerde (aangezien zij gebaseerd waren op eenzelfde bronmodel). Daarom werd in de daaropvolgende testen abstractie gemaakt van de packages van waaruit deze modellen werden toegepast.

3.3.1 Naïve Om een zo bruikbaar mogelijke voorspelling te verkrijgen werd er gekozen voor het gecombineerd model tussen de *naïve seasonal* en de *naïve drift*. Hierbij werden dus seizoenen en een trend gecombineerd op een naïeve/simpele manier. Voor de parameters was het hier enkel nodig om de seizoenen aan te geven aan de hand van tijdsintervallen. Deze parameter met waarde vier betekende dus dat er elke vier observaties een soort van 'seizoensgebondenheid'

Methode	Hyperparameters	Packages	Automatisch
Naïve	Seizoenen (SP)	Darts, SKTime	N
AutoArima/StatsFAA	SP	Darts, SKTime	J
Exponential Smoothing	SP, Trend & Seizoen (+ of X)	Darts, SKTime	J (ETS)
Theta	Theta	Darts	N
(T)BATS	SP	Darts, SKTime	J
Prophet	Wekelijkse/Dagelijkse Seizoenen	Darts, Prophet	N
NeuralProphet	Wekelijkse/Dagelijkse Seizoenen	NeuralProphet	N
RNN	Model (RNN, LSTM, GRU), Epochs, input_chunk_length	Darts, Keras	N
Random Forest	# Beslissingsbomen	Darts, SKLearn	N
Gradient Boosting	# Beslissingsbomen	Darts, XGBoost	N

Tabel 2: Verschillende voorspellingsmethodes met bijhorende uitleg over: de aangepaste hyperparameters, de packages waaruit ze afkomstig zijn en de mogelijkheid op het automatisch laten schatten van (hyper)parameters

(terugkerend patroon) voorkwam, waardoor het model hier rekening mee ging houden bij het voorspellen. De drift/trend werd zelf bepaald door het model op basis van de verleden data en hier werden geen parameters aan toegevoegd. Deze voorspellingsmethode werd uitgevoerd in de Darts library [29]. Dit model werd gekozen als benchmark waarmee de andere modellen zich gingen vergelijken met de gepaarde t-test.

3.3.2 AutoARIMA Het AutoARIMA model is terug te vinden in verschillende libraries zoals: Darts [29] en SKTime [67]. Beiden libraries zijn gebaseerd op de autoArima uit pmdArima [68]. De twee libraries, Darts en SKTime, zullen met elkaar vergeleken worden om te zien of er een beter presteert dan de andere. Voor de parameters zullen voor beide libraries dezelfde worden toegepast. Er werd aan beide modellen een seizoensperiode meegegeven in relatie met de gegeven dataset. Deze seizoensperioden werden ook aangepast naar de seizoenen die gevonden werden tijdens de analyse van de data om zo tot een optimaal model te komen. Voor de andere parameters laten we het model zelf schatten welke de beste resultaten geeft. Een alternatief op de normale autoArima van pmdArima is de 'snellere' StatsForecastAutoArima van SKTime. Deze methode werd ook uitgetest.

3.3.3 Exponential Smoothing & AutoETS Bij het opstellen van de ES modellen werd er rekening gehouden met verschillende parameters. Zo werden de trend en seizoenen uitgetest met additieve en multiplicatieve eigenschappen. Ook werden deze getest per verschillende seizoensperioden zoals aangegeven in sectie Sectie 3.1. De ES modellen werden toegepast in Darts [29] en SKTime [67] en waren geen automatische modellen. Het AutoETS model werd enkel toegepast in SKTime, waar dit een automatisch model is (parameters werden zelf bepaald) en enkel seizoensperioden werden meegegeven als hyperparameter.

3.3.4 Theta Voor de Theta modellen werd een eenvoudige optimalisatie-functie opgezet. Deze functie zocht naar de meest optimale theta parameter om tot de beste MAPE, MAE en RMSE te komen. Ook andere parameters zoals seizoenen en trend werden om te testen aangepast, maar deze leverden geen betere resultaten op. Het Theta model werd enkel toegepast via de Darts [29] library.

3.3.5 (T)BATS Er werd voor zowel BATS als TBATS gekeken of het manueel toevoegen van seizoensperioden een positieve invloed had op de prestaties van de beide methodes. Over de andere parameters gaat het model zelf kunnen bepalen welke het meest optimaal zijn. Beide methodes werden getest in zowel Darts [29] als SKTime [67].

3.3.6 Prophet & NeuralProphet Zowel Prophet als NeuralProphet methodes zijn afkomstig van Facebook data science teams [40, 42]. De Prophet methode is ook terug te vinden in andere libraries zoals Darts. Voor de Prophet methode werd er gebruikt gemaakt van volgende libraries: Darts en Prophet. De NeuralProphet methode werd enkel gebruikt via de neuralprophet library. Het NeuralProphet model werkt standaard met een min-max normalisatie om de data compacter te maken (voor neurale netwerken) en beter te kunnen werken met sterk fluctuerende data [42]. Veel hyperparameters werden automatisch gekozen zoals epochs en soorten seizoenen. De NeuralProphet methode werd niet toegepast in de experimenten van Sectie 7.5 door slechte *accuracy measures* uit eerdere experimenten.

3.3.7 RNN In Darts [29] en Keras [69] werden RNN modellen toegepast. Deze varieerden van methode, zoals LSTM of GRU. Er werden ook verschillende parameters aangepast zoals: n-rnn-layers, training-length, n-epochs,... Er werd gevarieerd voor het aantal epochs om toe te passen. Voor sommige modellen was het beter om deze niet te hoog te nemen om overfitting te voorkomen. Het aantal epochs geeft aan hoeveel keer het model over de trainingsdata gaat om het model te trainen [69]. In sommige gevallen is het dan beter om te kiezen voor een hoog aantal epochs. Dit om ervoor te zorgen dat het model voldoende vaak over de trainingset gaat om complexe relaties te ontdekken. De input lengte geeft aan hoeveel tijdsperioden er in rekening worden genomen bij het voorspellen van de volgende waarde.

3.3.8 Random Forest Voor het RF model van Darts [29] werd er enkel één parameters meegegeven. Het aantal beslissingsbomen dat het model moet schatten werd meegegeven. Ook werd er gebruik gemaakt van het Random Forest model uit SKLearn package [70]. Voor deze toepassing was het ook noodzakelijk om de data te transformeren naar supervised-data, waarbij dus een label bijgehouden wordt. Hierbij werd ook als parameter gekozen voor het aantal beslissingsbomen (1000) die het model zou genereren.

3.3.9 Gradient Boosting In Darts werd het LightGBM model gebruikt [71]. Hierbij werd enkel de parameter 'lags' aangepast. De *lags* geven aan hoeveel tijdsperioden in het verleden er gebruikt worden om voorspellingen te doen van een volgende tijdsperiode. Verder werd ook het 'Extreme Gradient Boosting' model uit XGBoost [72] toegepast. Voor deze XGBoost werd ook, zoals bij de RF van SKLearn, de tijdreeks getransformeerd naar een supervised-dataset. De LightGBM en XGBoost modellen verschillen licht van elkaar in termen van uitvoering van het Gradient Boosting algoritme.

4 Resultaten

In onderstaande subsecties worden de verschillende resultaten besproken die voortkwamen uit experimenten. Eerst volgt er een beschrijving van de resultaten voor de experimenten op de algemene data (aggregatieniveau). Verder worden de resultaten aangehaald die verkregen werden uit experimenten met data per winkeltype.

4.1 Algemeen

De volgende subsecties bevatten de resultaten uit verschillende experimenten voor de datasets: 1,2,3 en 4. Eerst worden de verschillende accuracy measures vergeleken voor de verschillende voorspellingsmodellen per dataset. Dan volgt er een vergelijking tussen de modellen over de verschillende datasets heen met behulp van de MAPE. Verder wordt de invloed van Covid-19 op de prestaties van voorspellingsmethodes besproken. In de meeste tabellen worden de waardes opgesplitst met het schrap '/' leesteken. Dit geeft aan dat eerst de waarden voor de originele data getoond worden, gevolgd door de waarden op de aan Covid-19 aangepaste data.

4.1.1 Accuracy Measures In onderstaande tabellen worden de verschillende maatstaven getoond met corresponderende voorspellingsmodellen voor de datasets uit Tabel 1. Per maatstaf worden de waardes meegegeven voor testen op de originele data en dan ook voor testen op de gewijzigde data. De tabel is opgedeeld in statistische voorspellingsmodellen en in ML modellen. Voor elke accuracy measure werd de drie best presterende scores in het **vetgedrukt** aangeduid. In de tabellen worden de modellen, die beter presteren dan de benchmark, aangegeven aan de hand van het symbool *. Dit wil zeggen dat de aangeduide modellen significant betere voorspellingen hadden dan de benchmark. Ook wordt er aangegeven met °, dat de Covid-gefilterde dataset significant beter scoorde dan de benchmark. De modellen die significant slechter presteerden dan de benchmark op de originele of de aangepaste dataset werden onderstreept. Wanneer het betreffende model op beide data significant slechter presteerde, werd deze dubbel onderstreept.

In Tabel 3 zijn de maatstaven van de verschillende voorspellingsmethodes te vinden, die werden toegepast op de wekelijkse data uit dataset1. Er zijn enkele

Model	MAPE (%)	MAE	RMSE	Duur (s)	Geheugen (Bytes)
Naïve combined - Darts	17.34 / 17.34	45 278 / 45 278	53 795 / 53 795	**0.02 / 0.02**	**55 632 / 55 632**
* AutoARIMA - SKTime °	14.91 / 13.71	36 440 / 32 208	45 322 / 41 632	240.02 / 50.82	869 456 / 1 334 000
* AutoARIMA - Darts °	16.75 / 13.71	41 282 / 32 208	48 465 / 41 632	85.31 / 46.37	1 094 832 / 1 082 576
* StatsAutoARIMA - SKTime °	15.93 / **13.18**	39 898 / **30 905**	47 103 / 40 688	32.43 / 35.82	45 357 104 / 45 309 200
* AutoETS - SKTime °	16.94 / **12.94**	42 935 / **30 604**	49 486 / **39 979**	2.32 / 1.48	887 216 / 887 536
* ES - SKTime °	**13.25** / 13.49	**28 634** / 31 611	**40 685** / **38 356**	2.91 / 0.78	213 488 / 468 528
* ES - Darts °	**13.25** / 13.49	**28 634** / 31 611	40 685 / 38 356	0.90 / 0.89	179 696 / **176 496**
* BATS - SKTime °	16.76 / 13.97	42 438 / 33 812	49 026 / 42 158	262.32 / 268.44	1 473 008 / 2 286 032
* BATS - Darts °	16.76 / 13.97	42 438 / 33 812	49 026 / 41 835	42.75 / 38.02	**108 112 / 94 704**
* TBATS - SKTime °	16.76 / 14.07	42 438 / 34 071	49 026 / 43 722	218.00 / 283.43	1 708 080 / 1 561 968
* TBATS - Darts °	16.76 / 14.07	42 438 / 34 205	49 026 / 42 158	55.48 / 85.46	**107 536** / 186 032
* Theta - Darts °	**13.06** / **12.45**	**30 170** / **28 234**	**40 829** / **40 268**	**0.12 / 0.11**	196 336 / 195 696
Prophet - Darts	23.90 / 20.05	61 198 / 50 682	73 335 / 63 516	2.94 / 3.24	802 608 / 812 144
Prophet - Prophet	23.90 / 20.05	61 198 / 50 682	73 335 / 63 516	4.18 / 4.01	833 168 / 837 904
NeuralProphet - NP °	24.83 / 17.43	65 164 / 42 336	74 972 / 49 645	32.22 / 28.32	931 216 / 919 760
* RNN - Darts (LSTM) °	15.08 / 17.04	40 676 / 42 559	49 143 / 48 953	6.23 / 11.11	1 014 480 / 1 011 824
* RNN - Keras (LSTM) °	**13.56** / 15.84	**29 476** / 35 380	**41 534** / 45 590	55.87 / 109.91	5 429 696 / 5 433 760
* RandomForest - Darts °	15.76 / 15.70	37 976 / 37 320	48 117 / 47 540	17.63 / 19.83	641 392 / 635 440
* RandomForest - SKLearn °	15.60 / 14.90	36 069 / 34 299	53 271 / 50 772	162.32 / 149.97	179 776 / 182 647
* LightGBM - Darts	19.25 / 25.80	45 214 / 57 887	54 373 / 74 592	**0.07 / 0.08**	202 288 / 205 264
* XGBRegressor - XGBoost °	16.40 / 15.80	37 647 / 36 763	60 502 / 55 958	67.76 / 72.67	187 552 / 193 120

Tabel 3: Accuracy Measures voor wekelijkse data van Dataset1. De benchmark, Naïve method, staat aangeduid in een grijze kleur. Er is te zien dat bijna alle modellen beter presteren (MAPE, MAE en RMSE) dan dit model buiten: Prophet, NeuralProphet en LightGBM. In termen van duur en geheugen presteert geen enkel model beter.

zaken terug te vinden in de tabel. Zo presteren het merendeel van de voorspellingsmethodes beter dan de naïeve methode, de benchmark. Dit is waarschijnlijk te verklaren door het te simpel omvatten van de tijdreeks door de naïeve methode.

Afgaande op de accuracy measures, zijn er enkele modellen die niet beter of zelfs opvallend minder presteren dan de benchmark. Vooral de Prophet en NeuralProphet modellen presteren aanzienlijk slechter. Er werd getracht seizoensparameters aan te passen voor deze twee modellen om betere accuracy measures te verkrijgen, maar dit werd niet bereikt. Bijvoorbeeld het toevoegen van maandelijkse seizoenspatronen per x-aantal maanden leverde geen betere resultaten op. Ook het toepassen van vakantiedagen hielp niet om de accuracy op de test data te verbeteren. Ook de LightGBM uit Darts presteert slechter en voornamelijk op de Covid-aangepaste data. Deze observaties werden ook deels statistisch bevestigd door de Wilcoxon-test uit te voeren, die aantoonde dat de NeuralProphet en de LightGBM methods significant slechter presteren dan de benchmark.

In Tabel 4 worden de verschillende voorspellingsmethodes getoond met bijhorende maatstaven voor de data uit dataset2. In deze tabel wordt duidelijk dat deze zeer hard verschilt van de voorgaande tabel, aangezien er op de maandelijkse data maar één methode significant beter presteert dan de benchmark. Dit kan door verschillende factoren te verklaren zijn. Zo zijn er veel minder datapunten beschikbaar voor deze dataset, waardoor het verkeerd interpreteren van data of het overfitten door verschillende modellen leidt tot slechtere accuraatheid.

De Prophet en NeuralProphet modellen, uit tabel 3 en 4, presteren opvallend slechter dan de benchmark. Op alle accuracy measures scoren zij slechter, ook op de gefilterde data.

Model	MAPE (%)	MAE	RMSE	Duur (s)	Geheugen (Bytes)
Naïve combined - Darts	17.06 / **13.61**	155 305 / 149 980	199 331 / 211 218	**0.003 / 0.01**	**56 208 / 56 240**
AutoARIMA - SKTime	18.15 / 15.64	190 843 / 155 336	227 784 / 190 074	2.68 / 4.46	1 429 040 / 1 362 832
AutoARIMA - Darts	18.15 / 15.71	190 843 / 156 004	227 784 / 192 577	1.59 / 4.31	1 069 296 / 1 068 592
StatsAutoARIMA - SKTime	15.72 / 13.80	168 529 / **132 993**	208 198 / 162 595	33.03 / 27.95	30 628 272 / 30 636 784
AutoETS - SKTime	14.63 / **12.67**	150 008 / **119 926**	177 459 / **153 456**	0.84 / 0.78	807 024 / 589 968
ES - SKTime	**14.54** / 14.11	**148 727** / 133 168	**176 040** / 165 376	**0.03 / 0.06**	453 456 / 464 496
ES - Darts	**14.54** / 14.11	**148 727** / 133 168	**176 040** / 165 376	0.08 / 0.71	168 240 / 164 944
* BATS - SKTime	15.09 / 14.05	152 676 / 134 510	183 341 / **164 211**	68.77 / 92.66	1 508 656 / 1 663 088
* BATS - Darts	15.09 / 14.05	152 676 / 134 510	183 341 / **164 211**	19.50 / 26.71	**105 040 / 93 488**
TBATS - SKTime	14.94 / 17.26	150 463 / 177 633	182 405 / 215 447	95.60 / 77.17	1 184 464 / 1 671 600
TBATS - Darts	14.94 / 17.26	150 463 / 177 633	182 405 / 215 447	40.51 / 39.10	183 952 / 183 214
Theta - Darts	**13.55 / 13.26**	**130 684 / 128 076**	**155 501 / 161 120**	**0.05** / 0.05	195 312 / 197 552
Prophet - Darts	24.78 / 20.17	234 893 / 187 918	263 039 / 217 361	4.50 / 20.14	997 904 / 807 280
Prophet - Prophet	24.78 / 20.17	234 893 / 187 918	263 039 / 217 361	4.17 / 9.98	837 296 / 842 288
NeuralProphet - NP	28.92 / 19.41	302 626 / 194 942	336 138 / 218 866	21.11 / 24.48	826 800 / 683 408
RNN - Darts (LSTM)	14.82 / 15.24	168 720 / 176 641	230 961 / 236 144	19.85 / 23.19	1 981 232 / 1 888 528
RNN - Keras (GRU)	**13.60** / 17.17	**124 244** / 167 193	**161 390** / 205 866	9.82 / 23.73	4 895 040 / 5 427 008
RandomForest - Darts	19.36 / 20.75	185 679 / 196 527	196 527 / 231 772	8.45 / 7.51	842 224 / 615 376
RandomForest - SKLearn	16.90 / 16.60	165 751 / 159 590	205 452 / 196 893	13.03 / 12.26	171 296 / 173 316
LightGBM - Darts	18.54 / 22.69	181 166 / 213 373	224 561 / 241 979	**0.05 / 0.03**	185 488 / 187 961
XGBRegressor - XGBoost	17.80 / 16.20	172 245 / 162 454	198 237 / 212 903	9.53 / 7.46	**92 655 / 90 464**

Tabel 4: Accuracy Measures voor Dataset2. Opmerkelijk is dat enkel BATS significant betere voorspellingen produceerde dan de benchmark. Er zijn wel enkele modellen zoals AutoETS, ES, Theta en RNN, die op bepaalde accuracy measures beter presteren.

Enkele voorspellingsmodellen die meer dan een keer terugkomen in de top drie van Tabel 3 en Tabel 4 zijn: AutoETS, ES, Theta en RNN. Deze kunnen dan ook aangeduid worden als relatief beter/goed presterende modellen over deze twee tabellen op basis van de accuracy measures MAPE, MAE en RMSE.

De accuracy measures uit Tabel 5 zijn relatief slecht in vergelijking met de twee voorgaande tabellen (3 & 4). Het ontbreken van twee jaar extra data heeft naar alle waarschijnlijkheid een invloed op de accuracy measures.

De benchmark presteert goed voor Dataset2, Dataset3 en in mindere mate Dataset4. Er zijn slechts enkele uitzonderingen die significant beter presteren dan de benchmark. In Tabel 5 scoren bijna alle modellen significant slechter dan de benchmark. Dit kan het gevolg zijn van de ingewikkelde patronen/seizoen/trends in de data. Aangezien automatische modellen zoals AutoARIMA en AutoETS ook significant slechter scoren, heeft het slecht presteren van de meeste modellen weinig te maken met de parameters die manueel gekozen werden.

Wanneer beide tabellen 5 en 6 vergeleken worden met elkaar, kan er gezegd worden dat er enkele modellen terugkomen in de top drie. Zo zijn (T)BATS, ES, Prophet en RNN de modellen die op basis van de accuracy measures beter presteren dan andere modellen. Er kan echter ook gekeken worden naar de modellen die significant beter scoren dan de benchmark. Dit zijn dan de beter presterende modellen: StatsAutoARIMA, (T)BATS, Theta, RNN en LightGBM.

Uit de vier tabellen (3,4,5,6) valt er op dat het Prophet model enkel bij Dataset3 en Dataset4 voorkomt als een van de top drie best presterende modellen voor de MAPE maatstaf op de originele data. Dit kan betekenen dat dit model beter presteert op meer ingewikkelde datasets met relatief weinig observaties om te testen. Wat sterk opvalt is dat de Covid-gefilterde data voor bijna alle voorspellingsmodellen betere accuracy measures geeft. Vooral de MAPE wordt voor elk model beter.

Model	MAPE (%)	MAE	RMSE	Duur (s)	Geheugen (Bytes)
Naïve combined - Darts	84.01 / **23.24**	**1 960** / **1 689**	**2 826** / 2 557	**0.03 / 0.03**	**86 768 / 86 800**
AutoARIMA - SKTime	138.56 / 43.51	2 805 / 2 394	3 625 / 2 950	7.58 / 2.09	1 408 336 / 969 968
AutoARIMA - Darts	138.56 / 43.51	2 805 / 2 394	3 625 / 2 950	8.51 / 3.55	757 392 / 1 065 584
StatsAutoARIMA - SKTime °	137.65 / 28.94	2 809 / **1 638**	3 623 / **2 407**	1.84 / 1.18	322 736 / 312 304
AutoETS - SKTime	143.32 / 28.87	2 985 / 1 791	3 749 / 2 519	0.82 / 1.90	887 120 / 906 832
ES - SKTime	93.36 / **27.12**	2 947 / 1 863	3 649 / 2 851	0.34 / 0.34	212 912 / 213 200
ES - Darts	93.36 / **27.12**	2 947 / 1 863	3 649 / 2 851	0.39 / 0.39	175 824 / **174 832**
BATS - SKTime	142.76 / 29.46	2 968 / 1 757	3 737 / **2 472**	69.57 / 57.96	1 558 800 / 1 152 848
BATS - Darts	142.76 / 29.46	2 968 / 1 757	3 737 / **2 472**	21.31 / 24.64	**94 512 / 94 704**
TBATS - SKTime	142.76 / **28.85**	2 968 / **1 730**	3 737 / **2 471**	124.27 / 160.68	1 588 272 / 1 494 928
TBATS - Darts	142.76 / **28.85**	2 968 / **1 730**	3 737 / **2 471**	59.93 / 71.51	196 464 / 191 024
Theta - Darts	122.49 / 43.65	3 218 / 2 633	4 242 / 3 641	**0.09 / 0.10**	184 592 / 192 848
Prophet - Darts	**60.16** / 39.13	2 453 / 2 613	**3 240** / 3 524	3.34 / 3.38	802 960 / 802 640
Prophet - Prophet	**60.16** / 39.13	2 453 / 2 613	**3 240** / 3 524	4.01 / 3.81	833 456/ 836 080
NeuralProphet - NP	107.82 / 54.06	2 957 / 3 274	3 792 / 4 028	29.22 / 27.99	806 480 / 921 712
RNN - Darts (GRU)	**39.65** / 36.19	3 687 / 3 222	4 483 / 4 112	6.75 / 6.74	1 012 048 / 1 020 336
RNN - Keras (LSTM)	66.93 / 31.45	**1 764** / 1 760	2 623 / 2 508	70.05 / 73.27	5 431 904 / 5 429 568
RandomForest - Darts	136.70 / 29.66	3 510 / 1 852	4 429 / 2 550	17.92 / 21.04	644 016 / 634 352
RandomForest - SKLearn	**57.80** / 42.60	**2 155** / 2 357	3 388 / 3 630	118.73 / 118.65	**181 088** / 182 650
LightGBM - Darts	118.17 / 33.11	2 603 / 1 931	**3 383** / 2 536	**0.08 / 0.07**	203 088 / 202 608
XGBRegressor - XGBoost	68.30 / 43.20	2 315 / 2 430	3 762 / 3 770	65.96 / 76.26	186 592 / 192 192

Tabel 5: Accuracy Measures voor Dataset3. Er zijn veel voorspellingsmethoden die significant slechtere voorspellingen produceren dan de benchmark. De benchmark presteert hier goed op basis van de verschillende accuracy measures.

Model	MAPE (%)	MAE	RMSE	Duur (s)	Geheugen (Bytes)
Naïve combined - Darts	72.40 / **25.61**	11 392 / 8 795	14 495 / 11 863	**0.01 / 0.01**	**56 144 / 56 144**
AutoARIMA - SKTime	65.86 / 35.49	11 967 / 10 953	14 327 / 13 750	7.74 / 27.76	877 840 / 1 348 464
AutoARIMA - Darts	63.24 / 35.25	11 122 / 10 844	13 772 / 13 546	4.34 / 25.51	822 480 / 799 376
StatsAutoARIMA - SKTime	65.10 / 36.36	12 878 / 11 797	16 304 / 15 670	1.45 / 32.44	233 168 / 31 080 048
AutoETS - SKTime	54.12 / 37.60	9 685 / 11 917	11 691 / 16 707	2.16 / 2.66	524 528 / 905 680
ES - SKTime	**40.65** / **25.30**	**7 466** / **7 838**	**9 214** / 11 482	0.31 / 0.23	210 352 / 466 896
ES - Darts	**40.65** / **25.30**	**7 466** / **7 838**	**9 214** / 11 482	0.35 / 0.30	182 000 / 179 344
BATS - SKTime °	54.59 / 26.30	9 214 / **7 748**	12 136 / **8 891**	58.67 / 30.37	1 513 328 / 1 242 704
BATS - Darts °	54.59 / 26.30	9 214 / **7 748**	12 136 / **8 891**	24.44 / 19.60	**96 048 / 98 192**
TBATS - SKTime °	45.92 / 27.77	**7 617** / 8 405	**9 727** / **10 258**	121.12 / 109.64	1 093 360 / 1 560 368
TBATS - Darts °	45.92 / 27.77	**7 617** / 8 405	**9 727** / **10 258**	51.91 / 47.31	193 392 / 192 400
* Theta - Darts	58.93 / 29.15	9 996 / 8 600	12 027 / 11 922	**0.05 / 0.04**	196 688 / 198 416
Prophet - Darts	**42.04** / 44.59	10 642 / 14 011	15 133 / 17 622	4.03 / 4.51	801 328 / 803 120
Prophet - Prophet	**42.04** / 44.59	10 642 / 14 011	15 133 / 17 622	4.13 / 5.09	827 696 / 831 760
NeuralProphet - NP	64.71 / 39.41	11 981 / 11 931	15 507 / 16 279	22.17 / 21.07	708 112 / 853 104
RNN - Darts (LSTM)	**33.66** / 26.28	12 945 / 10 882	17 005 / 16 617	22.02 / 22.45	1 909 360 / 1 924 400
* RNN - Keras (LSTM) °	68.26 / **16.75**	**9 485** / **5 859**	12 067 / **7 526**	15.67 / 20.45	5 426 592 / 5 445 728
RandomForest - Darts	92.14 / 93.07	13 196 / 13 357	16 535 / 16 565	8.31 / 10.75	621 552 / 615 568
RandomForest - SKLearn	86.00 / 31.70	12 987 / 8 940	16 274 / 14 346	54.04 / 51.57	171 232 / 171 264
* LightGBM - Darts	60.43 / 30.42	9 559 / 8 962	**10 955** / 10 944	**0.05 / 0.05**	186 832 / 186 832
XGBRegressor - XGBoost	88.40 / 32.40	14 862 / 9 505	18 430 / 13 551	10.72 / 12.19	**90 368 / 90 304**

Tabel 6: Accuracy Measures voor Dataset4. Enkele modellen die significant betere voorspellingen produceren dan de benchmark: (T)BATS, Theta, RNN en LightGBM. Op basis van de accuracy measures doen de ES modellen het ook goed.

Uit de verschillende accuracy measures uit bovenstaande tabellen en uit de errors afkomstig van experimenten, kan afgeleid worden dat methodes die gebaseerd zijn op eenzelfde bron-model ('wrapper'/omhulsel rond bron) bijna altijd dezelfde prestaties leveren. Dit is zo bij ES, BATS, TBATS en Prophet. Een model dat hier licht van afwijkt is AutoARIMA, die op sommige experimenten anders presteerde dan de tegenhanger uit de andere package. Dit kan te verklaren zijn door toch een verschil in default parameters of een verschil in implementatie van het onderliggende model.

4.1.2 Vergelijking modellen over datasets Een volgende aspect dat onderzocht werd, was het verschil in prestaties van modellen op verschillend aggregatieniveau en specifiek over wekelijks ten opzichte van maandelijkse data. In deze subsectie volgt een vergelijking van de verschillende voorspellingsmodellen over de datasets met verschillende schaal. Dit werd getest met behulp van de eerder gebruikte t-test. Er werd een verdeling opgesteld per dataset met de MAPE van alle toegepaste voorspellingsmodellen. Zo werden de MAPEs van alle methodes vergeleken met zijn tegenhanger op verschillend aggregatieniveau om te zien of hier een significant verschil aanwezig was.

Uit deze testen kwam naar boven dat er enkel een significant verschil gevonden werd voor Dataset3 en Dataset4. Concreet, de modellen toegepast op de maandelijkse data presteerde significant beter dan de modellen op de wekelijkse data. Voor Dataset1 ten opzichte van Dataset2 werd er enkel gevonden dat de modellen op de maandelijkse data een lagere gemiddelde MAPE hadden dan de modellen op de wekelijkse data, maar deze verschilden niet significant. Deze twee observaties zijn te verklaren door de ingewikkeldere seizoenen en hoge frequentie van seizoenen bij wekelijkse data, die het voor modellen moeilijker maakt om accurate voorspellingen te produceren. Anderzijds zou er verwacht worden dat de aanwezigheid van extra datapunten bij wekelijkse voorspellingen een positieve invloed heeft op de prestaties doordat het model langer en meer kan trainen.

4.1.3 Invloed Covid In Tabel 10 worden de gebruikte datasets opgesomd met bijhorende toegepaste voorspellingsmethodes. Deze tabel geeft in het groen de modellen aan die significant beter presteren op de Covid-gefilterde data. In het rood worden de modellen weergegeven die significant slechter presteerde, en in het wit de modellen die geen significante verschillen hadden tussen de originele en de gewijzigde dataset.

Er zijn enkele zaken die geleerd kunnen worden uit Tabel 10. Zo toont het, dat het wijzigen van de originele data om anomalieën in de data aan te pakken, geen garantie is op een significant beter resultaat. Toch presteren veel voorspellingsmethodes beter wanneer de data aangepast werd, maar dit verschilt ook per dataset. Dus in de meeste gevallen, hier 48.87 procent, zorgde het aanpassen van de Covid-anomalieën niet tot een significant verschil in de prestaties. In 39.77 procent van de gevallen zorgde het echter wel voor een significante verbetering en in 11.36 procent van de gevallen voor een significante verslechtering.

Verder toont Tabel 10 ook het verschil tussen de verschillende datasets aan. Zo kan voor een bepaalde methode een wijziging van data zorgen voor een beter effect op de prestaties, terwijl dat voor een andere methode net voor een verslechtering zorgt. Ook kan dit behoorlijk verschillen per dataset die er gebruikt wordt. Het aanpassen van de data op Dataset1 voor het ES model zorgt voor een significante daling in prestaties terwijl het voor Dataset2 net een significante verbetering veroorzaakt. Dit geeft aan hoe verschillend een voorspellingsprobleem kan zijn naargelang de eigenschappen van de data.

De resultaten bij Dataset4 kunnen ook betekenen dat deze data robuuster zijn tegen anomalieën dan de andere datasets. Zo kan er ook geredeneerd worden bij de modellen. RandomForest van SKLearn en XGBRegressor van XGBoost tonen geen significante verschillen in prestaties, wat aangeeft dat deze modellen robuuster zijn en minder waarde geven aan anomalieën in de data.

4.2 Winkeltypes

In de volgende twee subsecties wordt er verduidelijking gegeven over de verschillen die er te zien zijn in termen van prestaties voor verschillende aggregatieniveaus (winkeltypes & algemeen) en voor de verschillende winkeltypes onderling. Dit wordt gedaan met behulp van een t-test, die gespecificeerd werd in Sectie 3.2.

4.2.1 Prestaties Aggregatieniveaus Om een compleet overzicht te krijgen in de voorspellingen over de verschillende aggregatieniveaus, werden ook de voorspellingsmodellen op winkel niveau vergeleken met die op algemeen niveau. In Sectie 7.5 worden twee tabellen getoond die meer duidelijkheid geven over dit aspect.

De twee tabellen geven aan in welke mate er een significant verschil is tussen voorspellingen op winkel en algemeen niveau. Op beide tabellen zijn voornamelijk rode cellen te zien, die impliceren dat het model op algemeen niveau significant betere voorspellingen produceerde. Er is op zijn minst te zien dat de meest geaggregeerde modellen (algemeen niveau) beter presteren dan de minder geaggregeerde en dit voornamelijk voor de maandelijkse data. Voor de wekelijkse data scoorde bijna 45 procent van de modellen beter op de meest geaggregeerde data maar ook 18 procent scoorde significant slechter. Op de maandelijkse data is in bijna 30 procent van de gevallen significant beter gescoord door modellen op de algemene data en maar in één procent van de gevallen scoorde de algemene modellen slechter. Dit geen sluitend bewijs maar het geeft wel een indicatie dat algemene modellen beter presteren dan modellen apart per winkeltype. Het is ook duidelijk dat deze verschillen zeer afhankelijk zijn per voorspellingsmodel en winkeltype.

Het zou te verklaren kunnen zijn doordat de modellen op winkeltype niveau specifieker gaan zijn en dus ook aanpassingen aan parameters gaan vereisen om tot goede voorspellingen te komen, maar dan zou er ook verwacht worden dat automatische modellen betere prestaties hebben op winkeltype niveau, wat niet over het geval is.

4.2.2 Prestaties ten opzichte van winkeltypes Als laatste werden er nog experimenten uitgevoerd om te bekijken hoe winkeltypes onderling presteerden. In onderstaande tabellen 7 en 8 worden enkele belangrijke metrieken getoond met ook de drie best presterende modellen op basis van de MAPE. De metrieken rond error en MAPE geven een indicatie over de voorspelbaarheid van bepaalde winkeltypes en welke beter of slechter te voorspellen zijn.

Winkeltype	% Error	Afwijking	MAPE	StDev MAPE	Beste modellen (3) (MAPE)
FR+7500	22.64 / 20.74	27.46 / 28.80	19.81 / 19.20	9.75 / 9.50	RNN, AutoETS, AA / RNN, Theta, BATS
FR-7500	18.86 / 17.80	18.01 / 18.47	17.63 / 15.70	8.27 / 7.27	ES, RF, AutoETS / AA, Theta, SFAA
Online	25.29 / 21.62	15.26 / 13.59	18.67 / 18.16	13.53 / 13.10	RF, AA, RNN/ AA, SFAA, RF
PERDW	21.36 / 17.59	16.89 / 18.01	19.58 / 16.44	13.47 / 7.19	RF, AutoETS, ES / Theta, SFAA, TBATS
PERSA	19.31 / 12.06	11.34 / 9.81	19.15 / 11.08	11.67 / 9.24	RF, AutoETS, XGB / TBATS, Theta, AA
SHOP	25.51 / 7.53	11.51 / 6.16	23.47 / 6.55	19.64 / 6.99	Naïve, RF, AutoETS / RF, XGB, RNN
STAD	17.75 / 17.14	9.23 / 8.88	13.98 / 11.30	10.47 / 5.17	RNN, RF, XGB / RNN, RF, AA

Tabel 7: Verschillende metrieken voor Dataset1. De FR+7500 en Online categorie zijn het moeilijkst om te voorspellen gezien de hoge % Error en (standaard)Afwijking op deze errors. De meeste modellen werden voorspelbaarder wanneer deze toegepast werden op de aangepaste data.

In Tabel 7 zijn enkele zaken die opvallen en die besproken kunnen worden. Er zijn de gemiddelde percentage errors (% Error) die aangeven wat het gemiddelde percentage is van de fout bij voorspellingen over alle modellen heen. Bij de winkeltypes PERSA en SHOP hebben de aanpassingen van de data een goed effect gehad op de voorspelbaarheid van de dataset. Ook is het duidelijk dat de Covid-aanpassingen op de data een goede invloed hadden op de voorspelbaarheid per winkeltype. Deze bevindingen zijn verschillend per winkeltype en hangen dus af van de eigenschappen van de data.

Winkeltype	% Error	Afwijking	MAPE	StDev MAPE	Beste modellen (3) (MAPE)
FR+7500	21.24 / 19.20	15.32 / 14.82	18.19 / 16.51	9.34 / 8.02	AA, AutoETS, RNN/ AA, Theta, RNN
FR-7500	23.27 / 22.18	19.38 / 20.44	20.71 / 19.57	9.50 / 8.44	AutoETS, AA, RNN / AA, Theta, Naïve
Online	22.48 / 23.10	18.11 / 18.16	21.14 / 21.77	13.28 / 14.20	RNN, RF, Naïve/ RF, Naïve XGB
PERDW	18.59 / 17.17	14.03 / 13.87	16.92 / 15.12	9.23 / 7.29	AA, AutoETS, ES / AA, AutoETS, Theta
PERSA	19.97 / 16.92	15.33 / 13.47	18.30 / 15.91	10.00 / 7.86	AutoETS, AA, ES / RNN, Theta, AA
SHOP	26.69 / 23.09	19.73 / 18.81	25.16 / 21.68	15.50 / 12.36	RF, AutoETS, XGB/ Theta, XGB, RF
STAD	21.79 / 19.51	17.73 / 17.29	20.50 / 17.41	11.69 / 7.72	RNN, Naïve, AutoETS / Theta, TBATS, RNN

Tabel 8: Verschillende metrieken voor Dataset2. De FR+7500, PERDW en PERSA winkel categorieën zijn de meest voorspelbare. SHOP is echter wel moeilijker om te voorspellen en leidt tot grotere fouten.

De waarden in Tabel 8 tonen het verschil tussen de originele en Covid-aangepaste data minder. Er is minder sprake van een invloed op de prestaties. Dit geeft aan dat de maandelijkse data voorspellingen betere algemene voorspellingen produceerden (op basis van de MAPE) of minder vatbaar zijn voor wijzigingen in de data aangezien deze voorspellingen op hoger aggregatieniveau gebeuren.

Voor meer gedetailleerde resultaten over de MAPE van de verschillende modellen over de verschillende winkeltypes, kan er in de appendix Tabel 13

en Tabel 14 geraadpleegd worden. Op beide tabellen werden de de top drie presterende modellen per winkeltype **vetgedrukt**. Er zijn enkele modellen die vaak terugkomen in de top drie van de tabellen (8,9,13 en 14), nl. AutoARIMA, AutoETS, Theta en RNN. Voor de wekelijkse data presteren RandomForests en XGBRegressors ook relatief goed.

5 Conclusie

Een eerste en belangrijke conclusie die er getrokken kan worden na het uitvoeren van deze studie: er is geen algemeen best voorspellingsmodel voor verschillende datasets. Uit de resultaten is ook gebleken dat er nergens één speciaal voorspellingsmodel altijd als best presterend naar boven kwam. De prestaties van een model hangen af van de dataset en zijn eigenschappen. Elk voorspellingsprobleem is een probleem op zichzelf en zou zo ook behandeld moeten worden. Er kunnen wel enkele voorspellingsmethodes naar voor geschoven worden als "algemeen" goed of minder goed presterende modellen. Zo bleek uit de experimenten dat over het algemeen en over de verschillende datasets en aggregatieniveaus, voornamelijk: AutoETS, Theta en RNN de beste voorspellingen produceerden.

Er werd onderzocht of de prestaties van voorspellingen op wekelijkse data differentieerde met die op maandelijkse data. Uit de experimenten bleek dat er voor één databron een significant verschil opgemerkt kon worden tussen de twee aggregatieniveaus. De maandelijkse data uit deze databron gaven significant betere MAPE accuracy measures dan wekelijkse data. Voor de andere databron was er een gelijkaardig patroon te zien, maar verschilden deze niet significant. Voor de betreffende datasets is het dus beter om voorspellingen te doen op maandelijkse dan op wekelijkse data.

Verder werden ook de invloeden van de Covid-pandemie op de prestaties onderzocht. Sommige datasets waren robuuster en werden minder beïnvloed dan andere. Het grootste deel van de voorspellingsmodellen werden niet significant beter of slechter door het aanpassen van de Covid-data. Toch presteerde bijna 40 procent van de modellen beter wanneer de originele data werd aangepast. Slechts een klein deel (tien procent) van de modellen werden significant slechter door het aanpassen van de data.

Uiteindelijk werden de voorspellingen op verschillende aggregatieniveaus (winkeltypes ten opzichte van algemene voorspellingen) bekeken om te achterhalen of deze voor verschillen zorgden. Er werd geen sluitend bewijs gevonden over de al dan niet beter presterende modellen. Echter was er wel een grote indicatie naar de meerwaarde van voorspellingen op algemeen niveau. Dit valt ook te verklaren doordat de modellen op de data per winkeltype meer model tuning vereisen aangezien dit aparte voorspellingsproblemen zijn met eigen data-eigenschappen. Anderzijds zou er dan wel verwacht worden dat automatische modellen beter presteren, wat niet altijd het geval was.

6 Limitaties

In deze sectie volgt een korte beschrijving van verschillende aspecten die een invloed hebben gehad op de (bruikbaarheid van) resultaten uit het onderzoek. Deze aspecten zouden in toekomstig onderzoek verder aangepakt kunnen worden om tot betere inzichten te komen.

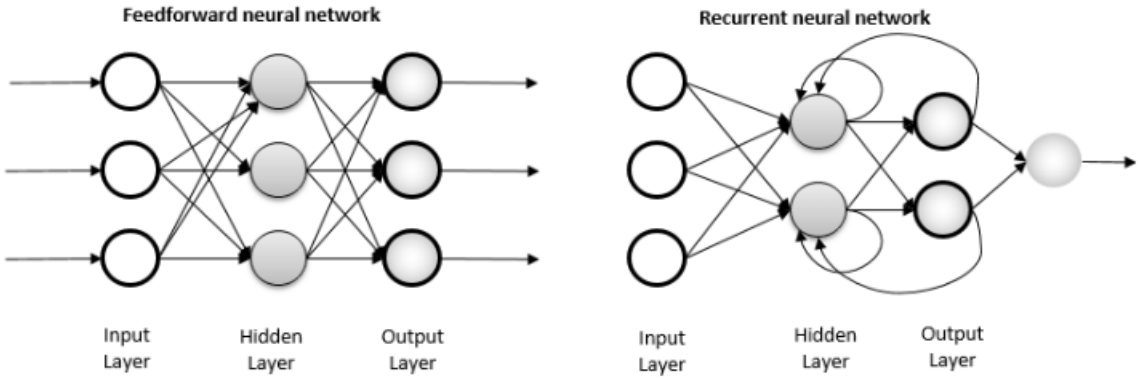
Een eerste limitatie van het onderzoek is dat de voorspellingsmethodes op te weinig datasets toegepast werden om een algemeen besluit te kunnen verkrijgen over de methodes zelf. Voor dit onderzoek, dat voornamelijk inzicht probeerde te geven in de prestaties van methodes op de aangeleverde datasets, heeft dit weinig invloed gehad. Toekomstige onderzoeken, die tot een algemenere besluitvorming willen komen, zouden gebruik moeten maken van meer databronnen met verschillende eigenschappen.

Er werd tijdens het onderzoek gebruik gemaakt van diverse voorspellingsmethodes / functies waarbij de methode / functie zelf op zoek ging naar een best presterend voorspellingsmodel zoals AutoARIMA of AutoETS. Dit wordt bekomen door automatisch verschillende voorspellingsmodellen op te stellen en automatisch te kiezen welke het best presteert. Ook AutoTS [73] werd bijvoorbeeld gebruikt, maar kon niet voldoende ontleed en aangepast worden om in de resultaten van het onderzoek mee op te nemen. Voor toekomstig onderzoek kan het waardevol zijn om meer van deze ”automatische” methodes toe te passen om te zien of significant betere voorspellingen geven.

Verder kan het ook een limitatie zijn dat er enkel univariate voorspellingsmodellen werden gebruikt in het onderzoek. Enkele methodes konden ook werken met multivariate datasets maar dit werd niet uitgetest. Als verdere uitbreiding zou het waardevol kunnen zijn om ook te bekijken of het voordelig is voor voorspellingsmodellen om meerdere verklarende variabelen te gebruiken en of deze de prestaties significant beïnvloeden.

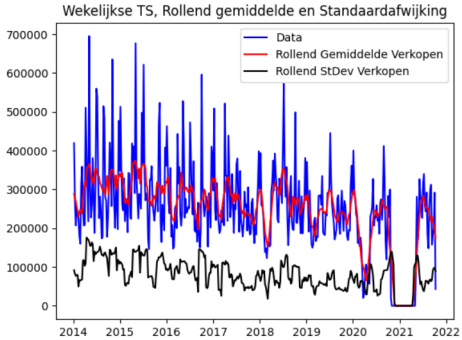
7 Appendix

7.1 Neurale Netwerken

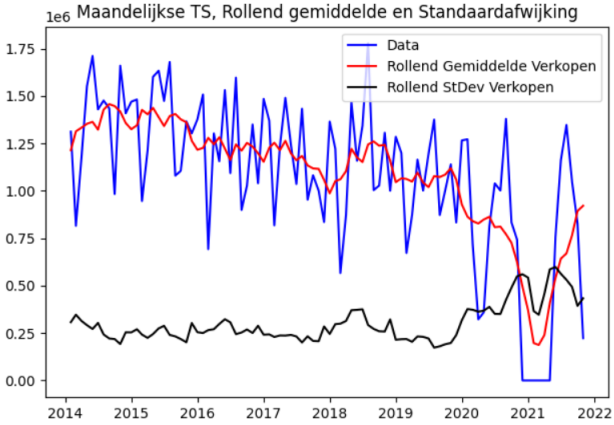


Figuur 1: FFNN vs RNN: Enkel het in rekening nemen van de input uit voorgaande nodes bij FFNN, in contrast met de "feedbackloops" en het gebruik van input en output van voorgaande nodes bij RNN. Dit is een vereenvoudigde weergave van FFNN en RNN, zonder achterliggende logica en details te vermelden.

7.2 Data



Figuur 2: Dataset1: Wekelijkse verkopen worden getoond met bijhorend rollend gemiddelde en standaardafwijking (StDev). Covid effecten te zien in maanden maart en april, alsook de ontbrekende data voor een periode van 6 maanden.

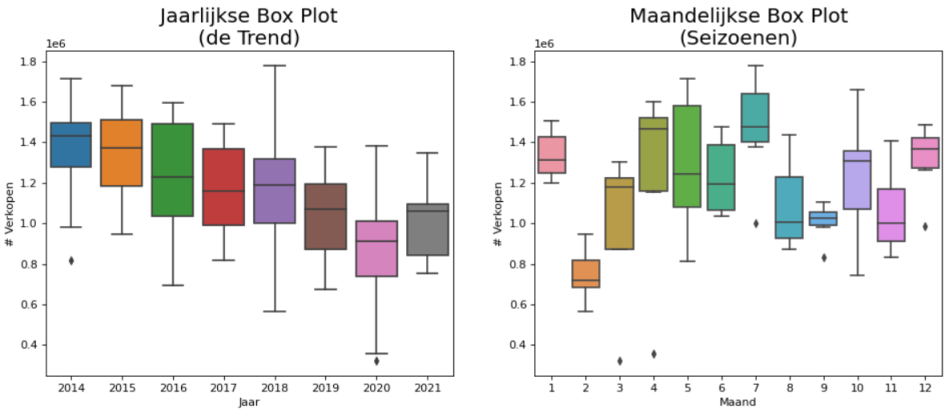


Figuur 3: Dataset1: Maandelijkse verkopen worden getoond met bijhorend rollend gemiddelde en standaardafwijking (StDev). Covid effecten te zien in maanden maart en april, alsook de ontbrekende data voor een periode van 6 maanden.

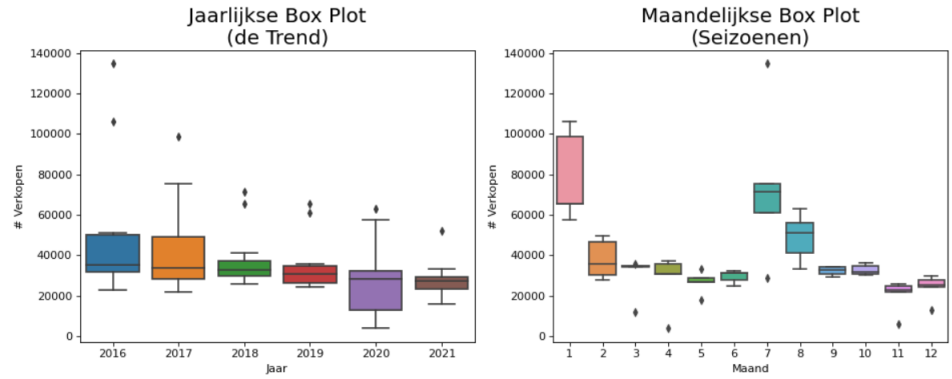
7.3 Trends en seizoenen

Data	Dataset2	Dataset4
Trend	- 5 050	- 495
Jan	89 502	31 496
Feb	- 408 696	- 3 024
Mrt	- 157 489	- 9 547
Apr	70 521	-11 832
Mei	72 613	- 10 266
Jun	68 562	- 7 892
Jul	313 027	34 518
Aug	- 58 984	9 801
Sept	- 151 399	- 3 466
Okt	74 942	- 1 716
Nov	- 88 991	- 16 229
Dec	176 392	- 11 842

Tabel 9: Gewichten verkregen uit lineaire regressie per maandelijkse dataset met bijhorende trend & seizoensgebondenheid per maand. Deze komen overeen met de boxplots uit Figuur 4 & 5.



Figuur 4: Dataset 1 & 2: Het aantal verkopen per jaar en maand afgebeeld via een box plot. Een negatieve trend is terug te vinden in de jaarlijkse box plot. De seizoenen/(on)populaire maanden zijn te zien in maandelijkse box plot. Het op en neer gaan van deze box plots geeft aan dat er seizoensgebondenheid te vinden is in de data.



Figuur 5: Dataset 1 & 2: Het aantal verkopen per jaar en maand afgebeeld via een box plot. Een lichte negatieve trend is terug te vinden in de jaarlijkse box plot. De seizoenen/(on)populaire maanden zijn te zien in maandelijkse box plot met de maand januari, juli en augustus als best verkopende.

7.4 Significante verschillen Covid-data

Model	Dataset1	Dataset2	Dataset3	Dataset4
Naïve combined - Darts				
AutoARIMA - SKTime				
AutoARIMA - Darts				
StatsAutoARIMA - SKTime				
AutoETS - SKTime				
ES - SKTime				
ES - Darts				
BATS - SKTime				
BATS - Darts				
TBATS - SKTime				
TBATS - Darts				
Theta - Darts				
Prophet - Darts				
Prophet - Prophet				
NeuralProphet - NP				
RNN - Darts (LSTM)				
RNN - Keras (GRU)				
RandomForest - Darts				
RandomForest - SKLearn				
LightGBM - Darts				
XGBRegressor - XGBoost				

Tabel 10: Significant verschil tussen gefilterde data en originele data met Covid invloeden. De groene cellen geven aan dat het model op de gefilterde data significant beter presteerde dan op de originele data. De rode cellen betekenen dat het model op de gefilterde data significant slechter presteerde dan op de originele data. Een groot deel, bijna 40 procent, groene cellen zijn terug te vinden in de tabel, wat aangeeft dat er een indicatie is van beter presterende modellen op Covid-aangepaste data.

7.5 Winkeltype vs. algemeen model

Model	FR+	FR-	ON	PERD	PERS	SH	ST
Naïve							
AA							
SFAA							
AES							
ES							
BATS							
TBATS							
Theta	/	/	/	/	/	/	/
Prophet							
RNN - Darts							
RNN - Keras							
RF - Darts							
RF - SKLearn							
LightGBM - Darts							
XGB - XGBoost							

Tabel 11: Resultaten voor de wekelijkse data (Dataset1): Er worden significante verschillen tussen algemeen en winkeltype aangegeven per kleur. Groene cellen betekenen dat de voorspellingen per winkeltype significant beter zijn dan die op algemeen niveau en vice versa voor de rode cellen. Hier presteren 45 procent van de modellen beter op algemeen niveau dan per winkeltype.

Model	FR+	FR-	ON	PERD	PERS	SH	ST
Naïve							
AA							
SFAA							
AES							
ES							
BATS							
TBATS							
Theta	/	/	/	/	/	/	/
Prophet							
RNN - Darts							
RNN - Keras							
RF - Darts							
RF - SKLearn							
LightGBM - Darts							
XGB - XGBoost							

Tabel 12: Resultaten voor de maandelijkse data (Dataset2): Er worden significante verschillen tussen algemeen en winkeltype aangegeven per kleur. Groene cellen betekenen dat de voorspellingen per winkeltype significant beter zijn dan die op algemeen niveau en vice versa voor de rode cellen. Hier presteren 30 procent van de modellen beter op algemeen niveau dan per winkeltype, maar ook 18 procent van de modellen presteert slechter op algemeen niveau.

7.6 Winkeltypes: MAPE per model

Model	FR+	FR-	Online	PERDW	PERSA	SHOP	STAD
Naïve - Darts	18.52	16.89	15.47	18.32	15.09	29.11	14.74
Naïve_cov - Darts	18.52	16.89	15.47	18.32	12.96	**1.61**	13.66
AA	**17.18**	17.66	**12.87**	17.90	15.60	34.29	9.95
AA_cov	17.02	**13.33**	**11.44**	16.18	**5.85**	3.94	**8.58**
SFAA - SKTime	17.89	15.95	13.96	20.46	27.26	33.00	80.00
SFAA_cov - SKTime	17.59	**14.26**	**11.73**	**14.32**	6.40	3.88	79.57
AutoETS - SKTime	**16.72**	**15.89**	32.13	**15.56**	**8.44**	28.14	13.80
AutoETS_cov - SKTime	19.23	16.00	17.24	16.83	10.91	**1.98**	11.35
ES	18.46	**15.13**	20.44	**15.67**	22.75	36.41	10.57
ES_cov	22.69	15.21	16.20	17.00	21.46	6.49	9.99
BATS	19.69	19.45	18.20	19.08	16.60	33.63	12.21
BATS_cov	**16.77**	14.96	16.06	15.32	6.42	4.25	12.36
TBATS	17.91	17.86	18.20	18.17	16.33	31.83	12.21
TBATS_cov	17.55	14.27	16.06	**14.43**	**5.51**	3.77	11.98
Theta - Darts	/	/	/	/	/	/	/
Theta_cov - Darts	**16.64**	**13.69**	14.44	**13.75**	**5.62**	5.58	12.20
Prophet	26.59	26.14	18.27	26.35	35.13	50.17	29.71
Prophet_cov	23.30	21.71	27.33	20.46	25.76	15.48	16.98
NeuralProphet (NP)	36.15	32.34	39.40	34.93	34.15	71.39	44.64
NP_cov	22.72	21.41	55.24	20.88	20.68	28.50	18.43
RNN - Darts	**16.35**	17.74	93.01	15.81	16.58	10.56	14.72
RNN_cov - Darts	**16.11**	18.73	67.51	23.49	27.03	9.92	19.58
RNN - Keras	45.20	24.00	**13.92**	41.50	10.23	**8.09**	**7.90**
RNN_cov - Keras	19.50	18.40	46.43	19.10	8.70	6.10	**7.10**
RF - Darts	19.45	17.63	20.82	17.78	18.03	11.20	12.23
RF_cov - Darts	19.79	17.88	20.04	17.21	8.83	2.56	12.94
GB - Darts	35.43	22.83	52.54	59.65	36.03	13.36	14.99
GB_cov - Darts	38.17	31.60	31.36	24.17	32.29	11.63	14.31
RF - SKLearn	19.90	**15.60**	**11.50**	**14.80**	**7.20**	**2.90**	**8.20**
RF_cov - SKLearn	19.20	15.80	**13.00**	15.30	5.90	**1.70**	**7.90**
XGB - XGBoost	23.80	17.70	14.80	19.60	**9.80**	**4.10**	**9.50**
XGB_cov - XGBoost	22.70	16.20	16.90	18.40	8.00	2.80	10.20

Tabel 13: MAPE voor de wekelijkse data, opgedeeld per winkeltype. In het **vetgedrukt** staan de drie best presterende modellen per winkeltype. Vaak voorkomende modellen in top drie: AA, AutoETS, Theta, RNN, RF en XGB.

Model	FR+	FR-	Online	PERDW	PERSA	SHOP	STAD
Naïve - Darts	19.97	18.95	**17.90**	18.79	23.51	21.73	**18.40**
Naïve_cov - Darts	21.44	**18.95**	**17.90**	20.49	17.65	21.73	18.87
AA	**15.16**	**17.76**	23.89	**13.25**	**14.50**	27.77	19.70
AA_cov	**14.07**	**17.44**	23.89	**12.12**	**12.90**	21.88	22.56
SFAA - SKTime	29.05	24.22	30.06	26.91	21.33	36.92	32.46
SFAA_cov - SKTime	23.06	22.39	30.06	20.45	14.42	25.89	19.63
AutoETS - SKTime	**15.24**	**17.41**	25.28	**13.76**	**14.13**	**18.95**	**18.47**
AutoETS_cov - SKTime	15.39	19.84	25.28	**13.61**	13.88	18.02	22.57
ES	18.73	20.56	21.38	**15.80**	**14.61**	28.27	22.85
ES_cov	15.77	22.08	21.38	14.01	13.85	21.72	18.18
BATS	19.68	23.06	25.68	16.97	15.84	29.52	23.67
BATS_cov	17.09	22.94	25.68	15.45	14.29	25.24	18.75
TBATS	18.71	23.63	25.00	16.69	15.21	28.70	21.19
TBATS_cov	24.98	23.32	25.00	16.49	14.85	23.44	**17.58**
Theta - Darts	/	/	/	/	/	/	/
Theta_cov - Darts	**14.39**	**19.30**	21.62	**13.66**	**12.51**	**16.33**	**16.16**
Prophet	26.01	31.87	30.68	28.03	31.90	38.83	25.46
Prophet_cov	22.87	29.37	30.67	23.08	23.94	28.56	19.06
NeuralProphet (NP)	41.51	38.49	56.42	40.58	41.75	69.92	54.74
NP_cov	31.68	31.67	60.99	30.05	22.48	54.77	32.67
RNN - Darts	**15.80**	21.50	21.30	16.20	22.50	22.00	**17.50**
RNN_cov - Darts	16.62	22.00	21.30	16.88	30.00	22.70	**17.70**
RNN - Keras	17.50	**17.90**	**17.30**	18.00	15.00	30.00	25.70
RNN_cov - Keras	**14.80**	21.10	29.70	18.90	**12.20**	21.70	18.50
RF - Darts	19.90	24.80	**17.70**	17.00	19.90	26.10	21.50
RF_cov - Darts	17.90	21.90	**17.20**	16.70	22.80	33.50	19.20
GB - Darts	21.10	21.80	18.30	17.10	15.80	20.70	19.90
GB_cov - Darts	25.10	24.00	18.30	17.20	18.40	19.70	19.80
RF - SKLearn	17.20	22.30	18.90	17.90	17.80	**17.70**	19.30
RF_cov - SKLearn	15.30	21.30	19.20	16.10	17.60	**17.30**	18.90
XGB - XGBoost	22.80	25.50	18.20	20.30	19.10	**19.40**	21.80
XGB_cov - XGBoost	15.90	19.40	**18.20**	21.00	20.90	**16.90**	22.30

Tabel 14: MAPE voor de maandelijkse data, opgedeeld per winkeltype. In het **vetgedrukt** staan de drie best presterende modellen per winkeltype. Vaak voorkomende modellen in top drie: AA, RNN, AutoETS en Theta.

Bronnen

[1] John C. Chambers, Satinder K. Mullick, and Donald D. Smith. “How to Choose the Right Forecasting Technique”. In: *Harvard Business Review* (July 1971). Section: Financial analysis.

[2] Özlem İpek Kalaoglu et al. “RETAIL DEMAND FORECASTING IN CLOTHING INDUSTRY”. scheme=”ISO639-1”. In: *Textile and Apparel* 25.2 (Dec. 2015). Number: 2, pp. 172–178.

[3] A. O. Adedayo, Olu Ojo, and JO Kolade. *Operations Research in Decision Analysis and Production Management*. 2006.

[4] Cássia Veiga, Claudimar Veiga, and Luiz Duclós. “THE ACCURACY OF DEMAND FORECAST MODELS AS A CRITICAL FACTOR IN THE FINANCIAL PERFORMANCE OF THE FOOD INDUSTRY”. In: *Future Studies Research Journal: Trends and Strategies* 2 (Aug. 2010), pp. 81–104.

[5] John Kolade Obamiro. “Demand Forecasting and Measuring Forecast Accuracy in a Pharmacy”. French. In: *Acta Universitatis Danubius. Oeconomica* 15.3 (2019). Place: Galati, Romania Publisher: Universitatea Danubius Galati Section: Economic Development, Technological Change, and Growth.

[6] Robert Fildes, Shaohui Ma, and Stephan Kolassa. “Retail forecasting: Research and practice”. en. In: *International Journal of Forecasting* (Dec. 2019). DOI: 10.1016/j.ijforecast.2019.06.004.

[7] Chaitanya Ingle et al. “Demand Forecasting : Literature Review On Various Methodologies”. In: *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. July 2021, pp. 1–7. DOI: 10.1109/ICCCNT51525.2021.9580139.

[8] Asher B. Curtis, Russell J. Lundholm, and Sarah E. Mcvay. “Forecasting Sales: A Model and Some Evidence from the Retail Industry”. en. In: *Contemporary Accounting Research* 31.2 (June 2014), pp. 581–608. DOI: 10.1111/1911-3846.12040.

[9] Saravanan Kesavan, Vishal Gaur, and Ananth Raman. “Do Inventory and Gross Margin Data Improve Sales Forecasts for U.S. Public Retailers?” In: *Management Science* 56.9 (2010). Publisher: INFORMS, pp. 1519–1533.

[10] Saumyadip Ghosh. “Forecasting of demand using ARIMA model”. In: *American Journal of Applied Mathematics and Computing* 1.2 (Apr. 2020), pp. 11–18. DOI: 10.15864/ajamc.124.

[11] Juliana C. Silva, Manuel Figueiredo, and A. C. Braga. “Demand forecasting: a case study in the food industry”. eng. In: Accepted: 2021-03-01T16:03:14Z ISSN: 0302-9743. Springer Verlag, 2019. DOI: 10.1007/978-3-030-24302-9_5.

[12] Yaohao Peng and Mateus Hiro Nagata. “An empirical overview of nonlinearity and overfitting in machine learning using COVID-19 data”. en. In: *Chaos, Solitons & Fractals* 139 (2020), p. 110055. DOI: 10.1016/j.chaos.2020.110055.

[13] Bing Xu and Jamal Ouenniche. "Performance evaluation of competing forecasting models: A multidimensional framework based on MCDA". en. In: *Expert Systems with Applications* 39.9 (2012), pp. 8312–8324.

[14] Mashael Khayyat et al. "Time Series Facebook Prophet Model and Python for COVID-19 Outbreak Prediction". English. In: *Computers, Materials, & Continua* 67.3 (2021). Num Pages: 3781-3793 Place: Henderson, United States Publisher: Tech Science Press Section: ARTICLE, pp. 3781–3793. DOI: http://dx.doi.org/10.32604/cmc.2021.014918.

[15] Frank Wilcoxon. "Individual Comparisons by Ranking Methods". In: *Biometrics Bulletin* 1.6 (1945). Publisher: [International Biometric Society, Wiley], pp. 80–83. DOI: 10.2307/3001968.

[16] Y Sakamoto, M Ishiguro, and G Kitagawa. *Akaike information criterion statistics*. English. OCLC: 13665112. Tokyo; Dordrecht; Boston; Hingham, MA: KTK Scientific Publishers ; D. Reidel ; Sold, distributed in the U.S.A., and Canada by Kluwer Academic Publishers, 1986.

[17] Denis Kwiatkowski et al. "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" en. In: *Journal of Econometrics* 54.1 (1992), pp. 159–178. DOI: 10.1016/0304-4076(92)90104-Y.

[18] Robin John Hyndman and George Athanasopoulos. "Forecasting: Principles and Practice". English. In: (2018). Publisher: OTexts.

[19] Rizwan Mushtaq. *Augmented Dickey Fuller Test*. en. SSRN Scholarly Paper ID 1911068. Rochester, NY: Social Science Research Network, Aug. 2011. DOI: 10.2139/ssrn.1911068.

[20] Quirin Stier, Tino Gehlert, and Michael C. Thrun. "Multiresolution Forecasting for Industrial Applications". en. In: *Processes* 9.10 (Oct. 2021). Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, p. 1697. DOI: 10.3390/pr9101697.

[21] Skipper Seabold and Josef Perktold. "Statsmodels: Econometric and Statistical Modeling with Python". en. In: Austin, Texas, 2010, pp. 92–96. DOI: 10.25080/Majora-92bf1922-011.

[22] Jr Everette S. Gardner and Ed Mckenzie. "Forecasting Trends in Time Series". In: *Management Science* 31.10 (1985). Publisher: INFORMS, pp. 1237–1246.

[23] Charles C. Holt. "Forecasting seasonals and trends by exponentially weighted moving averages". en. In: *International Journal of Forecasting* 20.1 (Jan. 2004), pp. 5–10. DOI: 10.1016/j.ijforecast.2003.09.015.

[24] Alysha De Livera, Rob Hyndman, and Ralph Snyder. "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing". In: *Journal of the American Statistical Association* 106 (Jan. 2010), pp. 1513–1527. DOI: 10.1198/jasa.2011.tm09771.

[25] Tommaso Proietti and Diego J. Pedregal. *Seasonality in High Frequency Time Series*. en. SSRN Scholarly Paper 3802611. Rochester, NY: Social Science Research Network, Mar. 2021. DOI: 10.2139/ssrn.3802611.

[26] Riaz Riazuddin and Mahmood-ul-Hasan Khan. "Detection and Forecasting of Islamic Calendar Effects in Time Series Data". en. In: 1.1 (2005), p. 10.

[27] Haiyan Song and Gang Li. "Tourism demand modelling and forecasting—A review of recent research". en. In: *Tourism Management* 29.2 (Apr. 2008), pp. 203–220. DOI: 10.1016/j.tourman.2007.07.016.

[28] Andrea Saayman and Ilse Botha. "Non-linear models for tourism demand forecasting". In: *Tourism Economics* 23 (2017), pp. 594–613. DOI: 10.5367/te.2015.0532.

[29] Julien Herzen et al. "Darts: User-Friendly Modern Machine Learning for Time Series". In: *arXiv:2110.03224 [cs, stat]* (Oct. 2021). arXiv: 2110.03224.

[30] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. "A Comparison of ARIMA and LSTM in Forecasting Time Series". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. Dec. 2018, pp. 1394–1401. DOI: 10.1109/ICMLA.2018.00227.

[31] G. Nunnari and V. Nunnari. "Forecasting Monthly Sales Retail Time Series: A Case Study". In: *2017 IEEE 19th Conference on Business Informatics (CBI)* (2017). DOI: 10.1109/CBI.2017.57.

[32] Balpreet Singh et al. "Sales Forecast for Amazon Sales with Time Series Modeling". In: *2020 First International Conference on Power, Control and Computing Technologies (ICPC2T)*. Jan. 2020, pp. 38–43. DOI: 10.1109/ICPC2T48082.2020.9071463.

[33] Sarah Gelper, Roland Fried, and Christophe Croux. "Robust forecasting with exponential and Holt–Winters smoothing". In: *Journal of Forecasting* 29.3 (Apr. 2010). Publisher: John Wiley & Sons, Inc., pp. 285–300. DOI: 10.1002/for.1125.

[34] Rob J Hyndman et al. "A state space framework for automatic forecasting using exponential smoothing methods". en. In: *International Journal of Forecasting* 18.3 (July 2002), pp. 439–454. DOI: 10.1016/S0169-2070(01)00110-8.

[35] Rob Hyndman et al. *Forecasting with Exponential Smoothing: The State Space Approach*. en. Google-Books-ID: GSyzox8Lu9YC. Springer Science & Business Media, June 2008.

[36] Rob J. Hyndman and Baki Billah. "Unmasking the Theta method". en. In: *International Journal of Forecasting* 19.2 (Apr. 2003), pp. 287–290. DOI: 10.1016/S0169-2070(01)00143-1.

[37] S. S. Wilks. "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses". In: *The Annals of Mathematical Statistics* 9.1 (Mar. 1938). Publisher: Institute of Mathematical Statistics, pp. 60–62. DOI: 10.1214/aoms/1177732360.

[38] Evangelos Spiliotis, Vassilios Assimakopoulos, and Spyros Makridakis. "Generalizing the Theta method for automatic forecasting". en. In: *European Journal of Operational Research* 284.2 (July 2020), pp. 550–558. DOI: 10.1016/j.ejor.2020.01.007.

[39] G. E. P. Box and D. R. Cox. "An Analysis of Transformations". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 26.2 (1964). Publisher: [Royal Statistical Society, Wiley], pp. 211–252.

[40] Sean J. Taylor and Benjamin Letham. *Forecasting at scale*. en. Tech. rep. e3190v2. ISSN: 2167-9843. PeerJ Inc., Sept. 2017. DOI: 10.7287/peerj.preprints.3190v2.

[41] A C Harvey and S Peters. "Estimation Procedures for Structural Time Series Models". en. In: 9.2 (1990), p. 21.

[42] Oskar Triebe et al. "NeuralProphet: Explainable Forecasting at Scale". In: *arXiv:2111.15397 [cs, stat]* (Nov. 2021). arXiv: 2111.15397.

[43] Guoqiang Zhang, B. Eddy Patuwo, and Michael Y. Hu. "Forecasting with artificial neural networks:: The state of the art". en. In: *International Journal of Forecasting* 14.1 (Mar. 1998), pp. 35–62. DOI: 10.1016/S0169-2070(97)00044-7.

[44] Anton Maximilian Schäfer and Hans-Georg Zimmermann. "Recurrent Neural Networks are universal approximators". eng. In: *International Journal of Neural Systems* 17.4 (Aug. 2007), pp. 253–263. DOI: 10.1142/S0129065707001111.

[45] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. "Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions". In: *International Journal of Forecasting* 37.1 (Jan. 2021). arXiv: 1909.00590, pp. 388–427. DOI: 10.1016/j.ijforecast.2020.06.008.

[46] Gábor Petneházi. "Recurrent Neural Networks for Time Series Forecasting". In: *arXiv:1901.00069 [cs, stat]* (Dec. 2018). arXiv: 1901.00069.

[47] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[48] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv:1406.1078 [cs, stat]* (Sept. 2014). arXiv: 1406.1078.

[49] Junyoung Chung et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". In: *arXiv:1412.3555 [cs]* (Dec. 2014). arXiv: 1412.3555.

[50] Leo Breiman. "Random Forests". en. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. DOI: 10.1023/A:1010933404324.

[51] Michael J. Kane et al. "Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks". English. In: *BMC Bioinformatics* 15 (2014). Num Pages: 276 Place: London, United Kingdom Publisher: BioMed Central, p. 276. DOI: http://dx.doi.org/10.1186/1471-2105-15-276.

[52] Karan Wanchoo. "Retail Demand Forecasting: a Comparison between Deep Neural Network and Gradient Boosting Method for Univariate Time Series". In: *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. Mar. 2019, pp. 1–5. DOI: 10.1109/I2CT45611.2019.9033651.

[53] Nijat Mehdiyev et al. "Evaluating Forecasting Methods by Considering Different Accuracy Measures". en. In: *Procedia Computer Science*. Complex Adaptive Systems Los Angeles, CA November 2-4, 2016 95 (Jan. 2016), pp. 264–271. DOI: 10.1016/j.procs.2016.09.332.

[54] Spyros Makridakis and Michèle Hibon. "The M3-Competition: results, conclusions and implications". en. In: *International Journal of Forecasting*. The M3- Competition 16.4 (2000), pp. 451–476. DOI: 10.1016/S0169-2070(00)00057-1.

[55] Teresa M. Mccarthy et al. "The evolution of sales forecasting management: a 20-year longitudinal study of forecasting practices". en. In: *Journal of Forecasting* 25.5 (2006). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/for.989, pp. 303–324. DOI: 10.1002/for.989.

[56] J. Scott Armstrong. "Evaluating Forecasting Methods". en. In: *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Ed. by J. Scott Armstrong. International Series in Operations Research & Management Science. Boston, MA: Springer US, 2001, pp. 443–472. DOI: 10.1007/978-0-306-47630-3_20.

[57] Rob J. Hyndman and Anne B. Koehler. "Another look at measures of forecast accuracy". en. In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688. DOI: 10.1016/j.ijforecast.2006.03.001.

[58] Diamantis Koutsandreas et al. "On the selection of forecasting accuracy measures". In: *Journal of the Operational Research Society* (Apr. 2021), pp. 1–18. DOI: 10.1080/01605682.2021.1892464.

[59] Yao Jin et al. "Forecasting With Temporally Aggregated Demand Signals in a Retail Supply Chain". In: *Journal of Business Logistics* 36 (2015). DOI: 10.1111/jbl.12091.

[60] Allan Valsaraj Mathai et al. "Development of new methods for measuring forecast error". In: *International Journal of Logistics Systems and Management* 24.2 (Jan. 2016). Publisher: Inderscience Publishers, pp. 213–225. DOI: 10.1504/IJLSM.2016.076472.

[61] Andrea Kolkova. "The Application of Forecasting Sales of Services to Increase Business Competitiveness". English. In: *Journal of Competitiveness* 12.2 (June 2020). Num Pages: 90–105 Place: Zlin, Czech Republic Publisher: Tomas Bata University in Zlin, Faculty of Management and Economics, pp. 90–105. DOI: http://dx.doi.org/10.7441/joc.2020.02.06.

[62] Karin Kandananond. "Forecasting Electricity Demand in Thailand with an Artificial Neural Network Approach". en. In: *Energies* 4.8 (Aug. 2011). Number: 8 Publisher: Molecular Diversity Preservation International, pp. 1246–1257. DOI: 10.3390/en4081246.

[63] Tae Kyun Kim. "T test as a parametric statistic". In: *Korean Journal of Anesthesiology* 68.6 (Dec. 2015), pp. 540–546. DOI: 10.4097/kjae.2015.68.6.540.

[64] João Henrique F. Flores, Paulo Martins Engel, and Rafael C. Pinto. "Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting". In: *The 2012 Interna-*

tional Joint Conference on Neural Networks (IJCNN). ISSN: 2161-4407. June 2012, pp. 1–8. DOI: 10.1109/IJCNN.2012.6252470.

[65] Alok Kumar Dwivedi, Indika Mallawaarachchi, and Luis A. Alvarado. "Analysis of small sample size studies using nonparametric bootstrap test with pooled resampling method: Nonparametric Bootstrap Test for Small Sample Size Studies". en. In: *Statistics in Medicine* (2017). DOI: 10.1002/sim.7263.

[66] *PyPI · The Python Package Index*. en.

[67] Markus Löning et al. "sktime: A Unified Interface for Machine Learning with Time Series". In: *arXiv:1909.07872 [cs, stat]* (Sept. 2019). arXiv: 1909.07872.

[68] Taylor G. Smith. *pmdarima*. original-date: 2017-03-30T14:58:30Z. Feb. 2022.

[69] François Chollet. *Deep learning with Python*. en. OCLC: ocn982650571. Shelter Island, New York: Manning Publications Co, 2018.

[70] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (Jan. 2012).

[71] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017.

[72] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2016). arXiv: 1603.02754, pp. 785–794. DOI: 10.1145/2939672.2939785.

[73] Colin Catlin. *AutoTS*. original-date: 2019-11-26T14:13:16Z. Apr. 2022.