Eindverslag

Sem Konings, Nico Pols en Lucas Ferwerda

Metis Montessori Lyceum

6 vwo, Frans Roukes

Inhoud

[Inleiding 2](#_Toc158955136)

[Opdrachtgever 3](#_Toc158955137)

[Experts 4](#_Toc158955138)

[Programma van eisen en wensen 4](#_Toc158955139)

[Situatieschets 5](#_Toc158955140)

[Opdracht 1 5](#_Toc158955141)

[Onderzoek 1 6](#_Toc158955142)

[Vormen van data-analyse 6](#_Toc158955143)

[Beschrijvende analyse 6](#_Toc158955144)

[Diagnostische analyse 7](#_Toc158955145)

[Voorspellende analyse 8](#_Toc158955146)

[Voorschrijvende analyse 9](#_Toc158955147)

[Verschillende datamodellen 10](#_Toc158955148)

[Relationeel 10](#_Toc158955149)

[Dimensioneel 11](#_Toc158955150)

[Entiteit-relatie 11](#_Toc158955151)

[Eindproduct 1 12](#_Toc158955152)

[Erlang formules 12](#_Toc158955153)

[Verschillende factoren en variabelen 13](#_Toc158955154)

[Opdracht 2 21](#_Toc158955155)

[Onderzoek 2 22](#_Toc158955156)

[Soorten modellen 22](#_Toc158955157)

[Queuing Theory-modellen 22](#_Toc158955158)

[Simulatie software 23](#_Toc158955159)

[WFM-software 24](#_Toc158955160)

[Machine Learning-modellen 25](#_Toc158955161)

[Programmeertalen 26](#_Toc158955162)

[Python 26](#_Toc158955163)

[Javascript 26](#_Toc158955164)

[C++ 26](#_Toc158955165)

[SQL 27](#_Toc158955166)

[C# 27](#_Toc158955167)

[Algemene regels van programmeren 27](#_Toc158955168)

[ChatGPT als programmeur 28](#_Toc158955169)

[Eindproduct 2 29](#_Toc158955170)

[Modelcode 29](#_Toc158955171)

[Werkend model 36](#_Toc158955172)

[Presentatie 37](#_Toc158955173)

[Reflectie en evaluatie 38](#_Toc158955174)

[Bronvermelding 39](#_Toc158955175)

# Inleiding

Dit onderzoeksverslag biedt een overzicht van ontwikkelingen en bevindingen op een breed scala aan onderzoeksgebieden. Dit verslag zal een weergave geven van enkele van de meest opvallende en belangrijke onderzoeksresultaten, evenals de trends en uitdagingen die de verschillende vakgebieden beïnvloeden. Door deze verkenning komen we terecht bij de uiteindelijke eindproducten.

Dit verslag dient ook als een reflectie op een afgerond onderzoeksproject dat gericht was op analyseren en modelleren. Gedurende het afgelopen half jaar hebben we ons toegewijd aan het verkennen, analyseren en interpreteren van gegevens met als doel een kwalitatief eindresultaat op te leveren.

Bovendien zullen we reflecteren op de uitdagingen, leermomenten en mogelijke toekomstige richtingen die voortvloeien uit ons onderzoek. Deze reflectie is bedoeld om inzicht te geven in het proces van onderzoek doen en de impact ervan op ons begrip van het onderwerp.

# Opdrachtgever

De opdrachtgever voor dit project is het bedrijf ContactMakers. Dit bedrijf is gespecialiseerd op het gebied van telecommunicatie voor bedrijven en is met twintig jaar ervaring zeer gerespecteerd binnen de Nederlandse telecomsector. Kort samengevat verzorgen zij de verbinding tussen consumenten en bedrijven door het communicatieve deel van bedrijven in te richten en te onderhouden (*ContactMakers.nl*, 2022).

Een groot onderdeel van het bedrijf is het zelfontwikkelde platform ContactSuite. Dit platform maakt het mogelijk voor bedrijven om gemakkelijk op één plek inkomende en uitgaande oproepen te beheren, gesprekken terug te luisteren, statistieken in te zien, overzicht te houden en meer (*ContactMakers.nl*, 2022).

Uit een overzicht van de klanten op de website van ContactMakers wordt duidelijk dat veel bekende namen zijn aangesloten en klant zijn bij ContactMakers. Denk hierbij aan Coolblue, Vodafone Ziggo, Tommy Teleshopping of HelloFresh (*ContactMakers.nl*, 2022).

Het bedrijf bevindt zich op het adres Veluwezoom 42, 1327 AH Almere. Er zijn meerdere manieren waarop je het bedrijf kunt bereiken. Zo kun je de chatfunctie van hun site gebruiken, telefonisch contact opnemen met één van de medewerkers, mailen of een afspraak maken op locatie (*ContactMakers.nl*, 2022).

De contactpersoon binnen dit bedrijf is Denise Konings. Zij is de algemeen directeur van dit bedrijf. Door een korte blik te werpen op haar LinkedIn wordt duidelijk dat zij al heel wat jaren meedraait in de telecomsector. In 1991 heeft zij de opleiding techniek en ondernemerschap aan de MTS Techniek en technologie afgerond om vervolgens een aantal jaar later meerdere eigen merken in de telecom tot een succes te brengen. Zo is zij niet alleen algemeen directeur van ContactMakers, maar ook van Beleasy en Massxess (*LinkedIn Denise Konings*, 2023). Zij is persoonlijk te bereiken via de mail, telefoon of in persoon. Onderstaand zijn de belangrijkste contactgegevens toegevoegd.

Bedrijfsmail: [dkonings@contactmakers.nl](mailto:dkonings@contactmakers.nl)

Bedrijfstelefoon: 088-5454081



# Experts

Bij dit project zijn er maar liefst twee soorten experts betrokken; het bedrijf ContactMakers zelf en twee docenten aan de Hogeschool van Amsterdam

De primaire experts Jan Kes en Linda schouten, beiden docent in applied sciences aan de faculteit wiskunde. Deze studie focust zich vooral op onderwerpen zoals algebra, programmeren, financiële wiskunde en statistiek en gaat verdiepend in op modules zoals Operations Research en Machine Learning (HvA, 2023). Deze studie sluit naadloos aan op dit project en maakt Jan Kes en Linda Schouten met ruim zeven jaar ervaring dé aangewezen experts. Zo kunnen zij helpen bij het analyseren van gevonden data en het efficiënt opstellen van een model.

Naast de eerder benoemde experts heeft mevrouw Konings aangegeven dat haar bedrijf ook zal fungeren als expert. ContactMakers zal bepaalde data leveren waarover zij beschikken om ons te helpen met het maken van een model. Haar data in combinatie met de experts van de HvA zal dan ook een grote hulp zijn gedurende dit project. Gedurende het verloop van dit project heeft nog een persoon zich gemeld als expert: meneer Remco Verhulst, programmeur bij ContactMakers. Deze expert helpt onder andere bij het verwerken van de gegevens en het maken van een toepasbaar model.

# Programma van eisen en wensen

Voor dit project zijn er vanuit de opdrachtgever nauwelijks eisen gesteld.    
Dit komt voornamelijk doordat deze opdracht één einddoel heeft waarin weinig ruimte voor interpretatie is. De opdracht is dan ook heel duidelijk geformuleerd. De enige eis die gesteld wordt, is dat het model rekening houdt met speciale factoren en variabelen die blijken uit de gegeven en gevonden datasets. Naast deze eis is er nog de wens om het eindproduct vóór 1 januari op te leveren. Mocht dit niet haalbaar zijn, dan is dat geen probleem. Een duidelijk overzicht van de eisen en wensen is:

* Het model moet rekening houden met factoren en variabelen
* Het eindproduct moet voor 1 januari opgeleverd worden
* Het eindproduct moet een model zijn die vertaalbaar is naar software
* Invoer van gegevens in het eindproduct moet leiden tot een voorspelling

# Situatieschets

In veel verschillende sectoren besteden bedrijven hun klantencontact uit door callcenters van externe partijen in te schakelen. Deze callcenters nemen de gesprekken aan en handelen deze in de meeste gevallen af    
(Call Factory Ltd., z.d.). In een gesprek met mevrouw Konings gaf zij aan dat de hoeveelheid telefoongesprekken die een bedrijf per dag ontvangt verschilt per sector. Logischerwijs kan de hoeveelheid telefoongesprekken ook per seizoen, per dag of zelfs per uur verschillen. Uit haar data blijkt bijvoorbeeld dat veel bedrijven en instanties vaak rond het begin en einde van de maand veel gesprekken ontvangen (D. Konings, 3 oktober, 2023).

Mevrouw Konings heeft vastgesteld dat veel bedrijven door deze fluctuatie problemen hebben met het vinden van de juiste bezetting. Dat houdt in dat er op momenten te veel of te weinig mensen op het callcenter aanwezig zijn die telefoongesprekken aannemen. Daaraan voegt zij toe: “Dit probleem gaat ten koste van de servicekwaliteit en kost in de meeste gevallen onnodig geld.”    
(D. Konings, 3 oktober, 2023) Als het probleem concreet geformuleerd moet worden, dan kan dat als volgt: bedrijven hebben door inschattingsfouten op momenten te veel of te weinig bezetting die inkomende gesprekken aanneemt.

# Opdracht 1

Uiteindelijk is het de bedoeling dat er een oplossing komt voor de eerder benoemde probleemstelling. Mevrouw Konings heeft aangegeven dat zij dit wil oplossen door zelf een tool voor haar klanten te ontwikkelen die een voorspelling doet van de juiste bezetting voor de gevraagde periode. Voordat zij een dergelijke tool kan ontwikkelen heeft zij een passend model of formule nodig gegevens leidt tot een voorspelling. Het is aan ons om dit model/deze formule te vinden. Naast het vinden van dit model moeten wij een gegeven dataset bestuderen en hier alle verschillende factoren en variabelen van berekenen. Voorbeelden hiervan zijn het verschil in bellers op maandagen ten opzichte van dinsdagen, de invloeden van feestdagen, gemiddeldes, etc.

Een concrete formulering van de opdracht zou het volgende kunnen zijn:    
het vinden van een (wiskundig) model, die rekening houdt met speciale factoren en variabelen, waarbij de invoer van gegevens leidt tot een nauwkeurige voorspelling van de juiste callcenter bezetting en het analyseren van een gegeven dataset.

# Onderzoek 1

## Vormen van data-analyse

### Beschrijvende analyse

Een belangrijk onderdeel in het proces van data-analyse is de beschrijvende analyse. Hierbij wordt gekeken naar historische data en wat er in verleden is gebeurd op vergelijkbare momenten. Het doel is om patronen, trends en belangrijke kenmerken in de gegevens te achterhalen. Een beschrijvende analyse omvat vaak eenvoudige statistieken en visualisatietechnieken om informatie gemakkelijk toegankelijk en begrijpelijk te maken. Hierdoor kunnen statistische samenvattingen worden gemaakt die ervoor moeten zorgen dat de informatie vanuit het verleden duidelijk en overzichtelijk wordt weergegeven. Voor het gebruik van basisstatistieken zoals gemiddelde, mediaan, modus, standaarddeviatie, en percentielen om een overzicht te krijgen van de centrale tendensen en de spreiding van de gegevens. Het gebruik van data-visualisatie door middel van grafieken, diagrammen en andere visuele hulpmiddelen zijn patronen en trends in de gegevens makkelijker identificeren. Deze informatie kan dan met behulp van ‘dashboarding’ worden gebruikt voor het creëren van dashboards waarop de belangrijkste prestatie-indicatoren (KPI's) en andere relevante informatie worden weergegeven, deze dashboards moeten ervoor zorgen dat er juist geanticipeerd kan worden op verschillende situaties, zoals bij een callcenter. Ook kan je zulke informatie beschouwen door middel van data-segmentatie. Hierbij worden de gegevens verdeeld in verschillende categorieën, om variaties en patronen binnen specifieke groepen te identificeren. Dit kan helpen bij het begrijpen van verschillen tussen verschillende producten, regio’s en klantsegmenten, waar een callcenter rekening mee moet houden.

Beschrijvende analyse geeft een duidelijk inzicht in het verleden. Dit helpt organisaties om historische patronen en trends te begrijpen en geeft ook direct de basis voor besluitvormingen. Samenvattingen van historische gegevens stelt organisaties namelijk in staat om beter geïnformeerde beslissingen te nemen op basis van feitelijke informatie en doet dit op een vrij eenvoudige manier. Hierdoor is het voor iedereen gemakkelijk te begrijpen wat de mogelijke problemen zijn voor in de toekomst, aangezien zulke factoren meerdere keren kunnen terug keren.

Een nadeel van een beschrijvende analyse is dat je beperkt bent aan de historische gegevens, dat is het enige dat geanalyseerd kan worden. Het biedt daarmee geen zekerheid voor in de toekomst, maar wel is het een gegronde voorspelling. Een ander nadeel is dat het slechts een algemeen beeld van de gegevens beidt. Hierdoor is het eenvoudig te begrijpen, maar mist het dus vaak diepgaande analyse die je nodig hebt om complexere oorzaken en verbanden te begrijpen en op te lossen. Hierdoor is men niet direct geneigd om actie te ondernemen aangezien men niet weet of alle informatie toepasbaar is voor het heden. Ook kan de informatie veroudert kan zijn.

### Diagnostische analyse

Een diagnostische analyse is een geavanceerder onderdeel van een data-analyse die zich verdiept in het begrijpen van de onderliggende oorzaken en factoren die hebben bijgedragen aan specifieke trends of gebeurtenissen in het verleden. Waar beschrijvende analyse zich bezighoudt met het beschrijven en samenvatten van historische gegevens, gaat diagnostische analyse verder door te onderzoeken waarom bepaalde resultaten zijn opgetreden. Het streeft ernaar antwoord te geven op de cruciale vraag: "Waarom is dit gebeurd?"

Een kenmerkende eigenschap van een diagnostische analyse is het gebruik van geavanceerde statistische methoden en technieken om niet alleen patronen en trends te identificeren, maar ook om oorzakelijke verbanden tussen verschillende variabelen te ontdekken. Hierbij komen statistische instrumenten zoals regressieanalyse, correlatieanalyse en andere geavanceerde modelleringstechnieken vaak aan bod.

Een belangrijk aspect van diagnostische analyse is de identificatie van de diepere oorzaken doormiddel van een root cause analysis, waarbij onderzoekers proberen de fundamenten van problemen te begrijpen. Dit kan variëren van technische aspecten tot menselijke factoren en organisatorische processen. Naast kwantitatieve methoden omvat diagnostische analyse vaak kwalitatieve onderzoekstechnieken. Diepte-interviews, focusgroepen en andere methoden voor kwalitatieve datacollectie worden ingezet om een dieper begrip te krijgen van de menselijke aspecten en context achter de kwantitatieve gegevens.

Het doel van een diagnostische analyse is niet alleen om het verleden te verklaren, maar ook om organisaties te voorzien van inzichten die hen in staat stellen om gerichte acties te ondernemen voor de toekomst. Door oorzakelijke verbanden en diepere oorzaken bloot te leggen, helpt diagnostische analyse bij het nemen van weloverwogen beslissingen en het implementeren van effectieve strategieën om herhaling van problemen te voorkomen. In essentie biedt diagnostische analyse een dieper begrip van wat er is gebeurd en waarom, waardoor organisaties proactief kunnen handelen om betere resultaten in de toekomst te bereiken.

### Voorspellende analyse

Nog een fase in een analyseproces is de voorspellende analyse. Dit soort analyses richten zich op het gebruik van historische gegevens en statistische algoritmen om toekomstige gebeurtenissen, trends en resultaten te voorspellen. In tegenstelling tot beschrijvende analyse, dat zich bezighoudt met het beschrijven van wat er in het verleden is gebeurd en diagnostische analyse, dat zich richt op het begrijpen waarom bepaalde gebeurtenissen hebben plaatsgevonden, kijkt voorspellende analyse vooruit en anticipeert op wat er waarschijnlijk zal gebeuren.

Een sleutelelement van voorspellende analyse is het gebruik van wiskundige modellen en algoritmen om patronen in historische gegevens te identificeren. Deze modellen kunnen variëren van eenvoudige regressiemodellen tot complexere machine learning-algoritmen, afhankelijk van de aard van de gegevens en het doel van de voorspellingen. Het doel is vaak om trends te begrijpen en te generaliseren, waardoor organisaties in staat worden gesteld toekomstige gebeurtenissen beter te voorspellen.

Organisaties maken gebruik van voorspellende analyse in verschillende domeinen, zoals financiën, marketing, gezondheidszorg en supply chain management. Bijvoorbeeld, in de financiële sector kan voorspellende analyse worden ingezet om het kredietrisico te beoordelen door patronen in het betaalgedrag van klanten te analyseren. In marketing kan het worden gebruikt om klantgedrag te voorspellen en gerichte marketingcampagnes te ontwikkelen.

Een belangrijk kenmerk van voorspellende analyse is het vermogen om proactief te handelen op basis van voorspellingen. Organisaties kunnen strategieën ontwikkelen en beslissingen nemen om zich voor te bereiden op mogelijke uitkomsten. Dit kan variëren van voorraadoptimalisatie en risicobeheer tot het personaliseren van klantinteracties om aan toekomstige behoeften te voldoen.

Een probleem van voorspellende analyse is het omgaan met onzekerheid. Voorspellingen zijn gebaseerd op historische gegevens en veronderstellingen, onverwachte gebeurtenissen kunnen hierbij de nauwkeurigheid beïnvloeden. Het is essentieel om modellen voortdurend te evalueren en aan te passen aan veranderende omstandigheden (Google Cloud, 2023).

### Voorschrijvende analyse

Voorschrijvende analyse vertegenwoordigt het hoogste niveau van gegevensanalyse waarbij geavanceerde technologieën en modellen worden gebruikt om niet alleen te voorspellen wat er waarschijnlijk zal gebeuren, maar ook om aanbevelingen te doen over de beste acties die moeten worden ondernomen om een specifiek doel te bereiken. In wezen gaat voorschrijvende analyse verder dan het identificeren van trends en het voorspellen van toekomstige gebeurtenissen, door strategische richtlijnen te bieden voor besluitvorming.

Een belangrijk kenmerk van voorschrijvende analyse is het gebruik van optimalisatie- en simulatiemodellen. Deze modellen evalueren verschillende mogelijke scenario's en suggereren de meest effectieve beslissingen op basis van vooraf gedefinieerde doelstellingen en beperkingen. Hierdoor kunnen organisaties niet alleen anticiperen op toekomstige gebeurtenissen, maar ook actieplannen ontwikkelen om de gewenste uitkomsten te bereiken.

Voorschrijvende analyse kan op diverse gebieden worden toegepast, zoals supply chain management, gezondheidszorg, financiën en operations. Bijvoorbeeld, in supply chain management kan het worden gebruikt om de optimale voorraadniveaus te bepalen en efficiënte distributieroutes te plannen. In de gezondheidszorg kan het helpen bij het ontwikkelen van behandelingsplannen op maat voor patiënten op basis van hun medische geschiedenis en genetische informatie.

Een ander aspect van voorschrijvende analyse is de real-time aanpassing aan veranderende omstandigheden. Door gebruik te maken van geavanceerde algoritmen kan het voortdurend evalueren en bijwerken van aanbevelingen op basis van nieuwe gegevens en variabelen. Dit maakt het een waardevol instrument in een snel veranderende zakelijke omgeving.

Hoewel voorschrijvende analyse krachtige inzichten en aanbevelingen biedt, zijn er uitdagingen zoals de complexiteit van modellering en de noodzaak van nauwkeurige gegevens. Organisaties moeten ook ethische overwegingen in acht nemen, vooral wanneer het gaat om gevoelige gebieden zoals gezondheidszorg en financiën.

In essentie stelt voorschrijvende analyse organisaties in staat om niet alleen te begrijpen wat er waarschijnlijk zal gebeuren, maar ook om doelgericht actie te ondernemen om de best mogelijke uitkomsten te bereiken. Het vormt de meest geavanceerde fase van data-analyse en speelt een cruciale rol in het informeren van strategische besluitvorming op alle niveaus van een organisatie.

## Verschillende datamodellen

### Relationeel

Veel informatie kan met behulp van een relationeel databeheersysteem op twee manieren worden ingedeeld. Hierbij wordt onderscheid gemaakt tussen dimensionale informatie en informatie in de vorm van meetwaardes. Dimensionale informatie is het gebruik van locaties en beschrijvingen van objecten of producten. Informatie in de vorm van meetwaardes zijn numerieke waarden, zoals hoeveelheden en inkomsten die gebruikt worden in de wiskundige berekeningen van een model (Sap, 2023).

De informatie in de vorm van meetwaardes wordt door het databeheersysteem opgeslagen in tabellen om onderlinge verbanden te leggen, zodat hiermee een model gebouwd kan worden. Deze indeling wordt gemaakt aan de hand van integriteitsregels om te voorkomen dat tabellen overbelast worden en/of foutieve informatie geven. (Sap, 2023). De modellen die hieruit voortkomen uit een relationeel databeheersysteem model wordt door veel verschillende organisaties gebruikt om een breder beeld te krijgen van hun informatie. Zo wordt het gebruikt om inventarissen bij te houden, e-commercetransacties te verwerken, enorme hoeveelheden cruciale klantinformatie te beheren en nog veel meer. De betrouwbaarheid en het eenvoudige gebruik van relationele databeheersystemen maakt dat het in de telecomsector haast door elk bedrijf wordt gebruikt. Ook bij grote hoeveelheden data kunnen de modellen optimaal functioneren   
(Microsoft, 2023). Onderstaand staat een voorbeeld van een relationeel datamodel uitgewerkt.

{

‘’EmployeesID’’: ‘’1’’,  
‘’Name’’: ‘’Tom’’,  
‘’Location’’: ‘’Germany’’

},

{

‘’EmployeesID’’: ‘’2’’,  
‘’Name’’: ‘’Jake’’,  
‘’Location’’: ‘’United States’’

}

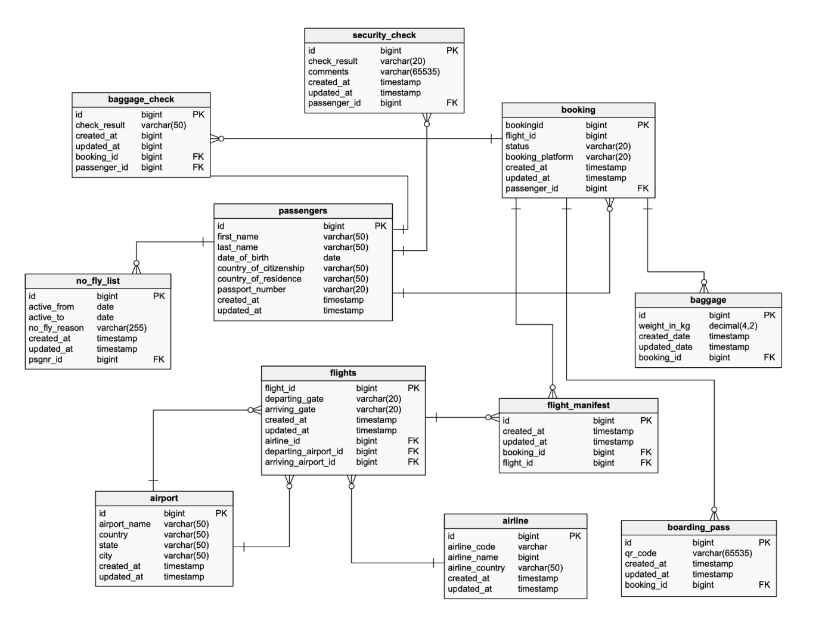
‘’EmployeesID’’: ‘’2’’,  
‘’Name’’: ‘’Jake’’,  
‘’Location’’: ‘’United States’’

### Dimensioneel

Naast de modellen die voortkomen uit meetwaardes, zijn er ook modellen die voortkomen uit dimensionale data. Deze datamodellen zijn minder rigide, maar wel gestructureerd. Dit bevordert een contextuele datastructuur die meer gerelateerd is aan het zakelijk gebruik ervan. Kritieke data-elementen, zoals een transactiehoeveelheid, worden "feiten" genoemd en gaan hand in hand met referentie-informatie, oftewel "dimensies". Dit is het product-ID, de eenheidsprijs of de transactiedatum. Een feitentabel is de primaire tabel in een dimensionaal model. Data kan snel en efficiënt worden opgehaald, waarbij data voor een bepaald type activiteit samen worden opgeslagen, maar het ontbreken van relatiekoppelingen kan het analytisch ophalen en gebruiken van de data bemoeilijken. Aangezien de datastructuur is gekoppeld aan de bedrijfsfunctie die de data produceert en gebruikt, kan het combineren van data die afkomstig zijn van verschillende systemen (bijvoorbeeld in een datawarehouse) problemen opleveren.

### Entiteit-relatie

Een E-R-model (zie figuur 2 en 3) vertegenwoordigt de datastructuur van een bedrijf in een grafische vorm. Hierbij worden verschillende categorieën die met elkaar in relatie staan weergegeven in modelvorm met vakken van verschillende vormen om de activiteit, functie of "entiteit" weer te geven en lijnen om associaties, afhankelijkheden of "relaties" weer te geven. Met het E-R-model wordt vervolgens een relationele database gemaakt waarin elke rij een entiteit voorstelt en de velden in die rij attributen bevatten. Net als in alle relationele databases worden belangrijke data-elementen gebruikt om tabellen aan elkaar te koppelen (Ludichart, 2023).

Afbeelding met tekst, schermopname, ontvangst, Lettertype

Automatisch gegenereerde beschrijving

Figuur 1 en 2: voorbeeld van E-R-modellen

# Eindproduct 1

## Erlang formules

De Erlang (E) is een eenheid voor de omvang van het telefoonverkeer. Het Consultative Committee for International Telegraphy and Telephony (CCITT) nam in 1946 de Erlang als internationale eenheid voor het telefoonverkeer aan. Het was aan de hand van het wetenschappelijk werk, uitgevoerd door de Deense wiskundige Agner Erlang, in verband met wachtrijtheorie, dat men tot de definitie van de Erlang kwam.

Een Erlang betekent dat één telefoonlijn elk uur een uur aan gesprekstijd heeft. Dit kan zijn: één telefoongesprek van één uur per uur, of twee gesprekken van een half uur per uur, enzovoort. Een gemiddeld gesprek kan bijvoorbeeld op 2 minuten vastgesteld worden wat bij één gesprek per uur gelijk is aan 2 van de 60 minuten in een uur. De Erlang van de telefoonlijn is dan E. Een analoge telefoonlijn kan maximaal één Erlang verwerken, ze wordt dan 100% gebruikt. De algemene formule voor de eenheid Erlang is:

Vaak heeft niet elke telefoonlijn binnen een callcenter een gelijke Erlang. Voor het totaal moet de Erlang van elke telefoonlijn apart berekend worden en vervolgens bij elkaar opgeteld worden. De formule hiervoor is:

Etot = E0 + E1 + E2 + … + En

Vaak voor de toepassing van de Erlang in formules wordt echter de gemiddelde Erlang gebruikt. Hiervoor geldt:

Egem = Etot / n

De Erlang wordt veel gebruikt om te bepalen of een telefooncentrale een te grote of te kleine capaciteit heeft (interne en externe oproepen). Er wordt gesteld dat het niet slagen van één oproep op elke 20, ofwel 5%, voor de meeste toepassingen aanvaardbaar is. Hiervoor wordt de Erlang B formule gebruikt:

‘P’ is hierbij het deel van het totaal aantal telefoongesprekken dat niet aangenomen kan worden. Zo is 0,05 gelijk aan 5% van het totaal en is 0,2 gelijk aan 20% van het totaal. ‘m’ is het aantal telefoonlijnen dat draait met dezelfde Erlang. Omdat de gemiddelde Erlang (Egem) voor elke lijn toepasbaar is, hoef je enkel het aantal telefoonlijnen voor ‘m’ in te vullen bij het gebruik van Egem.

Met de Erlang C formule bereken je de vermoedelijke wachttijden voor bellers op basis van drie uitgangspunten: het aantal agents, het aantal wachtenden, en de gemiddelde gesprekstijd per gesprek. De formule geeft ook aan hoeveel agents moeten worden ingezet om de wachttijden binnen vastgestelde grenzen te houden. Hiermee wordt de link gelegd met servicelevel. Omdat de Erlang C formule geen rekening houdt met verlaten gesprekken en een eventuele bezettoon (of een mededeling dat de lijn is overbelast), neigt de formule naar overbezetting als uitkomst.

‘Pw’ is hierbij de kans dat een beller in een wachtrij belandt, ‘A’ is gelijk aan het Etot van een callcenter en N is het aantal beschikbare agents. In een ideale wereld is Pw gelijk aan nul. Echter wordt er vaak een marge gehanteerd die verschilt per sector. Zo mag de marge bij nooddiensten vrijwel nul zijn, maar die van een bedrijf die producten verkoopt hoger liggen. Gemiddeld wordt voor niet noodzakelijke lijnen een marge van 5% gehanteerd

## Verschillende factoren en variabelen

Verkregen dataset:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ma | Di | Wo | Do | Vr |
| Totale jaarlijkse gespreksduur in minuten | 358725:43 | 262923:54 | 244772:54 | 210591:48 | 232513:33 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ma | Di | Wo | Do | Vr |
| Gemiddelde gespreksduur per dag in minuten | 6898:34 | 5056:13 | 4707:51 | 4049:51 | 4471:25 |

Uit grafiek en tabel 1 en 2 zijn een aantal dingen te concluderen. Het valt op dat de maandag over het algemeen verreweg de drukste dag is. Het is dus logisch dat er op maandagen meer agents ingeschakeld moeten worden. Een mogelijke verklaring hiervoor is dat er op zaterdagen en zondagen niet gebeld kan worden. Veel klanten zullen dan hun vragen bewaren tot de eerst mogelijke dag om te bellen, maandag.

Uit deze berekening blijkt dat er op de gemiddelde maandag maar liefst 70% meer wordt gebeld dan op de gemiddelde donderdag.

Gemiddeld wordt er per werkdag 5036 minuten gebeld:

Afwijking van de gemiddelde gespreksduur op maandag ten opzichte van de gemiddelde gespreksduur per werkdag:

Afwijking van de gemiddelde gespreksduur op dinsdag ten opzichte van de gemiddelde gespreksduur per werkdag:

Afwijking van de gemiddelde gespreksduur op woensdag ten opzichte van de gemiddelde gespreksduur per werkdag:

Afwijking van de gemiddelde gespreksduur op donderdag ten opzichte van de gemiddelde gespreksduur per werkdag:

Afwijking van de gemiddelde gespreksduur op vrijdag ten opzichte van de gemiddelde gespreksduur per werkdag:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Okt | Nov | Dec | Jan | Feb | Mrt |
| Totale maandelijkse gespreksduur in minuten | 32366:02 | 40596:27 | 32870:07 | 38349:06 | 32512:21 | 37279:03 |
| Apr | Mei | Jun | Jul | Aug | Sep |
| 30544:37 | 35755:43 | 37370:47 | 36979:11 | 35219:51 | 36337:09 |

In de periode tussen oktober en april is een patroon te zien waarbij de gespreksduur om de maand steeds van hoog naar laag wisselt. Dit patroon herhaalt zich vier keer voordat het in mei weer vrij constant wordt. De periode tussen mei en september heeft dus een relatief constante totale bezetting nodig. Ook valt op dat de maand november het meeste aantal belminuten heeft en april het minste aantal belminuten.

Gemiddeld wordt er per maand 33514 minuten gebeld:

Afwijking van de gemiddelde gespreksduur in oktober ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in november ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in december ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in januari ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in februari ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in maart ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in april ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in mei ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in juni ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in juli ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in augustus ten opzichte van de gemiddelde gespreksduur per maand:

Afwijking van de gemiddelde gespreksduur in september ten opzichte van de gemiddelde gespreksduur per maand:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Zomer | Herfst | Winter | Lente |
| Totale gespreksduur  per seizoen in minuten | 317228:41 | 404088:20 | 360686:32 | 321141:23 |

Herfst t.o.v. zomer:

Aan de hand van deze berekening is te zien dat er 27% meer gesprekstijd is in het herfstseizoen ten opzichte van het zomerseizoen. Met eenzelfde soort berekening kunnen we deze verhouding ook voor de rest van de seizoenen berekenen.

Herfst t.o.v. winter

Herfst t.o.v. lente

Zomer t.o.v. winter

Zomer t.o.v. lente

Winter t.o.v. lente

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gespreksduur per eerste maandag van de maand in minuten | Okt | Nov | Dec | Jan | Feb | Mrt |
| 7407:47 | 8123:20 | 7406:39 | 10999:33 | 8677:39 | 7219:44 |
| Apr | Mei | Jun | Jul | Aug | Sep |
| 5699:58 | 10813:43 | 7436:12 | 6310:30 | 6195:03 | 5250:18 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gespreksduur per eerste maandag van de maand in minuten | Okt | Nov | Dec | Jan | Feb | Mrt |
| 7407:47 | 4592:28 | 4116:11 | 10999:33 | 5652:33 | 3428:44 |
| Apr | Mei | Jun | Jul | Aug | Sep |
| 5699:58 | 10813:43 | 6728:56 | 6310:30 | 3911:35 | 4928:23 |

Met deze grafieken kunnen we de verschillende maanden wederom vergelijken in gespreksduur, maar hiermee kunnen we ook zien of het uitmaakt of de eerste werkdag van de maand een maandag is of niet. Dit is onderzoek-waardig, omdat we verwachten dat de eerste maandag van de maand een relatief hoge gespreksduur zal hebben ten opzichte van andere dag soorten die de eerste werkdag van de maand zijn.

Als wordt gekeken naar beide grafieken dan blijkt dat de maandagen inderdaad de drukste eerste werkdagen zijn. Wat wel frappant is, is het feit dat de maandag in april relatief rustig is. De reden hierachter is tot nog toe onbekend.

De gemiddelde gespreksduur van alle werkdagen in het jaar bleek uit een eerdere berekening 5036 minuten per dag. De gemiddelde gespreksduur van de eerste werkdagen van de maand is 6215 minuten:

Uit deze berekening blijkt dat ‘de eerste van de maand’ een positieve invloed heeft op het aantal belminuten. Er wordt namelijk ruim 23% meer gebeld dan op een gemiddelde werkdag. Dit kan echter ook komen door de maandagen. Daarom wordt nu ook gekeken naar de invloed van ‘de eerste van de maand’ zonder de maandagen.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gespreksduur per eerste maandag van de maand in minuten | Nov | Dec | Feb | Mrt | Mei | Jul | Aug |
| 4592:28 | 4116:11 | 5652:33 | 3428:44 | 6728:56 | 3911:35 | 4928:23 |

De gemiddelde gespreksduur van de eerste werkdagen van de maand zonder maandagen is 4765 minuten:

Aan de twee vergelijkingen hierboven is te zien dat het een groot verschil maakt als de maandagen niet meegerekend worden. Zonder de maandagen ligt het gemiddeld aantal belminuten van de ‘eersten van de maanden’ dus 5% lager dan het gemiddelde van een doodnormale dag.

In de bovenstaande grafiek is nog een vergelijking weergegeven tussen de eerste maandagen en eerste werkdagen van de maand. De balken die zwart gekleurd zijn hebben als eerste werkdag van de maand een maandag. Het blijkt dat de eerste maandag van de maand altijd een hogere gespreksduur heeft dan de eerste werkdag van de maand. Maandagen hebben dus meer invloed dan de eerste werkdag van de maand. De positieve invloed van een maandag overtreft dus de positieve invloed van de eerste werkdag van de maand

# Opdracht 2

Aan het begin van dit project was het enkel de bedoeling dat wij een passend model zouden vinden voor de tool die mevrouw Konings wil maken en dat we de door haar toegezonden dataset uit zouden pluizen, zoals beschreven in ‘opdracht 1’ en het projectplan. Een concrete eis daarbij was dat dit voor 1 januari klaar moest zijn. Wij kunnen met trots zeggen dat dit ons is gelukt. Nog trotser zijn wij op het volgende: nadat wij onze resultaten hadden ingediend bij de opdrachtgeefster en met haar evalueerde over het resultaat vroeg zij ons of wij zin hadden in ‘nog een rondje’. Gezien de tijd die nog resteerde tot de presentaties en de kwaliteit die wij in zo een korte periode hebben geleverd wilde de opdrachtgeefster namelijk wel met ons verder. Na kort overleg binnen het team en het aanhoren van de vervolgopdracht gingen wij akkoord.

Voor de desbetreffende vervolgopdracht werd aan ons gevraagd om in samenwerking met een programmeur van ContactMakers, Remco Verhulst, onze gevonden formule (de Erlang C formule) in combinatie met de uitgewerkte dataset te verwerken in een werkende tool. Deze tool wordt vervolgens aangesloten op het datanetwerk van ContactMakers zodat de tool live gebruikt kan worden door het gehele klantenbestand (vet gaaf!).

# Onderzoek 2

## Soorten modellen

### Queuing Theory-modellen

De Queuing Theorie geeft een wiskundige schatting wat er gebeurt wanneer er mensen in een rij bevinden en wachten om bediend te worden. Je kan dit vergelijken met een rij die staat bij de kassa van een supermarkt; dit alledaagse scenario illustreert perfect de kernprincipes van Queuing Theorie.

Aankomsten en servicekwaliteit zijn twee essentiële concepten in dit kader. Aankomsten verwijzen naar hoe vaak nieuwe klanten zich bij voegen, terwijl servicekwaliteit aangeeft hoe snel de mensen worden bediend. Dit is vergelijkbaar met het ritme en de werksnelheid van een medewerker. Deze dynamiek vormt de basis voor het begrijpen van wachtrijen.

Queuing Theorie houdt rekening met diverse grootheden, waaronder de hoeveelheid tijd die gemiddeld verstrijkt voordat een nieuwe klant in de rij staat. Daarnaast wordt er gekeken naar de servicetijd, de tijd die een persoon gemiddeld besteedt aan het ontvangen van de service. Het aantal beschikbare servers is ook van cruciaal belang. Wachtrijlengte en wachttijd zijn concepten die voorspellen hoeveel mensen er gemiddeld in de wachtrij staan en hoe lang mensen in de rij moeten wachten. Hier draait het om het begrijpen van de dynamiek van een wachtrij, wat essentieel is voor een effectieve planning.

Queuing Theorie maakt gebruik van wiskundige formules en modellen, zoals de ‘M/M/1-queue’ (Markovian Queue met één server). Deze modellen bieden een gestructureerde aanpak om de complexiteit van wachtrijen te begrijpen en helpen bij het maken van schattingen over de wachttijd, de gemiddelde wachtrijlengte en andere indicatoren.

Het doel van Queuing Theorie is optimalisatie. Organisaties streven naar de juiste balans tussen kosten en klanttevredenheid. In wezen biedt Queuing Theorie een wiskundige basis om de werking van wachtrijen te begrijpen en organisaties te helpen hun processen aan te passen voor meer efficiëntie. Door middel van deze analyse kunnen ze niet alleen voorspellen hoe lang mensen moeten wachten, maar ook bepalen hoeveel agents nodig zijn om aan de vraag te voldoen. Uiteindelijk leidt dit tot een verbeterde service en kostenbesparingen.

### Simulatie software

Simulatiesoftware is als het ware een soort geavanceerd digitaal laboratorium voor bedrijven. Het stelt organisaties in staat om virtuele omgevingen te creëren die lijken op hun werkelijke scenario’s en vervolgens verschillende situaties te simuleren om inzichten te verkrijgen en besluitvorming te verbeteren.

In plaats van puur op basis van giswerk of intuïtie beslissingen te nemen, maakt simulatiesoftware gebruik van gegevens en wiskundige modellen om de werkelijkheid na te bootsen. Het proces begint met het invoeren van relevante gegevens, zoals de aankomst van klanten, de duur van service-interacties, en andere belangrijke grootheden die het bedrijf beïnvloeden.

Wanneer de simulatie wordt uitgevoerd, analyseert de software de interacties en dynamiek van het systeem. Het onthult patronen en trends, identificeert mogelijke knelpunten en geeft inzicht in hoe het systeem reageert op veranderende omstandigheden. Het is als het observeren van een virtuele wereld, waarbij verschillende testen gedaan kunnen worden om de mogelijke problemen en consequentie te ervaren.

De kracht van simulatiesoftware ligt in de mogelijkheid om 'wat-als' scenario's te bekijken. Wat als de vraag plotseling toeneemt? Of wat als bepaalde middelen schaarser worden? Bedrijven kunnen experimenteren met verschillende variabelen en zien hoe het systeem reageert, zonder daadwerkelijk risico's te nemen in de echte wereld. Hierdoor kan er goed geanticipeerd worden in de verdere toekomst.

De resultaten van de simulatie bieden bedrijven waardevolle inzichten waarmee ze beter onderbouwde beslissingen kunnen nemen. Of het nu gaat om het optimaliseren van bedrijfsprocessen, het plannen van resources of het anticiperen op toekomstige uitdagingen, simulatiesoftware is een instrument waarmee bedrijven hun operationele efficiëntie kunnen verbeteren en strategieën kunnen ontwikkelen op basis van een dieper begrip van complexe systemen.

### WFM-software

Workforce Management (WFM) software is een plannersoftware die ervoor moet zorgen dat verschillende zaken binnen het bedrijf geordend en efficiënt lopen. Zoals de administratieve werkzaamheden, klantenservice en bezetting van medewerkers. Ideaal dus voor bedrijven waarbij bezetting en ordenen cruciaal is, zoals een callcenter.

In feite verzamelt de WFM-software allerlei informatie, zoals het normale werkvolume, drukke tijden en de geschatte tijdsduur van taken. Met deze gegevens maakt de software een soort digitale agenda. Het plant wanneer medewerkers moeten werken op basis van verwachte drukte. Als er bijvoorbeeld meer werk te doen is tijdens bepaalde uren, zal de software meer medewerkers inplannen.

De WFM-software is ideaal aangezien het ook rekening houdt met andere factoren, zoals de vaardigheden van de medewerkers en hun beschikbaarheid. Als een medewerker bijvoorbeeld goed is in het afhandelen van specifieke taken, kan de software ervoor zorgen dat deze persoon wordt ingezet wanneer die vaardigheden het meest nodig zijn.

Daarnaast helpt de software om overbelasting van medewerkers te voorkomen. Het zorgt ervoor dat niemand te veel werk heeft en dat taken eerlijk worden verdeeld. Kortom, WFM-software maakt het bedrijfsleven efficiënter door ervoor te zorgen dat het juiste aantal mensen, met de juiste vaardigheden, op het juiste moment aan het werk is.

### Machine Learning-modellen

Machine Learning-modellen worden gemaakt door algoritmes te laten trainen met verschillende gegevens, zowel gelabeld als ongelabeld. Hierdoor kan het algoritme zichzelf ontwikkelen en kan het specifieker worden. Er zijn verschillende algoritmes: begeleid leren, leren zonder toezicht, Semi-gesuperviseerd leren en versterkend leren (Staff, 2023).

Begeleid leren is onderdeel van machine learning en artificiële intelligentie. Het gebruikt niet bestaande of niet expliciete labels en gegevens om verschillende categorieën te creëren, hierdoor kan zo’n algoritme constant zijn kennis vergroten. Er komen dan verschillende tabellen, labels en kolommen die nieuwe thema’s en onderwerpen van elkaar kan onderscheiden. Het resultaat hiervan is vervolgens dat een algoritme steeds specifieker wordt, wanneer er naar informatie wordt opgevraagd. Ook kan het na ontwikkelingen steeds meer informatie geven aan de ontvanger. Hierdoor is zo’n model ontzettend handig voor bedrijven die zichzelf constant aan het ontwikkelen zijn (IBM, 2024)

Een model kan ook leren door middel van expliciete informatie, dit noemt men dan ook wel leren zonder toezicht. Aangezien een algoritme gebruik maakt van bestaande gegevens om zichzelf verder te ontwikkelen. Hierdoor hebben ze nog een grotere invloed op het dagelijks leven als de algoritmes die gebruik maken van onbestaande gegevens, aangezien de algoritmes die werken met bestaande informatie relevanter en specifieker zijn in hun antwoorden (Google Cloud, 2024)

Naast de twee eerdere mogelijkheden om een algoritme te trainen en te ontwikkelen met ongelabeld als gelabeld gegevens, kan dit ook gedaan worden door een mix van gelabeld en ongelabelde gegevens. Hierbij zorg je ervoor dat het model nog steeds zichzelf met specifieke informatie kan ontwikkelen, maar tegelijkertijd zal deze gelabelde gegevens zich goed samenstellen met de ongelabelde gegevens, waardoor het model gelabelde en ongelabelde gegevens niet meer van elkaar kan onderscheiden (Bewtra, 2023).

## Programmeertalen

In de programmeerwereld zijn er verschillende programmeertalen, elk van deze talen heeft een andere manier van coderen. Een paar van de populairste programmeertalen zijn: Python, Javascript, C++, SQL, C#. De verschillende manieren van coderen bieden ondersteuning in de onlinewereld, waarbij verschillende taken uitgevoerd worden door deze talen.

### Python

Python is een staat bekend voor simpele structuur en goede leesbaarheid. Python is een populaire keuze voor zowel beginners als ervaren programmeurs. De taal legt de nadruk op leesbaarheid en overzichtelijkheid, waardoor ontwikkelaars productieve en onderhoudbare code kunnen schrijven. Python heeft uitgebreide bibliotheken en templates voor verschillende toepassingen, van websiteontwikkeling en data-analyse tot kunstmatige intelligentie. De intuïtieve aard van Python maakt snelle ontwikkeling mogelijk, wat bijdraagt aan de goede reputatie die Python bezit onder programmeurs.

### Javascript

Javascript is zoals eerder benoemd een veelgebruikte programmeertaal die vaak wordt gebruikt om websites creatief te ontwerpen. Javascript gaat in samenwerking met Html, Html is de kern achter de creatieve opmaak. Je kan op een vrij eenvoudige manier je website ontwerpen en zelfs online zetten. Daarnaast kan je met Html de gebruikers hun handelingen op de site traceren en tussen verschillende servers communiceren. Dit allemaal gebaseerd hoe je je code opstelt zodat de server reageert op de input van de gebruiker. Javascript wordt geschreven tussen <script>-tags in een HTML-document. Wanneer de website wordt geladen voert de server de codes uit, maar zijn deze codes niet te zien op de website. Hierdoor kan eenvoudig lijkende website al ontzettend veel codes bevatten.

### C++

C++ is een populaire programmeertaal die werkt op de basis van de C-taal. Deze programmeertaal is gericht op efficiënte systeemprogrammering. Deze taal combineert de flexibiliteit van laagniveau-programmeren met de abstractie van geavanceerd programmeren. C++ wordt voor meerdere toepassingen gebruikt, zoals game- en systeemontwikkeling. Het bezit een weide selectie van functies, waardoor programmeurs complexe software kunnen ontwikkelen. De taal staat bekend om zijn prestatiegerichtheid. C++ is een belangrijk instrument voor veel programmeurs en blijft relevant in moderne softwareontwikkeling.

### SQL

Structured Query Language (SQL) is een taal die veel gebruikt wordt voor relationele databases. In SQL worden queries gebruikt om interactie te hebben met databases. Hierbij zijn er verschillende opties zoals SELECT, INSERT, UPDATE EN DELETE. SELECT wordt gebruikt om gegevens uit de database op te halen. INSERT om nieuwe gegevens toe te voegen, UPDATE om bestaande gegevens uit de database op te halen en DELETE om gegevens te verwijderen. De codeertaal die gebruikt wordt bij SQL noemt men ook wel declaratieve syntaxis, waarbij gebruikers van de database specifiek hun opdrachten kunnen overbrengen op de server. Met de CREATE, ALTER EN DROP opties kan je de structuur van de database aanpassen naar omstandigheden. Al deze opties en code die gebruikt worden moeten het gebruik van databases eenvoudiger maken.

### C#

C# (uitgesproken als "C-sharp") is net zoals de eerder benoemd één van de bekendste programmeertalen die er is. De naamsbekendheid komt mede doordat Microsoft achter deze taal zit. De taal is bedacht en ontworpen voor het ontwikkelen van Windows- en websitetoepassingen maar ook voor softwareprojecten. C# voegt verschillende elementen van C en eerder benoemde C++ met moderne programmeerregels. Hierbij wordt het vervolgens naar een tussentaal omgeschreven genaamd IL (intermediate Language) en deze taal wordt uitgevoerd op het .Net. Hierdoor is C# gebruikelijk op meerdere platformen, waardoor programmeurs code kunnen gebruiken die simultaan op verschillende besturingssystemen werkt. De taal bezit uitgebreide bibliotheken voor diverse ontwikkelingsbehoeften. Door deze weide selectie is C# een krachtige keuze voor de ontwikkeling van software.

### Algemene regels van programmeren

Bij coderen is het van belang dat het leesbaar, te onderhouden en efficiënte code is. Belangrijke principes in de codeerwereld is de opmaak, gebruik van juiste variabelen voor namen en functies. Dit allemaal om consistent en duidelijk te houden voor andere die de code lezen. Men vindt namelijk in de codeerwereld, hoe korter de werkende code is, hoe beter je bent in coderen. Hierbij is het dus belangrijk om commentaar van andere te implanteren in de code, om zo jezelf steeds verder te ontwikkelen. Hierbij wil je dus herhalingen en duplicatie van code vermijden, aangezien die overbodig is voor het programma, en het juist onduidelijker wordt voor andere.

## ChatGPT als programmeur

ChatGPT is een krachtig hulpmiddel voor kunstmatige intelligentie dat kan worden gebruikt om code te schrijven in programmeertalen zoals C++, Java en Python. Het maakt gebruik van natuurlijke taalverwerking om de vragen van gebruikers te begrijpen en genereert vervolgens stukjes code met oplossingen.

Deze machine-learning technologie is klaar om een revolutie teweeg te brengen in de manier waarop mensen softwaretoepassingen schrijven. Dankzij de mogelijkheid om tekstinvoer te interpreteren in functionerende coderegels, krijgen ontwikkelaars de beschikking over een krachtig hulpmiddel om effectieve programma's te maken zonder dat ze uren bezig zijn met het handmatig schrijven ervan. Aangezien de taalmodellen die ChatGPT gebruikt voortdurend worden bijgewerkt en verfijnd op basis van feedback van gebruikers, kan de kwaliteit van de gegenereerde code na verloop van tijd verbeteren.

ChatGPT kan de ontwikkeltijd aanzienlijk verkorten, terwijl programmeurs zich kunnen richten op taken van een hoger niveau in plaats van regel na regel code te schrijven. ChatGPT kan codefragmenten, voorbeeldcode en kopieerbare daadwerkelijke programmeercode in natuurlijke taal genereren. Dit maakt het zeer geschikt voor taken op het gebied van applicatieontwikkeling, zoals het maken van apps of het debuggen van bestaande apps.

Met ChatGPT kunnen ontwikkelaars snel concepten van hun app maken zonder handmatig elke regel code te schrijven. De gegenereerde uitvoer versnelt de bewerkingsfase door een volledig framework te bieden, waardoor ontwikkelaars de parameters van hun projecten eenvoudig kunnen aanpassen aan hun behoeften en voorkeuren.

Dankzij de mogelijkheid tot dialoog kunnen gebruikers vragen stellen over hun projecten en krijgen ze antwoorden rechtstreeks van de software zelf terwijl ze eraan werken. Dit vereenvoudigt het ontwikkelproces nog meer en zorgt ervoor dat ontwikkelaars alle informatie bij de hand hebben die ze nodig hebben bij het ontwikkelen van applicaties.

# Eindproduct 2

## Modelcode

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using Common.Database;

namespace Common.Data

{

    /// <summary>

    /// Class voor Erlang berekeningen

    /// </summary>

    public static class Erlang

    {

        /// <summary>

        /// Lees de benodigde data (AHT, Erlang en aantallen)

        /// </summary>

        /// <param name="startDate">Begindatum</param>

        /// <param name="endDate">Einddatum</param>

        /// <param name="customerId">KlantId</param>

        /// <returns></returns>

        public static DataTable GetData(DateTime startDate, DateTime endDate, Int32 customerId)

        {

            DataTable dt;

            try

            {

                using (IDatabase db = DatabaseFactory.GetDatabase(Globals.LogDatabase))

                {

                    db.SetCommand("SelServicenummerCalls", CommandType.StoredProcedure);

                    db.AddInParameter(DbType.Int32, "@KlantId", customerId);

                    db.AddInParameter(DbType.DateTime, "@StartDatum", startDate);

                    db.AddInParameter(DbType.DateTime, "@EindDatum", endDate);

                    dt = db.ExecuteDataTable();

                }

            }

            catch

            {

                return new DataTable();

            }

            return dt;

        }

        /// <summary>

        /// Bepaal het aantal Erlangs

        /// </summary>

        /// <param name="totalCalls">Totaal aantal calls in een periode</param>

        /// <param name="reportingPeriod">Periode in minuten</param>

        /// <param name="aht">Average Handling Time: gemiddelde gespreksduur plus afhandeltijd (wrapup) in seconden</param>

        /// <returns></returns>

        public static double GetErlangs(Int32 totalCalls, Int32 reportingPeriod, Int32 aht)

        {

            if (reportingPeriod == 0) { return 0.00; }

            return totalCalls \* aht / 3600 \* 60 / reportingPeriod;

        }

        /// <summary>

        /// Bereken het aantal benodigde agenten op basis van de gegeven waarden met de Erlang-C formule

        /// </summary>

        /// <param name="totalCalls">Totaal aantal calls in een periode</param>

        /// <param name="reportingPeriod">Periode in minuten</param>

        /// <param name="aht">Average Handling Time: gemiddelde gespreksduur plus afhandeltijd (wrapup) in seconden</param>

        /// <param name="servicelevelPerc">Servicelevel in %. Aantal calls wat binnen de gestelde serviceleveltijd moet zijn opgenomen</param>

        /// <param name="servicelevelTime">Tijd in seconden waarin een call moet zijn opgenomen</param>

        /// <param name="shrinkage">Niet werkbare tijd in %. De tijd dat een agent niet aan bellen toekomt, bijvoorbeeld door overleg, wc-bezoek, etc. Gemiddeld ligt dit rond 30%</param>

        /// <returns></returns>

        public static Int32 NeededAgents(double totalCalls, long reportingPeriod, double aht, double servicelevelPerc, double servicelevelTime, double shrinkage)

        {

            if (reportingPeriod <= 0) { return 0; }

            double intensity;

            servicelevelPerc = servicelevelPerc / 100;

            shrinkage = shrinkage / 100;

            intensity = (totalCalls / (reportingPeriod \* 60)) \* aht;

            long minimumAgents = (Int64)intensity;

            if (minimumAgents < 1) { minimumAgents = 1; }

            long agents = minimumAgents;

            while (Servicelevel((long)totalCalls, reportingPeriod, aht, servicelevelTime, agents) < servicelevelPerc)

            {

                agents++;

            }

            if (shrinkage < 0) { shrinkage = 0; }

            if (shrinkage > 1) { shrinkage = 0.99; }

            double result = agents / (1 - shrinkage) + 0.5; //Afronden naar boven

            if (totalCalls == 0) { result = 0; }

            return (Int32)result;

        }

        /// <summary>

        /// Bereken het servicelevel

        /// </summary>

        /// <param name="totalCalls">Totaal aantal calls in een periode</param>

        /// <param name="reportingPeriod">Periode in minuten</param>

        /// <param name="aht">Average Handling Time: gemiddelde gespreksduur plus afhandeltijd (wrapup) in seconden</param>

        /// <param name="servicelevelTime">Aantal seconden waarin een call moet zijn opgenomen</param>

        /// <param name="agents">Aantal agenten</param>

        /// <returns></returns>

        private static double Servicelevel(double totalCalls, long reportingPeriod, double aht, double servicelevelTime, long agents)

        {

            double intensity = (totalCalls / (reportingPeriod \* 60)) \* aht;

            double result = 1 - (ProbCallWaits(totalCalls, reportingPeriod, aht, agents) \* Math.Exp(-(agents - intensity) \* servicelevelTime / aht));

            if (result > 1) { result = 1; }

            if (result < 0) { result = 0; }

            return result;

        }

        /// <summary>

        /// Bereken de mogelijkheid dat een call moet wachten

        /// </summary>

        /// <param name="totalCalls">Totaal aantal calls in een periode</param>

        /// <param name="reportingPeriod">Periode in minuten</param>

        /// <param name="aht">Average Handling Time: gemiddelde gespreksduur plus afhandeltijd (wrapup) in seconden</param>

        /// <param name="agents">Aantal agenten</param>

        /// <returns></returns>

        private static double ProbCallWaits(double totalCalls, long reportingPeriod, double aht, long agents)

        {

            double intensity = (totalCalls / (reportingPeriod \* 60)) \* aht;

            double occupancy = intensity / agents;

            double a\_n = 1;

            double suma\_k = 0;

            double a\_k;

            for (long k = agents; k > 0; k--)

            {

                a\_k = a\_n \* k / intensity;

                suma\_k += a\_k;

                a\_n = a\_k;

            }

            double result = 1 / (1 + ((1 - occupancy) \* suma\_k));

            if (result < 0) { result = 0; }

            if (result > 1) { result = 1; }

            return result;

        }

    }

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Common.DataObjects;

namespace Toolbox

{

    public partial class frmErlang : Form

    {

        private Common.DataObjects.Klant \_Klant;

        public frmErlang()

        {

            InitializeComponent();

        }

        private void FrmErlang\_Load(object sender, EventArgs e)

        {

            \_Klant = new Common.DataObjects.Klant();

            cmbCustomer.Init(true);

            cmbCustomer.CustomerSelectedEvent += CmbCustomer\_CustomerSelectedEvent;

            dtDatum.MaxDate = DateTime.Now.Date.AddDays(-1);

            this.Text = "Erlang-C";

        }

        private void CmbCustomer\_CustomerSelectedEvent(Klant klant)

        {

            \_Klant = klant;

        }

        private void BtnOk\_Click(object sender, EventArgs e)

        {

            DataTable dt = GetData();

            DG.DataSource = dt;

        }

        private DataTable GetData()

        {

            if (txtCorrection.Text == "") { txtCorrection.Text = "0"; }

            if (txtServicelevel.Text == "") { txtServicelevel.Text = "80"; }

            if (txtSla.Text == "") { txtSla.Text = "20"; }

            if (txtShrinkage.Text == "") { txtShrinkage.Text = "0"; }

            Int32 correction = Int32.Parse(txtCorrection.Text);

            Int32 servicelevel = Int32.Parse(txtServicelevel.Text);

            Int32 sla = Int32.Parse(txtSla.Text);

            Int32 shrinkage = Int32.Parse(txtShrinkage.Text);

            if (shrinkage < 0 || shrinkage > 100) { shrinkage = 0; }

            if (servicelevel < 0 || servicelevel > 100) { servicelevel = 80; }

            if (sla < 0 || sla > 500) { sla = 20; }

            if (correction <= 0 || correction > 100) { correction = 0; }

            txtServicelevel.Text = servicelevel.ToString();

            txtCorrection.Text = correction.ToString();

            txtShrinkage.Text = shrinkage.ToString();

            txtSla.Text = sla.ToString();

            DataTable dt = Common.Data.Erlang.GetData(dtDatum.Value.Date.AddHours(8), dtDatum.Value.Date.AddDays(1), \_Klant.Id);

            dt.Columns.Add(new DataColumn("Agents needed", typeof(Int32)));

            for (int i = 0; i < dt.Rows.Count; i++)

            {

                Int32 callcount = (Int32)dt.Rows[i]["Call\_Count"];

                if (correction > 0) { callcount += (callcount / 100) \* correction; }

                dt.Rows[i]["Agents needed"] = Common.Data.Erlang.NeededAgents(callcount, 60, (Int32)dt.Rows[i]["AHT\_Seconds"], servicelevel, sla, shrinkage);

                dt.Rows[i]["Call\_Count"] = callcount;

            }

            return dt;

        }

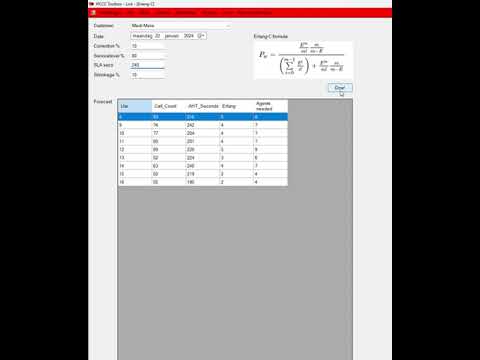
    }

}

Hierboven vind je de werkende geschreven code die schuilt achter de interface van de tool. De code is geschreven in de programmeertaal Python. Deze code kan door eenieder worden gebruikt en worden toegepast die beschikt over een eigen dataset. Voor het schrijven van deze code is zeer nauw samengewerkt met Remco Verhulst. Wij hebben veel van hem mogen leren en zijn dan ook tot een mooi resultaat gekomen.

Ook ChatGPT is een toereikende hand geweest in het ontwikkelen van deze code. Meneer Verhulst heeft ons laten zien wat de mogelijkheden zijn van ChatGPT en hoe dit programma gebruikt kan worden als ondersteuning, maar ook als controle. Zo kan het veel code versimpelen en korter schrijven, suggesties doen en zelfs nagaan of een deel van de code goed in elkaar steekt of opsporen waar fouten zijn gemaakt.

## Werkend model

[](https://www.youtube.com/embed/wy11aa-Kmj8?feature=oembed)

De tool werk als volgt: allereerst kiest de klant de naam van zijn of haar bedrijf, zo weet de tool in welke dataset hij moet zoeken. Vervolgens kiest men een voorbijgegane datum die vergelijkbaar is met de dag waarvoor de klant een planning wil maken. Als het bijvoorbeeld een aankomende maandag is waar geen bijzonderheden zoals feest- of actiedagen voor geldt, dan kiest men een maandag uit het verleden met een vergelijkbare drukte/situatie. Hierna vult de klant de verschillende variabelen in naar wens om daarna op de ‘doel’ knopt te drukken. Tot slot komt er een tabel tevoorschijn met daarin de gegevens die nodig zijn om een rooster te maken voor de gewenste datum.

Hieronder staat per kolom weergegeven wat de verschillende dingen binnen de tool-interface betekenen en doen:

-Customer: een zoekbalk waar men zijn of haar bedrijf in kan opzoeken.

-Date: een agenda waar voorbijgegane data in geselecteerd kan worden.

-Correcection %: dit is een variabele waar correcties in percentages kunnen worden toegepast voor de afwijking van het aantal bellers ten opzichte van de gekozen datum.

-Servicelevel %: dit is het percentage aan bellers waarvan de klant wil dat zij binnen het gekozen aantal ‘SLA secs’ geholpen worden.

-SLA secs: het aantal seconden waarin een beller wordt opgenomen.

-Shrinkage %: het percentage van het totaal aantal uren die agents ingeroosterd staan dat verloren gaat aan rook-, plas-, lunchpauzes, etc.

-Uur: tijdstip op de werkdag per heel uur aangegeven.

-Call\_Count: het aantal gesprekken in dat uur.

-AHT\_Seconds: de gemiddelde gespreksduur per beller in seconden.

-Erlang: de bijbehorende uitgerekende Erlang eenheid.

-Agents needed: het aantal mensen dat aanwezig moet zijn in het callcenter.

# Bronvermelding

Dzsquared. (2023, 3 april). *Use the Visual Studio code MSSQL Extension - SQL server*. Microsoft Learn. <https://learn.microsoft.com/en-us/sql/tools/visual-studio-code/sql-server-develop-use-vscode?view=sql-server-ver16>

Deze bron heeft ons verdiepende informatie gegeven over SQL servers.

*Wat is data modeling? | Definitie, belang, & types | SAP Insights*. (z.d.). SAP. <https://www.sap.com/netherlands/products/technology-platform/datasphere/what-is-data-modeling.html>

De informatie uit de bron heeft ons geholpen bij de verschillende vormen van data modeling.

Dialpad. (2023, 7 september). *What is call center optimization: An in-depth guide | Dialpad*. <https://www.dialpad.com/blog/call-center-optimization/>

Dialpad heeft ons informatie gegeven over de beste bezetting van een callcenter.

Vivian.Schreyer. (2023, 31 oktober). The importance of contact center optimization for customer service leaders. *Bucher + Suter*. <https://www.bucher-suter.com/contact-center-optimization/>

Ook deze bron heeft ons geholpen bij het onderzoeken van de optimale bezetting van een callcenter.

*What is predictive analytics and how does it work? | Google Cloud*. (z.d.). Google Cloud. <https://cloud.google.com/learn/what-is-predictive-analytics#:~:text=with%20predictive%20analytics.-,Predictive%20analytics%20defined,that%20might%20predict%20future%20behavior>

Informatie gekregen over predictive analytics

*Wat is een datamodel?* (z.d.). Lucidchart. <https://www.lucidchart.com/pages/nl/wat-is-een-datamodel>

Lucidchart heeft ons verdiepende informatie gegeven wat een datamodel exact is.

*Wat is een relationele database?* (z.d.). <https://www.oracle.com/nl/database/what-is-a-relational-database/#:~:text=Het%20relationele%20model,-In%20de%20beginjaren&text=Het%20relationele%20databasemodel%20is%20ontworpen,elke%20applicatie%20kon%20worden%20gebruikt>

Deze bron heeft ons de verdiepende informatie gegeven over een relationeel database.

Wat is SQL? En dit is wat je er allemaal over moet weten. (2023, 12 september). *icttrainingen*. Geraadpleegd op 9 januari 2024, van <https://icttrainingen.nl/blog/wat-is-sql#:~:text=SQL%20is%20een%20Engelse%20afkorting,voor%20verschillende%20toepassingen%20worden%20gebruikt>

Deze bron heeft ons nog meer informatie gegeven over een SQL-server.

Rijkaart, J. (2021, 17 juni). *Hoe ontwikkel je een voorspelmodel voor de vraag naar levensmiddelen? - Orcado*. Orcado. <https://orcado.nl/hoe-ontwikkel-je-een-voorspelmodel-voor-de-vraag-naar-levensmiddelen/>

Tips en voorbeelden hoe je zo goed mogelijk een voorspeldmodel kan ontwikkelen.

*Wat is data modeling? | Definitie, belang, & types | SAP Insights*. (z.d.-b). SAP. <https://www.sap.com/netherlands/products/technology-platform/datasphere/what-is-data-modeling.html>

Deze bron heeft informatie gegeven over de definities, belangen en types van data modeling.

ICTinformatiecentrum.nl. (2023, 16 november). *Wat is data analytics | uitleg en definitie - ICTinformatiecentrum.nl*. <https://www.ictinformatiecentrum.nl/data-analytics/wat-is-data-analytics>

ICTinformatiecentrum zijn site stond informatie over data analytics.

*Aberdeen Report: Most contact centers struggle to use data effectively | NICE*. (z.d.). NICE. <https://www.nice.com/blog/aberdeen-report-most-contact-centers-struggle-to-use-data-effectively>

Dit verslag gaf ons een duideijk beeld hoe je data efficiënt kan gebruiken.

*Wat is een entity relationship diagram?* (z.d.). Lucidchart. <https://www.lucidchart.com/pages/nl/wat-is-een-entity-relationship-diagram>

Deze bron heeft ons een verdiepend beeld gegeven over een entity relationship diagram

*7 Tips for a Good ER Diagram Layout*. (2021, 21 augustus). vertabelo. Geraadpleegd op 10 januari 2024, van <https://vertabelo.com/blog/vertabelo-tips-good-er-diagram-layout/>

Verschillende tips hoe je een ER diagram layout zo goed mogelijk kan maken.

*Wat is een callcenter* (2023, 27 augustus). indeed. Geraadpleegd op 10 januari 2024, van <https://nl.indeed.com/carrieregids/baan-vinden/wat-is-een-call-center>

Duidelijke informatie wat een callcenter is en hoe het werkt.

*What is Call Centre CRM Software? | Genesys*. (z.d.). Genesys. <https://www.genesys.com/en-sg/definitions/what-is-call-centre-crm>

Informatie over een CRM software van een call center.

*What is Contact Center Interactive Voice Response IVR*. (z.d.). NICE. <https://www.nice.com/glossary/what-is-contact-center-interactive-voice-response-ivr>

Informatie wat over IVR software, en hoe dit in praktijk werkt.

Staff, C. (2023, 29 november). *Machine Learning Models: What they are and How to build them*. Coursera. <https://www.coursera.org/articles/machine-learning-models>

Hoe call center modellen werken en hoe ze zijn gebouwd.

*What is supervised learning? | IBM*. (z.d.). <https://www.ibm.com/topics/supervised-learning#:~:text=the%20next%20step-,What%20is%20supervised%20learning%3F,data%20or%20predict%20outcomes%20accurately>

Hoe werkt supervised learning, deze informatie konden wij goed gebruiken bij de onderzoeken van verschillende modellen.

*What is unsupervised learning? | Google Cloud*. (z.d.). Google Cloud. <https://cloud.google.com/discover/what-is-unsupervised-learning#:~:text=Unsupervised%20learning%20in%20artificial%20intelligence,any%20explicit%20guidance%20or%20instruction>

Google cloud heeft informatie gegeven over unsupervised learning.

Dialpad. (2023b, september 7). *What is Call Center Optimization: An In-depth Guide | Dialpad*. <https://www.dialpad.com/blog/call-center-optimization/>

Wat is de optimale call center, hoe zorg je hiervoor.

Narayan, G. (2022, 16 november). *Six of the best strategies for contact center optimization*. Freshcaller Blog. <https://www.freshworks.com/freshcaller-cloud-pbx/call-center-software/contact-center-optimization-blog/>

Informatie wat de zes beste strategieën zijn voor de beste call center optimalisatie is.

*What is predictive Analytics? 5 examples | HBS Online*. (2021, 26 oktober). Business Insights Blog. <https://online.hbs.edu/blog/post/predictive-analytics>

*Informatie over predictive analytics en wat de beste strategieën zijn.*