

# DOCUMENTATIERAPPORT

onderdeel van de BACHELORPROEF

---

## ML.NET research binnen een WMS

Machine Learning mogelijkheden met ML.NET

Bachelor	Toegepaste Informatica
----------	------------------------

Keuzetraject Afstudeerrichting	Software Engineer
-----------------------------------	-------------------

Academiejaar	2023 - 2024
--------------	-------------

Student	Vanmarcke Aaron
---------	-----------------

Interne begeleider	Van Eeckhout Guy
--------------------	------------------

Externe promotor	Verstraete Kevin
------------------	------------------

## **Toelating tot bruikleen**

---

De auteur(s) geeft (geven) de toelating dit documentatierapport (onderdeel van de bachelorproef )voor consultatie beschikbaar te stellen en delen van dit rapport te kopiëren voor persoonlijk gebruik. Elk ander gebruik valt onder de bepalingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit dit rapport.

3/06/2024

## Woord vooraf

---

Bij het afronden van mijn bachelorproef wil ik graag enkele personen en organisaties bedanken die een belangrijke bijdrage hebben geleverd aan dit werk.

Allereerst wil ik mijn bachelorproefmentor en ideegever, Kevin Verstraete, Developer bij C&W Logistics, bedanken. Zijn begeleiding, expertise en waardevolle feedback hebben deze thesis mogelijk gemaakt en naar een hoger niveau getild.

Ook wil ik C&W Logistics bedanken voor het verstrekken van alle benodigde middelen en voor de kans om mijn bachelorproef grotendeels bij hen uit te voeren. De praktische ervaring en toegang tot relevante data hebben een cruciale rol gespeeld in het succes van dit onderzoek.

Daarnaast ben ik mijn familie dankbaar voor hun onvoorwaardelijke steun en hulp bij het nalezen van dit document. Hun geduld en betrokkenheid waren van onschatbare waarde.

Dank aan iedereen die, direct of indirect, heeft bijgedragen aan de totstandkoming van deze bachelorproef.

Aaron Vanmarcke

Kortrijk, 02/06/2024

# Inhoudsopgave

---

## Woord vooraf

<b>1</b>	<b>Inleiding.....</b>	<b>6</b>
1.1	Algemeen .....	6
1.2	Probleemstelling .....	7
1.3	Onderzoeksvraag .....	7
1.4	Experiment .....	7
1.4.1	Literatuuronderzoek: .....	7
1.4.2	Ontwikkeling van een ML.NET-model: .....	7
1.4.3	Evaluatie van het model:.....	7
1.4.4	Analyse van de resultaten: .....	8
<b>2</b>	<b>Experiment .....</b>	<b>9</b>
2.1	Inleiding .....	9
2.2	Machine Learning voordelen in een WMS .....	10
2.2.1	Vermindering van Kosten en Fouten .....	10
2.2.2	Verbeterde Klanttevredenheid .....	10
2.2.3	Verhoogde Operationele Efficiëntie .....	10
2.2.4	Financiële Impact van AI Implementatie .....	11
2.3	Hoe kan dit waargemaakt worden? .....	11
2.4	Implementatie voorbeelden.....	15
2.4.1	Voorbeeld 1: Productlocatiebepaling .....	15
2.4.2	Voorbeeld 2: Order Voorspelling .....	17
2.4.3	Case Study: Mercedes-Benz's Digital Twin Technology [7] .....	19
2.5	Toegang tot Code en Handleiding .....	20
2.5.1	Toegang tot praktische implementatiegids en code .....	20
2.5.2	Stap-voor-stap gids: .....	20
2.5.3	Code en datasets: .....	20
2.6	Kritische Analyse van de Resultaten.....	21
2.6.1	Het Belang van Kwaliteitsdata .....	21
2.6.2	Voordelen van ML.NET in een WMS .....	22
2.6.3	Beperkingen en Voorzichtigheid .....	22
2.6.4	Beperkingen en Negatieve Aspecten .....	23
<b>3</b>	<b>Conclusie.....</b>	<b>24</b>
3.1	Samenvatting van de Bevindingen.....	24
3.2	ML.NET biedt tal van voordelen voor WMS-systemen: .....	24
3.3	Praktische Implementaties: .....	24
3.4	Casestudy .....	24
3.5	Kritische Analyse en Aanbevelingen .....	24
3.6	Conclusie.....	25
	<b>AI Engineering Prompts.....</b>	<b>26</b>
<b>4</b>	<b>Referentielijst.....</b>	<b>28</b>
	<b>Bijlagen</b>	<b>29</b>



# 1 Inleiding

---

## 1.1 Algemeen

Warehouse Management Systemen (WMS) vormen een essentiële schakel in moderne logistieke operaties, waarbij ze voorraadbeheer, orderafhandeling en magazijnprocessen optimaliseren. Echter, met de toenemende complexiteit van supply chains worden traditionele WMS-oplossingen geconfronteerd met uitdagingen die kunnen resulteren in inefficiënties en suboptimale besluitvorming. Deze bachelor thesis onderzoekt het potentieel van ML.NET, een krachtig Machine Learning-framework voor .NET-ontwikkelaars, om deze uitdagingen aan te pakken en warehouse management te transformeren.

Geïnspireerd door mijn stage bij C&W Logistics richt dit onderzoek zich op de integratie van ML.NET in WMS-systemen om operationele efficiëntie te verbeteren, voorraadbeheer te optimaliseren en besluitvormingsprocessen te versterken. Moderne magazijnen worden geconfronteerd met een scala aan uitdagingen, waaronder stijgende klantverwachtingen, fluctuerende vraagpatronen en de noodzaak van real-time inzichten voor voorraadbeheer en orderafhandeling. Traditionele WMS-oplossingen schieten vaak tekort om deze uitdagingen effectief aan te pakken, wat leidt tot operationele inefficiënties en suboptimale besluitvorming.

In deze thesis heb ik verschillende stappen doorlopen om de integratie van ML.NET in een WMS te onderzoeken en te implementeren. Het onderzoek begon met een grondige analyse van bestaande casestudies en wetenschappelijke literatuur over de integratie van AI en ML in WMS-systemen. Deze literatuurstudie gaf inzicht in de huidige best practices en leidde tot geïnformeerde keuzes over de te gebruiken methodologieën en technologieën.

Vervolgens heb ik relevante datasets verzameld en geanalyseerd. Voor de ontwikkeling van een Product Voorraad Predictor heb ik gegevens verzameld over historische verkoopgegevens, inclusief datum en orderaantallen, aangevuld met seizoensgebonden trends en feestdagen. Deze gegevens werden opgeschoond en genormaliseerd om consistente en betrouwbare input voor de Machine Learning-modellen te verzekeren.

Bij de modelselectie en ontwikkeling werden verschillende ML-modellen binnen ML.NET geëvalueerd, zoals neurale netwerken, decision trees en regressiealgoritmen. Deze modellen werden geoptimaliseerd voor specifieke taken binnen het WMS, zoals dynamische locatietoewijzing van producten en tijdreeksvoorspellingen voor voorraadbeheer. Na een grondige evaluatie van de prestaties van verschillende modellen, werden de best presterende modellen geselecteerd en verder verfijnd.

De implementatie van ML.NET-functionaliteiten in het WMS omvatte de integratie van Machine Learning-modellen voor voorraadoptimalisatie en magazijnindelingsoptimalisatie. Dit stelde het systeem in staat om real-time beslissingen te nemen over de plaatsing van inkomende pallets. Daarnaast werd een gebruiksvriendelijke interface ontwikkeld, zodat gebruikers gemakkelijk toegang hadden tot ML-gestuurde beslissingen en aanbevelingen.

Kortom, de integratie van ML.NET in WMS-systemen biedt een transformerende kans om magazijnactiviteiten te verbeteren, resourcegebruik te optimaliseren en waardevolle inzichten uit

data te halen. Ondanks mogelijke uitdagingen zoals data-integratie en modeltraining, zijn de voordelen significant. Dit onderzoek draagt bij aan het begrip van de potentiële voordelen en uitdagingen van het gebruik van ML.NET in een WMS-context, en biedt een basis voor verdere ontwikkelingen en toepassingen in de logistieke industrie.

## 1.2 Probleemstelling

De kernproblematiek die ik in deze thesis behandelen is de noodzaak om Warehouse Management Systemen (WMS) te moderniseren en te optimaliseren in het licht van de toenevende complexiteit van supply chains. Traditionele WMS-oplossingen kampen met beperkingen op het gebied van flexibiliteit, schaalbaarheid en analytische mogelijkheden. Deze beperkingen leiden tot inefficiënties, suboptimale beslissingen en uiteindelijk suboptimale prestaties van mijn magazijnoperaties. In een wereld waar klantverwachtingen blijven stijgen en vraagpatronen voortdurend fluctueren, is er een dringende behoefte aan geavanceerde technologieën die in staat zijn om real-time inzichten te bieden en mijn besluitvorming te verbeteren.

## 1.3 Onderzoeksvraag

In deze thesis beantwoorden ik de volgende onderzoeksvraag: **Hoe kan ML.NET worden toegepast in WMS-systemen om operationele efficiëntie, voorraadbeheer en besluitvormingsprocessen te verbeteren?**

## 1.4 Experiment

Ik ga een experimentele aanpak gebruiken om mijn onderzoeksvraag te beantwoorden, die uit de volgende stappen bestaat:

### 1.4.1 Literatuuronderzoek:

Ik begin met een grondige studie van bestaand onderzoek over de toepassing van Machine Learning (ML) in WMS-systemen. Dit omvat het analyseren van wetenschappelijke artikelen, case studies en technische rapporten om de huidige stand van zaken te begrijpen en potentiële toepassingsgebieden van ML.NET te identificeren. Hierbij ligt de focus op de voordelen en beperkingen van ML-integratie in WMS, evenals de verschillende ML-modellen die in eerdere studies zijn toegepast.

### 1.4.2 Ontwikkeling van een ML.NET-model:

Op basis van de inzichten uit het literatuuronderzoek ontwikkel ik een ML.NET-model dat is afgestemd op een specifiek WMS-scenario. Dit scenario richt zich op warehouse optimalisatie en voorraadbeheer. Het model wordt getraind met historische data, inclusief verkoopgegevens, seizoensgebonden trends en andere relevante factoren. Hierbij maak ik gebruik van verschillende ML-algoritmen binnen ML.NET, zoals neurale netwerken, decision trees en regressiealgoritmen, om het meest geschikte model voor mijn toepassing te identificeren.

### 1.4.3 Evaluatie van het model:

Na de ontwikkeling van het ML.NET-model evalueer ik de prestatie ervan door middel van experimenten en simulaties. Ik gebruik verschillende datasets om de nauwkeurigheid en efficiëntie van het model te testen. Hierbij gebruik ik zowel trainings- als testdata om de generaliseerbaarheid van het model te waarborgen. De evaluatiecriteria omvatten onder andere de voorspellingsnauwkeurigheid, de snelheid van het model en de robuustheid onder verschillende omstandigheden.

#### 1.4.4 Analyse van de resultaten:

Ik analyseer de resultaten van de evaluatie grondig om de effectiviteit van het ML.NET-model te beoordelen. Ik kijk naar de verbeteringen in operationele efficiëntie, voorraadbeheer en besluitvormingsprocessen binnen het WMS. Daarnaast bespreek ik de potentiële impact van de implementatie van ML.NET in WMS-systemen, inclusief de voordelen en mogelijke uitdagingen. Deze analyse biedt inzicht in hoe ML.NET kan bijdragen aan de modernisering en optimalisatie van WMS-systemen, en vormt een basis voor verdere ontwikkelingen en toepassingen in de logistieke industrie.

Door deze stappen te volgen, streef ik ernaar om een gedetailleerd en praktisch inzicht te bieden in de mogelijkheden en uitdagingen van het toepassen van ML.NET in WMS-systemen.



## 2 Experiment

---

### 2.1 Inleiding

Magazijnbeheer is al lange tijd een cruciaal onderdeel van de supply chain en bedrijfsvoering. Een goed georganiseerd magazijn speelt een essentiële rol bij het garanderen van de nauwkeurigheid en efficiëntie van voorraadbeheer. Het belang van magazijnbeheer kan niet worden overschat; het is van belang voor het succes van een bedrijf. In een wereld waar bedrijven streven naar het minimaliseren van magazijnen door bewegingen zoals Just-in-Time (JIT) en Lean manufacturing, blijven magazijnen cruciaal voor snelle levering en klanttevredenheid. Ze bieden waarde toevoegende diensten zoals retourverwerking [1].

Traditionele methoden van magazijnbeheer vertrouwen vaak op menselijke input en geheugen, wat kan leiden tot fouten. Dit handmatige proces is foutgevoelig en kan resulteren in onjuiste voorraadniveaus, inefficiënte orderverwerking en vertragingen in de levering. Door de complexiteit en het volume van de gegevens die betrokken zijn bij magazijnbeheer, is het gemakkelijk in te zien hoe menselijke fouten kunnen optreden.

Hier komt Machine Learning (ML) om de hoek kijken. Machine Learning biedt de mogelijkheid om patronen en trends in magazijngegevens te herkennen en te voorspellen, wat kan leiden tot verbeterde efficiëntie en nauwkeurigheid. Door het toepassen van ML in een Warehouse Management System (WMS), kunnen bedrijven beter anticiperen op vraagveranderingen, voorraadniveaus optimaliseren en de algehele prestaties van het magazijn verbeteren.

De toepassing van Machine Learning in magazijnbeheer betekent een aanzienlijke vooruitgang in de manier waarop bedrijven hun magazijnen kunnen beheren. Het stelt bedrijven in staat om nauwkeurigere beslissingen te nemen op basis van datasets, wat leidt tot verbeterde operationele efficiëntie en klanttevredenheid. In het volgende deel van deze thesis zal ik dieper ingaan op de impact van Machine Learning op magazijnbeheer en de specifieke voordelen van het gebruik van ML.NET in een WMS-omgeving.

## **2.2 Machine Learning voordelen in een WMS**

Machine Learning (ML) biedt aanzienlijke voordelen in de context van een Warehouse Management System (WMS). Door het gebruik van ML kunnen bedrijven kosten en fouten verminderen, klanttevredenheid verbeteren, en operationele efficiëntie verhogen.

### **2.2.1 Vermindering van Kosten en Fouten**

AI zal fouten in voorspellingen en over bevoorrading tot 50 % verminderen [2] Traditionele methoden voor voorraadbeheer zijn vaak foutgevoelig omdat ze sterk afhankelijk zijn van menselijke input. Machine Learning kan deze foutmarges aanzienlijk verlagen door nauwkeurige voorspellingen te maken op basis van historische gegevens en realtime analyses.

### **2.2.2 Verbeterde Klanttevredenheid**

De nauwkeurigheid van voorraadbeheer door ML leidt direct tot verbeterde klanttevredenheid. Door ervoor te zorgen dat producten beschikbaar zijn wanneer ze nodig zijn, en door sneller en efficiënter te reageren op klantbehoeften, kunnen bedrijven de verwachtingen van hun klanten overtreffen. Een goed beheerd magazijn zorgt ervoor dat klanten minder lang hoeven te wachten op hun bestellingen en dat retourprocessen soepel verlopen.

### **2.2.3 Verhoogde Operationele Efficiëntie**

Het gebruik van ML in een WMS verhoogt de operationele efficiëntie door repetitieve taken te automatiseren en door betere besluitvorming te ondersteunen. Dit resulteert in een beter gebruik van middelen, optimalisatie van orderverwerking en verbetering van de algehele workflow binnen het magazijn. Volgens G-Nius [3] kunnen bedrijven door de implementatie van AI en ML in logistiek en WMS aanzienlijke verbeteringen zien in hun operationele efficiëntie .

## 2.2.4 Financiële Impact van AI Implementatie

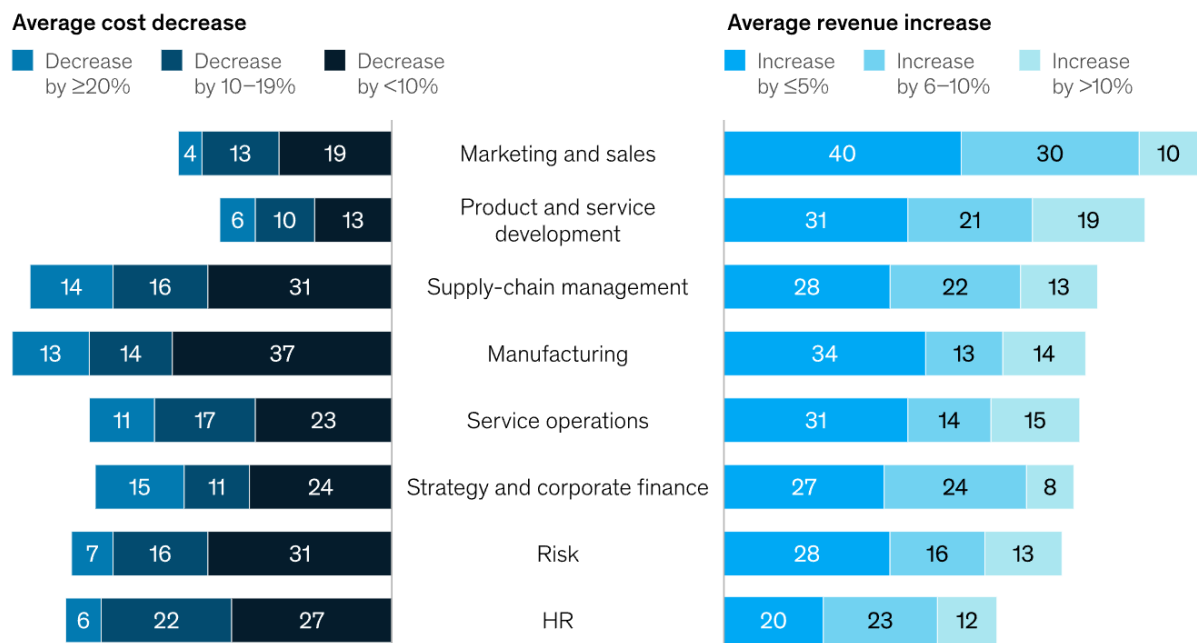
De financiële voordelen van AI-implementatie in supply chain management zijn ook duidelijk zichtbaar. Volgens een onderzoek van McKinsey [4] onder 2000 bedrijven, rapporteerde 31 % van de bedrijven een kostenreductie van minder dan 10 %, 16% van de bedrijven een reductie tussen 10 % en 19 %, en 14 % van de bedrijven een reductie van 20 % of meer. Bovendien meldde 28 % van de bedrijven een winststijging van minder dan 5 %, 22 % een stijging van 6 % tot 10 %, en 13 % een winststijging van 10 % of meer (zie Figuur 1 ).

Deze cijfers illustreren hoe effectief AI kan zijn in het transformeren van magazijnbeheer en het genereren van financiële voordelen. Door deze technologische vooruitgang kunnen bedrijven concurrerender worden en tegelijkertijd de kosten verlagen en de winst verhogen.

In het volgende deel van deze thesis zal ik dieper ingaan op specifieke Machine Learning technieken en modellen die kunnen worden toegepast binnen een WMS. Ik zal uitleggen hoe deze technieken kunnen worden geïmplementeerd om de gewenste resultaten te bereiken, waardoor kosten kunnen worden verlaagd en winst kan worden verhoogd.

**Revenue increases from adopting AI are reported most often in marketing and sales, and cost decreases most often in manufacturing.**

Cost decrease and revenue increase from AI adoption, by function,<sup>1</sup> % of respondents<sup>2</sup>



Figuur 1 <https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-proves-its-worth-but-few-scale-impact>

## 2.3 Hoe kan dit waargemaakt worden?

Machine Learning (ML) biedt een scala aan mogelijkheden om warehouse management systems (WMS) te optimaliseren. Door gebruik te maken van historische gegevens kunnen nauwkeurige voorspellingen en verbeterde beslissingen worden genomen, wat leidt tot efficiëntere operaties en lagere kosten. Voorbeelden hiervan zijn order voorspellingen, voorraadbeheer en route-optimalisatie. Om deze resultaten te bereiken, zijn er enkele cruciale stappen en hulpmiddelen nodig.

## Data Verzamelen en Voorbereiden

De eerste stap in het toepassen van Machine Learning in een WMS is het verzamelen en voorbereiden van gegevens. Voor order voorspellingen, bijvoorbeeld, heb je historische ordergegevens nodig, evenals aanvullende data die de hoeveelheid orders beïnvloedt, zoals seizoensgebonden trends, marketingacties en economische indicatoren. Het is van essentieel belang dat deze data nauwkeurig en compleet is om betrouwbare modellen te kunnen bouwen.

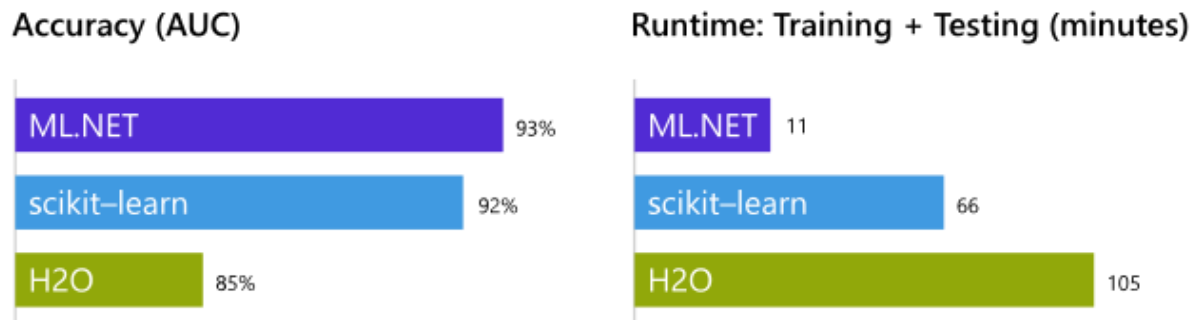
## Keuze van het Machine Learning Model

Nadat de data is verzameld en voorbereid, is de volgende stap het kiezen van een geschikt Machine Learning model. Voor deze thesis maak ik gebruik van ML.NET, een Machine Learning framework ontwikkeld door Microsoft.

## Waarom ML.NET?

ML.NET biedt verschillende voordelen ten opzichte van andere populaire Machine Learning frameworks zoals scikit-learn, een bekend Python-gebaseerd framework. Hieronder staan enkele redenen waarom ML.NET een uitstekende keuze is:

- **Integratie met het .NET Ecosysteem:** ML.NET is volledig geïntegreerd met het .NET ecosysteem, waardoor je eenvoudig Machine Learning functionaliteiten kunt toevoegen aan bestaande .NET applicaties zonder extra interoperabiliteitslagen.
- **Bekendheid voor .NET Ontwikkelaars:** Voor ontwikkelaars die al vertrouwd zijn met C# en het .NET framework, biedt ML.NET een gemakkelijke leercurve omdat het dezelfde taal en omgeving gebruikt.
- **Efficiëntie en Prestaties:** ML.NET is ontworpen voor hoge prestaties, wat resulteert in snelle en efficiënte Machine Learning modellen die geschikt zijn voor productieomgevingen [5] (zie Figuur 2 voor een vergelijking van de accuraatheid en snelheid van training en testen met ML.NET versus scikit-learn). Dit is bijzonder belangrijk voor WMS-bedrijven die grote hoeveelheden data moeten verwerken, waarbij snelheid en nauwkeurigheid essentieel zijn.
- **Flexibiliteit en Mogelijkheden:** ML.NET ondersteunt een breed scala aan Machine Learning taken, zoals regressie, classificatie, clustering en tijdreeksanalyse, wat het veelzijdig maakt voor verschillende toepassingen in een WMS.
- **Geen Python Vereist:** Hoewel veel Machine Learning frameworks op Python zijn gebaseerd, stelt ML.NET ontwikkelaars in staat om krachtige Machine Learning modellen te bouwen zonder Python te hoeven leren of gebruiken. Dit kan de complexiteit en overhead van het leren van een nieuwe programmeertaal verminderen.



Figuur 2 <https://dotnet.microsoft.com/en-us/apps/MachineLearning-ai/ml-dotnet>

### ML.NET Model Builder

Een van de gebruiksvriendelijke functies van ML.NET is de Model Builder, een intuïtieve UI die het mogelijk maakt voor mensen met weinig tot geen Machine Learning kennis om toch effectieve modellen te creëren. De Model Builder kan worden geïnstalleerd via de Visual Studio installatie tool en biedt een gemakkelijke interface om Machine Learning modellen te bouwen.

- **Scenario Selectie:** De Model Builder laat gebruikers kiezen uit verschillende scenario's die het beste passen bij hun voorspellingstaak. Voor locatie voorspellingen kun je bijvoorbeeld kiezen voor data classificatie (Regression). Voor order voorspellingen kun je kiezen voor (forecasting) (zie Figuur 3).
- **Data Selectie en Training:** Vervolgens selecteer je de data die je wilt testen en geef je aan wat je als uitkomst wilt van je data. Hier kun je bepalen welke data relevant is voor dit resultaat en welke niet. ML.NET zal je dataset trainen door verschillende algoritmen te proberen en het beste model te selecteren op basis van de prestaties.
- **Gebruik van het Model:** Na de training genereert Visual Studio een model dat makkelijk bruikbaar is voor programmeurs. Dit model kan worden geïntegreerd in applicaties om nieuwe gegevens te voorspellen op basis van Machine Learning.

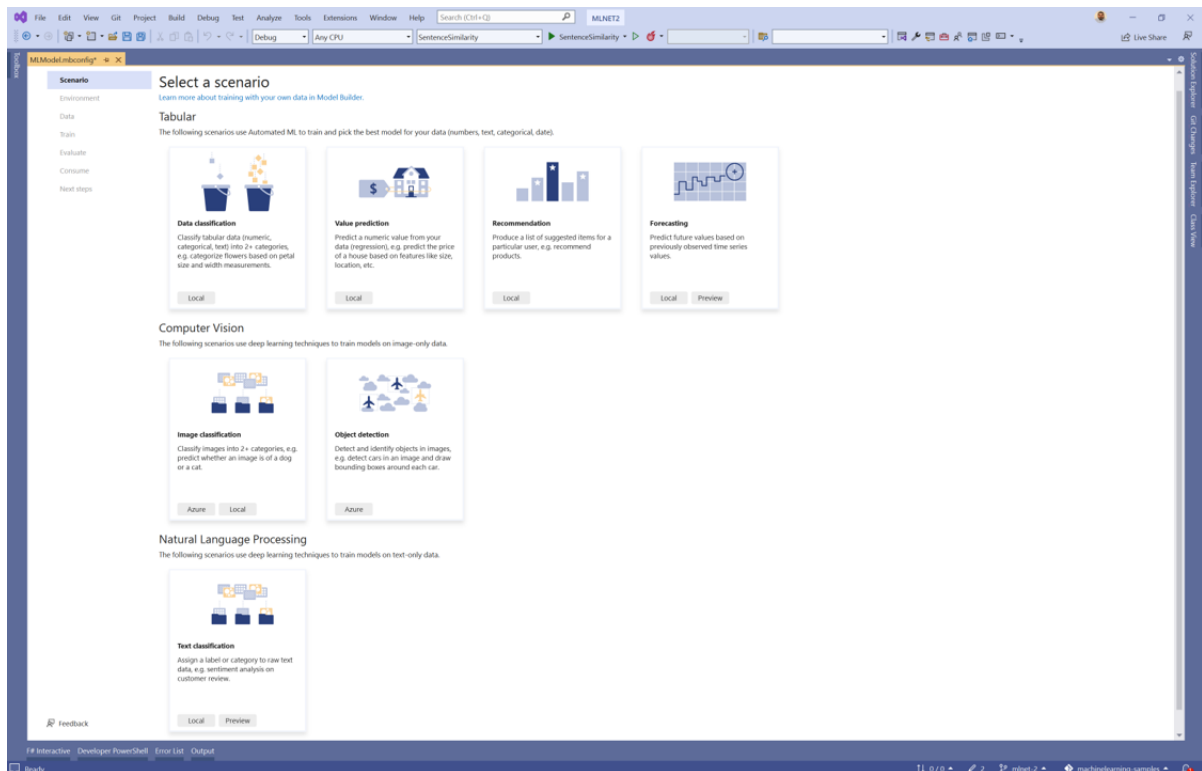
### Praktische Implementatie van ML.NET in een WMS

Door gebruik te maken van ML.NET kunnen bedrijven profiteren van verbeterde nauwkeurigheid en efficiëntie in hun WMS. Enkele praktische toepassingen zijn:

- **Voorraadbeheer:** Met nauwkeurige voorraadvoorspellingen kunnen bedrijven de voorraadniveaus optimaliseren, wat leidt tot lagere kosten en minder verspilling.
- **Order Voorspellingen:** Door historische ordergegevens te analyseren, kan ML.NET helpen bij het voorspellen van toekomstige orders, wat essentieel is voor het plannen van inkoop en productie.
- **Route-optimalisatie:** Machine Learning kan worden gebruikt om de meest efficiënte routes te bepalen voor orderverwerking en verzending, wat de operationele efficiëntie verhoogt. In magazijnen spenderen orderpickers tot wel 50 % van hun tijd aan wandelen om items te verzamelen voor bestellingen [6]. Een orderpicker is een medewerker

die verantwoordelijk is voor het verzamelen van items uit het magazijn om bestellingen te vervullen. Door route-optimalisatie kunnen orderpickers hun tijd efficiënter besteden aan het verzamelen van items, waardoor de algehele productiviteit van het magazijn wordt verbeterd.

Kortom, ML.NET biedt een krachtige en toegankelijke oplossing voor Machine Learning in een WMS, vooral voor organisaties die al gebruik maken van een .NET backend. Het biedt niet alleen hoge prestaties en nauwkeurigheid, maar ook een naadloze integratie met bestaande .NET applicaties, wat het een aantrekkelijk alternatief maakt voor traditionele Python-gebaseerde Machine Learning frameworks.



Figuur 3 <https://learn.microsoft.com/en-us/dotnet/Machine-Learning/automate-training-with-model-builder>

## 2.4 Implementatie voorbeelden

Voor een concreet begrip van hoe Machine Learning toegepast kan worden in een WMS-systeem, worden hier twee implementatievoorbeelden gepresenteerd.

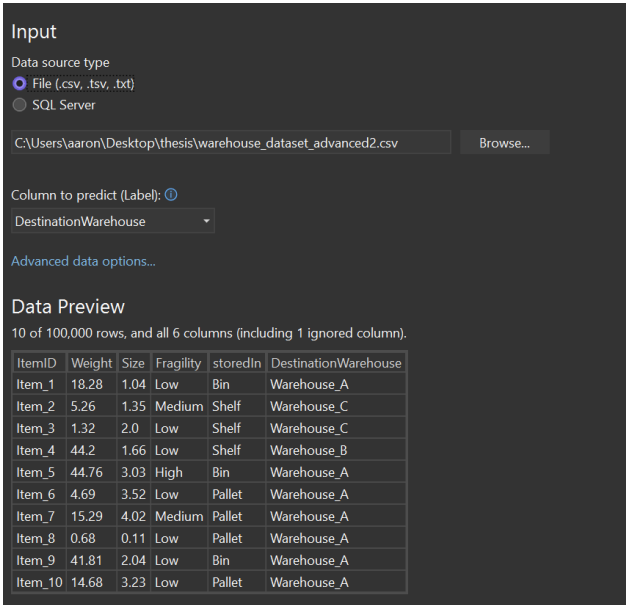
### 2.4.1 Voorbeeld 1: Productlocatiebepaling

Een van de uitdagingen in een warehouse management systeem (WMS) is het efficiënt opslaan van producten op basis van hun kenmerken en historische patronen. Om dit aan te pakken, is een Machine Learning model ontwikkeld met behulp van ML.NET.

Dit model maakt gebruik van historische gegevens, zoals het gewicht, de grootte, de breekbaarheid van producten, de huidige opslaglocatie en de bestemming van het magazijn (zie Figuur 4) , om te voorspellen waar toekomstige orders moeten worden geplaatst. Deze gegevens zijn afkomstig van een bedrijf en zijn ingevoerd in de Model Builder van ML.NET.

Na analyse bleek dat het LightGbmMulti-model het beste presteerde (zie Figuur 5) voor deze data. Door dit model te implementeren, kan het WMS nu nauwkeurig voorspellen waar producten moeten worden opgeslagen. Dit minimaliseert menselijke fouten en zorgt ervoor dat producten consistent worden opgeslagen volgens historische patronen, waardoor de operationele efficiëntie wordt verbeterd.

En kan dus geïmplementeerd worden op volgend manier (zie figuur 6).



**Input**

Data source type

☒ File (.csv, .tsv, .txt)

☐ SQL Server

C:\Users\aaroon\Desktop\thesis\warehouse\_dataset\_advanced2.csv [Browse...](#)

Column to predict (Label): [?](#)

DestinationWarehouse

[Advanced data options...](#)

**Data Preview**

10 of 100,000 rows, and all 6 columns (including 1 ignored column).

ItemID	Weight	Size	Fragility	storedIn	DestinationWarehouse
Item_1	18.28	1.04	Low	Bin	Warehouse_A
Item_2	5.26	1.35	Medium	Shelf	Warehouse_C
Item_3	1.32	2.0	Low	Shelf	Warehouse_C
Item_4	44.2	1.66	Low	Shelf	Warehouse_B
Item_5	44.76	3.03	High	Bin	Warehouse_A
Item_6	4.69	3.52	Low	Pallet	Warehouse_A
Item_7	15.29	4.02	Medium	Pallet	Warehouse_A
Item_8	0.68	0.11	Low	Pallet	Warehouse_A
Item_9	41.81	2.04	Low	Bin	Warehouse_A
Item_10	14.68	3.23	Low	Pallet	Warehouse_A

Figuur 4 data gebruikt voor productlocatie bepaling

Top 5 models explored					
	Trainer	MicroAccuracy	MacroAccuracy	Duration	#Iteration
29	LightGbmMulti	0.9998	0.9999	0.2	29
13	LightGbmMulti	0.9997	0.9993	0.2	13
42	LightGbmMulti	0.9995	0.9990	1.8	42
30	FastTreeOva	0.9994	0.9989	0.7	30
19	LbfgsMaximumEntropyMulti	0.9994	0.9992	0.6	19

Figuur 5 output van ML.NET analyse

## Warehouse Management System

Weight (kg):

Size (m³):

Fragility:

Stored In:

Add Pallet

Pallet added successfully. Destination warehouse: Warehouse\_A

Figuur 6 resultaat / implementatie van het model

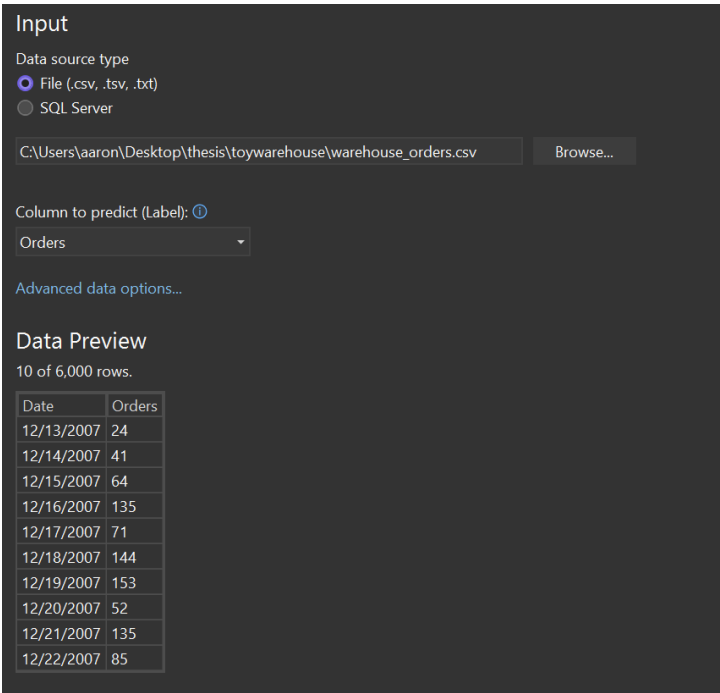


### 2.4.2 Voorbeeld 2: Order Voorspelling

Een cruciale uitdaging voor magazijnbeheerders is het anticiperen op de vraag en het plannen van de benodigde voorraadniveaus. Om dit aan te pakken, is een Machine Learning model ontwikkeld met behulp van ML.NET.

Dit model maakt gebruik van historische gegevens over de datum en de hoeveelheid orders om te voorspellen hoeveel orders er op een bepaalde dag zullen binnenkomen. Deze gegevens zijn afkomstig van een bedrijf en zijn geanalyseerd met behulp van ML.NET Model Builder (zie Figuur 7).

Na analyse bleek dat het FastForestRegression-model de beste score behaalde voor het voorspellen van orderaantallen op basis van historische gegevens. Door dit model te implementeren, kunnen magazijnbeheerders nauwkeuriger anticiperen op de vraag en de benodigde voorraadniveaus plannen. Dit zorgt voor een efficiënter voorraadbeheer, waarbij de kans op tekorten of overbodige voorraden wordt geminimaliseerd (Zie Figuur 8).



**Input**

Data source type

☒ File (.csv, .tsv, .txt)

☐ SQL Server

C:\Users\aaroon\Desktop\thesis\toywarehouse\warehouse\_orders.csv [Browse...](#)

Column to predict (Label): ⓘ

Orders

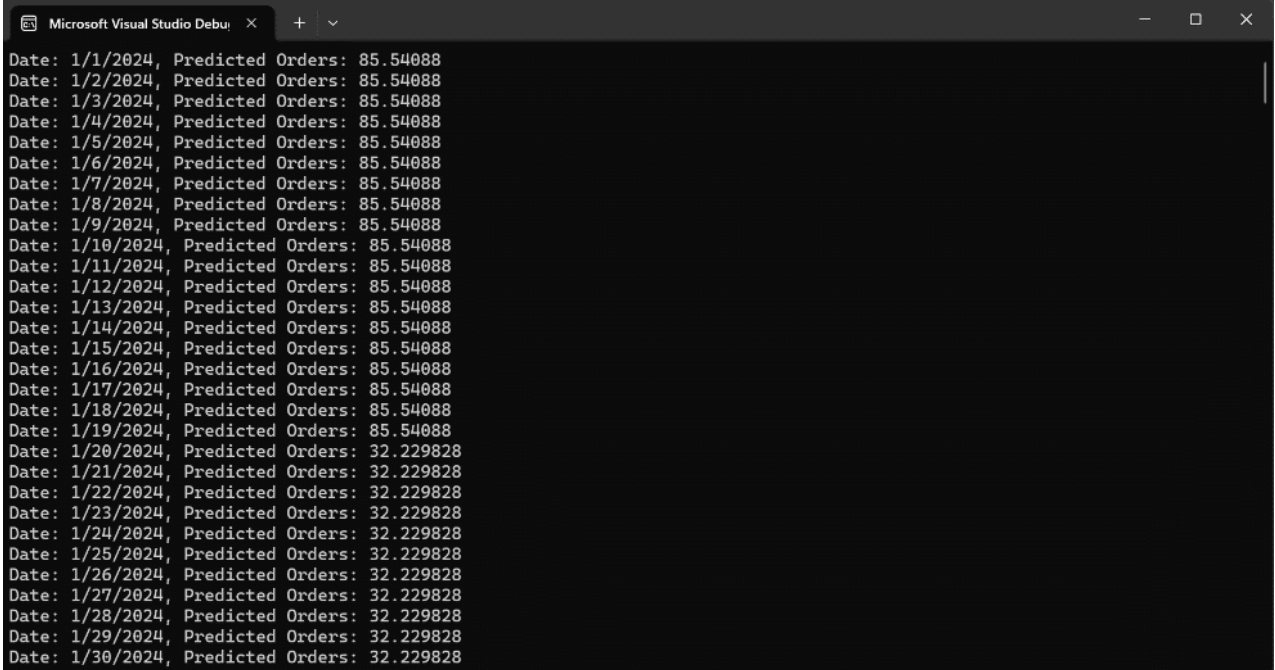
[Advanced data options...](#)

**Data Preview**

10 of 6,000 rows.

Date	Orders
12/13/2007	24
12/14/2007	41
12/15/2007	64
12/16/2007	135
12/17/2007	71
12/18/2007	144
12/19/2007	153
12/20/2007	52
12/21/2007	135
12/22/2007	85

*Figuur 7 data voor order voorspelling*



```
Microsoft Visual Studio Debug Console
Date: 1/1/2024, Predicted Orders: 85.54088
Date: 1/2/2024, Predicted Orders: 85.54088
Date: 1/3/2024, Predicted Orders: 85.54088
Date: 1/4/2024, Predicted Orders: 85.54088
Date: 1/5/2024, Predicted Orders: 85.54088
Date: 1/6/2024, Predicted Orders: 85.54088
Date: 1/7/2024, Predicted Orders: 85.54088
Date: 1/8/2024, Predicted Orders: 85.54088
Date: 1/9/2024, Predicted Orders: 85.54088
Date: 1/10/2024, Predicted Orders: 85.54088
Date: 1/11/2024, Predicted Orders: 85.54088
Date: 1/12/2024, Predicted Orders: 85.54088
Date: 1/13/2024, Predicted Orders: 85.54088
Date: 1/14/2024, Predicted Orders: 85.54088
Date: 1/15/2024, Predicted Orders: 85.54088
Date: 1/16/2024, Predicted Orders: 85.54088
Date: 1/17/2024, Predicted Orders: 85.54088
Date: 1/18/2024, Predicted Orders: 85.54088
Date: 1/19/2024, Predicted Orders: 85.54088
Date: 1/20/2024, Predicted Orders: 32.229828
Date: 1/21/2024, Predicted Orders: 32.229828
Date: 1/22/2024, Predicted Orders: 32.229828
Date: 1/23/2024, Predicted Orders: 32.229828
Date: 1/24/2024, Predicted Orders: 32.229828
Date: 1/25/2024, Predicted Orders: 32.229828
Date: 1/26/2024, Predicted Orders: 32.229828
Date: 1/27/2024, Predicted Orders: 32.229828
Date: 1/28/2024, Predicted Orders: 32.229828
Date: 1/29/2024, Predicted Orders: 32.229828
Date: 1/30/2024, Predicted Orders: 32.229828
```

*Figuur 8 resultaat voorbeeld 2*

### 2.4.3 Case Study: Mercedes-Benz's Digital Twin Technology [7]

Een illustratief voorbeeld van de toepassing van Machine Learning en AI in een Warehouse Management System wordt geleverd door Mercedes-Benz's Digital Twin Technology. Deze technologie gebruikt digitale tweelingen om virtuele kopieën van hun magazijnen te creëren, die real-time data ontvangen van sensoren in het echte magazijn. Deze data-rijke omgeving biedt een ideaal platform voor het toepassen van Machine Learning-algoritmen.

Door AI toe te passen op hun digitale tweelingen heeft Mercedes-Benz aanzienlijke verbeteringen gezien:

- **Slimme picking-routes:** Door het analyseren van de data in de digitale tweeling kan AI de meest efficiënte routes bepalen voor orderpickers, wat resulteert in tijdsbesparing en foutenvermindering.
- **Optimale productlocatie:** Met behulp van de overvloed aan data in de digitale tweeling kan AI de plaatsing van producten in het magazijn optimaliseren, waardoor opslag- en ophaalprocessen efficiënter worden.

Dit voorbeeld benadrukt niet alleen de voordelen van het gebruik van digitale tweeling-technologie in magazijnbeheer, maar toont ook hoe Machine Learning-technieken kunnen worden toegepast op gegevensrijke omgevingen om de operationele efficiëntie te verbeteren.

Mercedes-Benz's benadering kan als inspiratie dienen voor de implementatie van Machine Learning in WMS met ML.NET. Door een digitale tweeling te creëren, kunnen bedrijven zoals magazijnbeheerders een gedetailleerd, real-time overzicht krijgen van hun operaties, wat het toepassen van ML.NET-modellen vergemakkelijkt. Deze modellen kunnen helpen bij het voorspellen van optimale opslaglocaties, efficiënte order picking-routes en het beheren van voorraden op een manier die voorheen onbereikbaar was.

#### Implementatie van Digital Twin in WMS met ML.NET

De implementatie van een digitale tweeling in een WMS kan worden ondersteund door ML.NET door de volgende stappen te volgen:

1. **Data Verzamelen en Integreren:** Verzamel real-time data van sensoren en andere IoT-apparaten in het magazijn en integreer deze gegevens in de digitale tweeling.
2. **Model Training:** Gebruik ML.NET Model Builder om Machine Learning-modellen te trainen met behulp van historische en real-time data.
3. **Model Implementatie:** Implementeer de getrainde modellen in de digitale tweeling om voorspellingen en optimalisaties in real-time mogelijk te maken.
4. **Continue Verbetering:** Regelmatige updates en hertraining van de modellen om ze up-to-date te houden met nieuwe gegevens en veranderende operationele omstandigheden.

Het gebruik van ML.NET in combinatie met digitale tweeling-technologie kan magazijnbeheerders helpen om sneller en nauwkeuriger beslissingen te nemen, wat leidt tot verbeterde efficiëntie en lagere operationele kosten.

## **2.5 Toegang tot Code en Handleiding**

### **2.5.1 Toegang tot praktische implementatiegids en code**

Ben je geïnteresseerd in de praktische implementatie van de besproken experimenten? Dan vind je hier alle benodigde informatie:

#### **2.5.2 Stap-voor-stap gids:**

In de bijlage vind je screenshots met een gedetailleerde gids die je stap voor stap meeneemt door het proces van het maken en implementeren van je eigen Machine Learning model met ML.NET. Deze gids is perfect om je op weg te helpen met het creëren van je eigen modellen.

#### **2.5.3 Code en datasets:**

Via de link naar mijn GitHub-repository (zie onderaan deze thesis in de bijlagen) heb je toegang tot alle code die gebruikt is voor de voorbeelden in deze thesis. Dit omvat ook de datasets die gebruikt zijn om de modellen te trainen.

## 2.6 Kritische Analyse van de Resultaten

De experimenten en implementaties in deze thesis tonen duidelijk aan dat Machine Learning, met behulp van ML.NET, een waardevolle toevoeging kan zijn aan een Warehouse Management System (WMS). Toch is het essentieel om een kritische blik te werpen op de resultaten en de factoren die de effectiviteit van deze technologie beïnvloeden.

### 2.6.1 Het Belang van Kwaliteitsdata

Een van de belangrijkste bevindingen uit dit onderzoek is het cruciale belang van kwalitatief hoogwaardige data. Machine Learning-modellen zijn zo goed als de data waarmee ze worden gevoed; met slechte data krijg je immers slechte resultaten. Dit principe geldt ook voor de toepassingen van ML.NET in een WMS. Enkele observaties die dit benadrukken zijn:

- **Nauwkeurigheid van Voorspellingen:** De nauwkeurigheid van voorspellingen, zoals order volumes en productlocaties, hangt sterk af van de kwaliteit en volledigheid van de historische gegevens. Inconsistenties, ontbrekende waarden of verouderde data kunnen leiden tot onnauwkeurige en onbetrouwbare voorspellingen.
- **Training en Validatie:** Voor een betrouwbaar model is het noodzakelijk om een uitgebreide en representatieve dataset te hebben voor zowel training als validatie. Dit zorgt ervoor dat het model niet alleen leert van historische patronen maar ook goed generaliseert naar nieuwe, ongeziene data.
- **Relevantie van Data:** Het is essentieel dat de data die wordt gebruikt relevant is voor de uitkomst die men probeert te voorspellen. Relevante data beïnvloedt direct de nauwkeurigheid en bruikbaarheid van de voorspellingen.

#### Voorbeeld: Verder Uitwerking van Productlocatie voor Order Picker

Een praktisch voorbeeld is de verdere uitwerking van de productlocatiebepaling voor order pickers. Voor een efficiënte orderpicking is het belangrijk dat zware orders dichtbij de verzendingslocatie worden geplaatst, evenals producten die vaak samen worden besteld. Dit helpt de loopafstand te beperken, waardoor minder tijd verloren gaat en meer orders in kortere tijd kunnen worden afgehandeld, wat uiteindelijk leidt tot lagere kosten.

#### Waarom is deze data belangrijk?

- **Productkenmerken:** Gegevens over het gewicht, de afmetingen en de breekbaarheid van producten zijn cruciaal. Zware en grote items moeten dichtbij de verzendingslocatie worden geplaatst om de inspanning van de order pickers te verminderen.
- **Bestelpatronen:** Historische gegevens over welke producten vaak samen worden besteld, kunnen helpen bij het optimaliseren van de magazijnindeling. Dit zorgt ervoor dat order pickers minder hoeven te lopen, wat de efficiëntie verhoogt.
- **Orderfrequentie:** Producten die vaak worden besteld, moeten zich op gemakkelijk toegankelijke locaties bevinden om de orderpicktijd te minimaliseren.

In mijn onderzoek heb ik gebruikgemaakt van basisgegevens zoals gewicht, grootte, breekbaarheid, huidige opslaglocatie en bestemmingsmagazijn. Hoewel deze gegevens een solide basis bieden voor het maken van voorspellingen, zou het toevoegen van meer gedetailleerde en specifieke gegevens de nauwkeurigheid van de voorspellingen verder kunnen verbeteren. Dit leidt tot een hogere efficiëntie en lagere kosten.

Kortom, het gebruik van relevante en hoogwaardige data is essentieel voor de nauwkeurigheid van Machine Learning-modellen in een WMS. Hoe meer relevante gegevens beschikbaar zijn, hoe beter de voorspellingen en hoe groter de voordelen voor de operationele efficiëntie en kostenbesparing.

### 2.6.2 Voordelen van ML.NET in een WMS

Ondanks de uitdagingen met data, laten de resultaten zien dat ML.NET een krachtige en effectieve tool kan zijn voor het optimaliseren van WMS-processen. Enkele specifieke voordelen zijn:

- **Kostenreductie en Winstverhoging:** Veel bedrijven die ML.NET hebben geïmplementeerd, rapporteren aanzienlijke kostenreducties en winstverhogingen. Door beter voorraadbeheer en efficiëntere orderverwerking kunnen operationele kosten worden verminderd en klanttevredenheid worden verhoogd.
- **Aanpasbaarheid en Integratie:** ML.NET biedt een uitstekende aanpasbaarheid en kan gemakkelijk worden geïntegreerd in bestaande .NET-systemen. Dit maakt het een aantrekkelijke optie voor bedrijven die al een .NET-backend hebben, omdat ze geen nieuwe infrastructuur hoeven op te zetten.

### 2.6.3 Beperkingen en Voorzichtigheid

Hoewel de voordelen duidelijk zijn, is het belangrijk om ML.NET-modellen met de nodige voorzichtigheid te gebruiken:

- **Suggesties in Plaats van Automatisering:** Het is raadzaam om de resultaten van ML.NET-modellen te gebruiken als referentie of suggestie, in plaats van volledig geautomatiseerde beslissingen te nemen. De modellen moeten als een hulpmiddel worden gezien, dat magazijnbeheerders helpt om beter geïnformeerde beslissingen te nemen, in plaats van de beslissingen volledig over te nemen. Dit voorkomt potentieel kostbare fouten die kunnen ontstaan door overmatige afhankelijkheid van geautomatiseerde systemen zonder menselijke controle.
- **Continue Monitoring en Verbetering:** Machine Learning-modellen hebben voortdurende monitoring en periodieke updates nodig om hun effectiviteit te behouden. Operationele omstandigheden veranderen, en modellen moeten regelmatig worden hergetraind met nieuwe data om relevant en nauwkeurig te blijven.

#### 2.6.4 Beperkingen en Negatieve Aspecten

Hoewel de voordelen duidelijk zijn, zijn er ook enkele beperkingen en negatieve aspecten die moeten worden overwogen:

- **Kosten van Onderhoud en Implementatie:** Het opzetten en onderhouden van Machine Learning-modellen kan kostbaar zijn. Dit omvat niet alleen de initiële ontwikkelingskosten, maar ook de voortdurende kosten voor data-acquisitie, modeltraining, en updates. Bedrijven moeten zich bewust zijn van deze doorlopende kosten en deze in hun budgetten opnemen.
- **Angst voor Vervanging:** Er is een reële angst onder werknemers dat Machine Learning hen zal vervangen. Dit kan leiden tot weerstand tegen de implementatie van nieuwe technologieën en kan ook een negatieve impact hebben op de moraal en het behoud van werknemers.
- **Verlies van Kennis:** Als Machines routinetaken overnemen, bestaat het risico dat menselijke kennis en vaardigheden in die gebieden verloren gaan. Dit kan problematisch zijn als de technologie faalt of als er behoefte is aan menselijke interventie bij complexe problemen.
- **Complexiteit van Implementatie:** De implementatie van Machine Learning in een WMS is complex en vereist gespecialiseerde kennis. Hoewel de ML.NET Model Builder mensen met beperkte kennis in staat stelt om Machine Learning-modellen te maken, blijft er een aanzienlijke behoefte aan deskundigen voor het analyseren van data, het opschonen en voorbereiden van datasets, en het interpreteren van de modelresultaten. Bedrijven die niet beschikken over interne expertise op het gebied van Machine Learning, kunnen moeite hebben met de implementatie en onderhoud van dergelijke systemen. Het is essentieel om te investeren in training en mogelijk externe expertise in te huren om deze technologie effectief te kunnen toepassen en onderhouden.
- **Afhankelijkheid van Technologie:** Een overmatige afhankelijkheid van technologie kan risicovol zijn. Als er technische storingen of problemen optreden, kan dit de operationele continuïteit verstoren. Het is belangrijk om altijd back-up plannen en systemen te hebben om dergelijke scenario's te vermijden.

### 3 Conclusie

---

In deze thesis heb ik de onderzoeksvraag onderzocht: "Hoe kan ML.NET worden toegepast in WMS-systemen om operationele efficiëntie, voorraadbeheer en besluitvormingsprocessen te verbeteren?" Door middel van uitgebreide literatuurstudie, experimenten en casestudies heb ik aangetoond dat Machine Learning, en specifiek ML.NET, een krachtig hulpmiddel kan zijn voor het optimaliseren van warehouse management systemen (WMS).

#### 3.1 Samenvatting van de Bevindingen

**Kwaliteit van Data:** Een van de belangrijkste bevindingen is het cruciale belang van kwalitatief hoogwaardige data. Machine Learning-modellen presteren alleen goed als ze gevoed worden met accurate, consistente en relevante data. Inconsistenties en onvolledigheden in data leiden tot onnauwkeurige voorspellingen en suboptimale resultaten.

#### 3.2 ML.NET biedt tal van voordelen voor WMS-systemen:

- **Integratie met .NET Ecosysteem:** Voor bedrijven die al een .NET-backend hebben, biedt ML.NET een naadloze integratie, wat de implementatie en het gebruik ervan vereenvoudigt.
- **Efficiëntie en Prestaties:** ML.NET is ontworpen voor hoge prestaties en kan snel en nauwkeurig Machine Learning-modellen trainen, wat essentieel is voor productieomgevingen.
- **Gebruiksvriendelijkheid:** De ML.NET Model Builder maakt het mogelijk voor ontwikkelaars met weinig tot geen Machine Learning-ervaring om effectieve modellen te bouwen en te implementeren.

#### 3.3 Praktische Implementaties:

Door de bespreking van praktische implementatievoorbeelden, zoals productlocatiebepaling en ordervoorspelling, heb ik laten zien hoe ML.NET specifiek kan worden toegepast om operationele processen binnen een WMS te verbeteren. Deze toepassingen hebben geleid tot verbeteringen in efficiëntie, nauwkeurigheid en besluitvorming.

#### 3.4 Casestudy

**Mercedes-Benz's Digital Twin Technology:** De casestudy van Mercedes-Benz heeft verder geïllustreerd hoe geavanceerde technologieën, zoals digital twins in combinatie met Machine Learning, aanzienlijke operationele voordelen kunnen opleveren, waaronder verbeterde pick-routes en optimale productlocatie.

#### 3.5 Kritische Analyse en Aanbevelingen

Hoewel de voordelen van het gebruik van ML.NET in WMS-systemen duidelijk zijn, moeten bedrijven voorzichtig zijn met de implementatie. Kwaliteitsdata is een noodzakelijke voorwaarde voor succes. Verder moeten de resultaten van ML.NET-modellen worden gebruikt als referentiepunten in plaats van volledig geautomatiseerde beslissingen. Menselijke controle en



supervisie blijven cruciaal om fouten te minimaliseren en de betrouwbaarheid van het systeem te waarborgen.

### **3.6 Conclusie**

Het onderzoek in deze thesis toont aan dat ML.NET een veelbelovend hulpmiddel is voor het verbeteren van de operationele efficiëntie, voorraadbeheer en besluitvormingsprocessen in WMS-systemen. Met de juiste data en een zorgvuldige implementatie kan ML.NET bedrijven helpen om hun magazijnbeheerprocessen te optimaliseren, kosten te reduceren en winst te verhogen. De technologie biedt een toegankelijke en krachtige oplossing voor bedrijven die al gebruik maken van een .NET-backend en op zoek zijn naar manieren om hun WMS te moderniseren en te verbeteren.

Door de voordelen van Machine Learning te benutten, kunnen bedrijven niet alleen hun huidige processen optimaliseren maar zich ook voorbereiden op toekomstige uitdagingen in het magazijnbeheer. Met ML.NET als betrouwbare partner kunnen bedrijven een strategisch voordeel behalen in een steeds concurrerender markt.

Ondanks de positieve aspecten zijn er ook uitdagingen en potentiële nadelen die niet over het hoofd mogen worden gezien. Bedrijven moeten deze overwegingen zorgvuldig afwegen bij de besluitvorming over de implementatie van Machine Learning in hun WMS-systemen.

## AI Engineering Prompts

---

- “Voor mijn thesis heb ik een gelijkaardige thesis gevonden, waar ik zeer gelijkaardige studies naar doe, hij doet dit over ML in een WMS systeem en ik doe dit meer specifiek over ML.NET, een Machine Learning framework, hoe die kan toegepast worden. Nu heeft deze een gelijkaardige introductie zoals ik deze ook zou doen. Kunt u deze even samenvatten zodat ik een leidraad heb, over wat ik precies moet vertellen, om het zelfde qua informatie te delen?”
- “Als volgende wil ik de Machine Learning voordelen aankaarten, en wat die als gevolgen kunnen hebben in een Machine Learning omgeving. Vermindering van kosten en fouten / Verbeterde klanttevredenheid / Verhoogde operationele efficiëntie: volgens : <https://www.q-nius.nl/nl/ai-en-Machine-Learning-in-logistiek-en-wms/> en : “Volgens analisten zal AI fouten verminderen bij overvoorraad tot wel 50% en voorraadvoorspellingen verbeteren tot wel 50%.” volgens <https://www.principallogisticstechnologies.com/wms-how-will-generative-ai-chatgpt-openai-change-warehouse-management-systems-software-2/> “
- “Nu wil ik het hebben in mijn thesis over " hoe dit waargemaakt kan worden", hoe je Machine Learning gebruikt om een WMS beter te laten werken en een betere magazijn voorraad etc. te hebben.”
- “Ook vermelden dat ML.NET een model builder heeft, het is een makkelijke UI die ervoor zorgt dat mensen die geen tot amper ML kennis hebben toch een Machine Learning model kunnen maken in mijn tekst.
- ‘Nu, voor paragraaf 5.4 "Implementatievoorbeelden", wil ik graag mijn twee uitgewerkte voorbeelden delen. Productlocatiebepaler:  
Deze applicatie voorspelt waar toekomstige bestellingen geplaatst moeten worden, op basis van historische data van eerdere bestellocaties. Het model gebruikt hiervoor de volgende informatie: Gewicht, Afmetingen, Breekbaarheid en Opslaglocatie in magazijn (zie figuur 4).
- “Ik heb een conclusie nodig voor mijn onderzoek, kan je helpen?”
- “Als einde van mijn experiment in mijn thesis moet ik kritische analyse doen van de resultaten. Ik wil hier duidelijk maken dat goeie data zeer belangrijk is en dat je met slechte data enkel slechte resultaten zult krijgen, maar dat het zeker mogelijk is en een goede optie is om ML.NET te gebruiken in een WMS systeem en dat al vele bedrijven hier winst mee hebben behaald en hun kosten hebben kunnen drukken. Ik wil ook vermelden dat het best is om de resultaten van de ML.NET modellen te gebruiken als referentie en niet automatisch zelf laten orders plaatsen en productlocaties verplaatsen, maar dat het een suggestie is dat ML.NET je vriend moet zijn.”

- “Ik moet ook een woord vooraf schrijven: mijn bachelorproef mentor en ideegever met functie: “Developer bij C&W Logistics”, naam: Kevin Verstraete, mijn familie die mij heeft geholpen met nalezen van dit document, het bedrijf C&W Logistics die me alle benodigdheden bezorgde en waar ik mijn bachelorproef grotendeels heb afgelegd.”
- “Ik wil in het volgende deel:  
 2.6.1 “Het Belang van Kwaliteitsdata” nog iets extra bespreken en verwerken in de huidige tekst indien het al aanwezig is :  
 bespreken dat het heel belangrijk is voor relevante data die de uitkomst kan beïnvloeden is nodig voor een correcte uitkomst, en geef een voorbeeld van:  
 verdere uitwerking van productlocatie voor order picker,  
 zware orders dichtbij, veel bestelde orders dichtbij, gemakkelijk voor ML om te berekenen en kan zo de wandelafstand beperken -> minder tijd verloren -> meer orders naar buiten in kortere tijd -> minder kosten.  
 En waarom die data zo belangrijk is, en voorbeeld geven van welke data zeker nodig zou zijn om dit te realiseren en verwijzen dat ik in mijn onderzoek het als basis heb gebruikt maar dat meer data betere voorspellingen = minder kosten etc.”

## 4 Referentielijst

---

- [1] E. Frazelle, World-Class Warehousing and Material Handling, 2nd ed., McGraw-Hill, 2016.
- [2] „How will Generative AI change Warehouse Management Systems?,” principallogisticstechnologies, [Online]. Available: <https://www.principallogisticstechnologies.com/wms-how-will-generative-ai-chatgpt-openai-change-warehouse-management-systems-software-2/>. [Geopend 3 Juni 2024].
- [3] „AI en Machine Learning in logistiek en WMS,” g-nius, 29 Maart 2024. [Online]. Available: <https://www.g-nius.nl/nl/ai-en-machine-learning-in-logistiek-en-wms/>. [Geopend 3 Juni 2024].
- [4] Arif Cam, „Global AI Survey: AI proves its worth, but few scale impact,” mckinsey, 22 November 2019. [Online]. Available: <https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-survey-ai-proves-its-worth-but-few-scale-impact>. [Geopend 3 Juni 2024].
- [5] „ML.NET,” microsoft, [Online]. Available: <https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet>. [Geopend 3 Juni 2024].
- [6] „Order pick routing Van theorie naar praktijk,” aibyexample, [Online]. Available: <https://www.aibyexample.be/solutions/order-pick-routing>. [Geopend 3 Juni 2024].
- [7] G. Sisinna, „How AI Transforms WMS and Warehousing,” 13 Februari 2024. [Online]. Available: <https://www.linkedin.com/pulse/how-ai-transforms-wms-warehousing-giovanni-sisinna-dh0tf/>. [Geopend 3 Juni 2024].

## Bijlagen

Github: <https://github.com/AaronWheezer/ML-BAP>

### Hoe maak je je eigen ML.NET predictie

1. Installeer ML.NET: Zorg ervoor dat je de ML.NET bibliotheek hebt geïnstalleerd. Dit kun je doen via NuGet Package Manager in Visual Studio of via de .NET CLI:

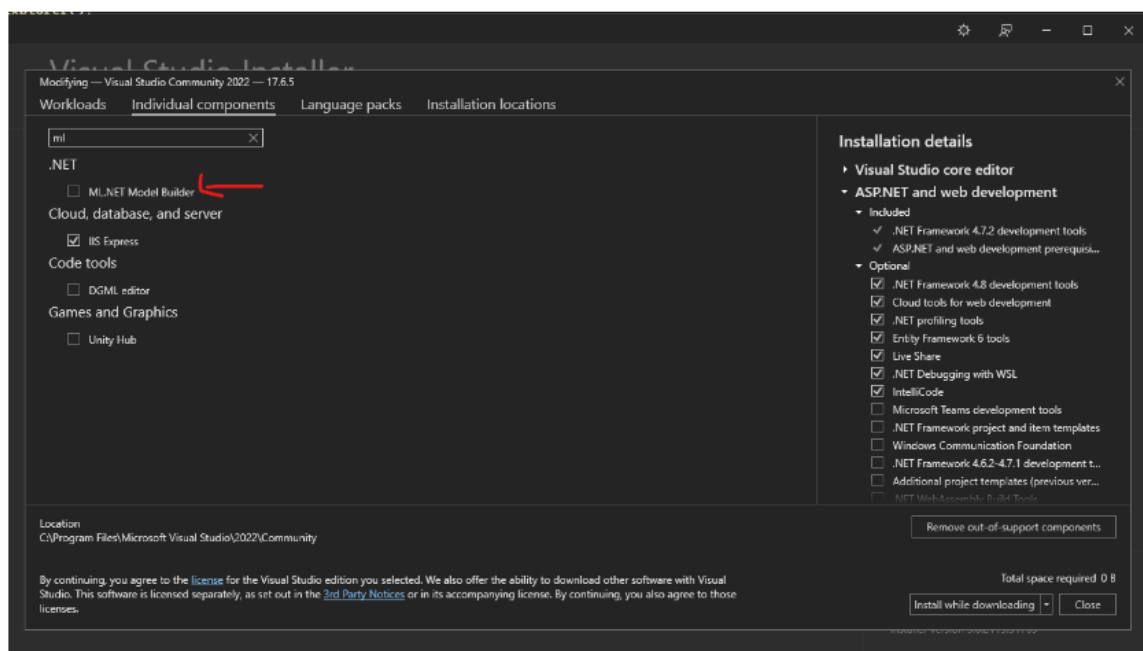
```
dotnet add package Microsoft.ML
```



2. Maak een model: Gebruik Model Builder in Visual Studio om een machine learning model te maken. Model Builder maakt het gemakkelijk om een model te trainen en te evalueren op basis van je dataset.

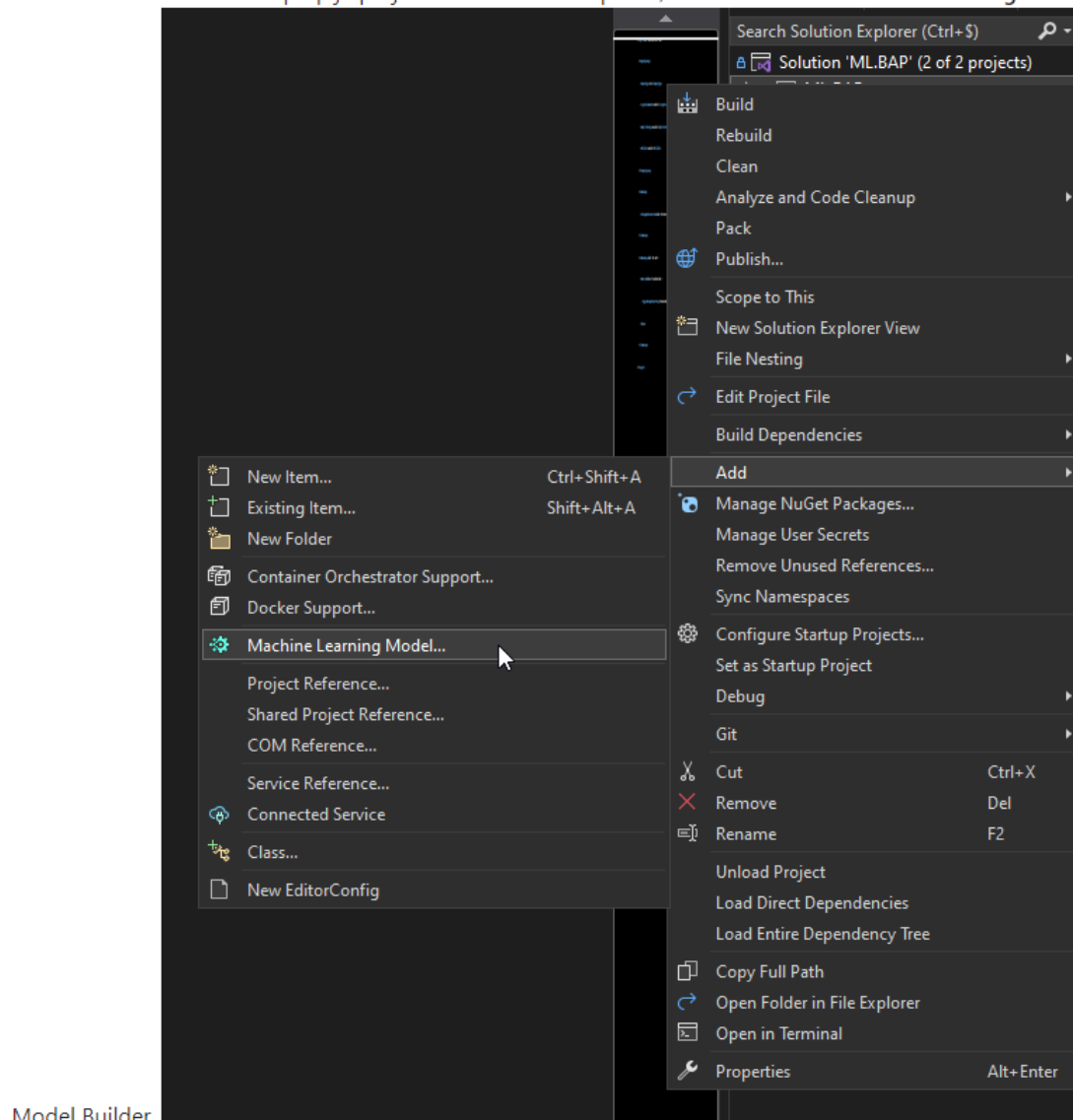
hoe te Model Builder te installeren:

Installeer de ML.NET Model Builder: Dit doe je door Visual Studio Installer te openen, je huidige Visua

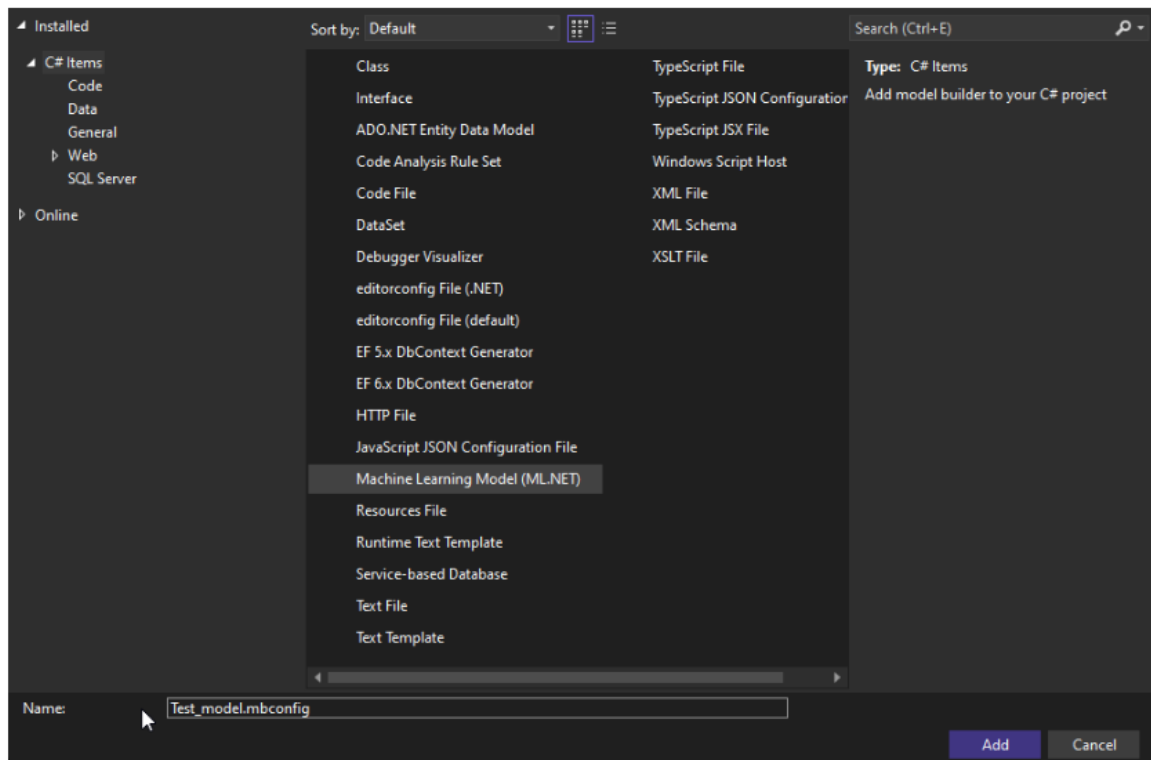


### 3. Open Model Builder

Klik met de rechtermuisknop op je project in de Solution Explorer, selecteer **Add > Machine Learning**. Dit opent de

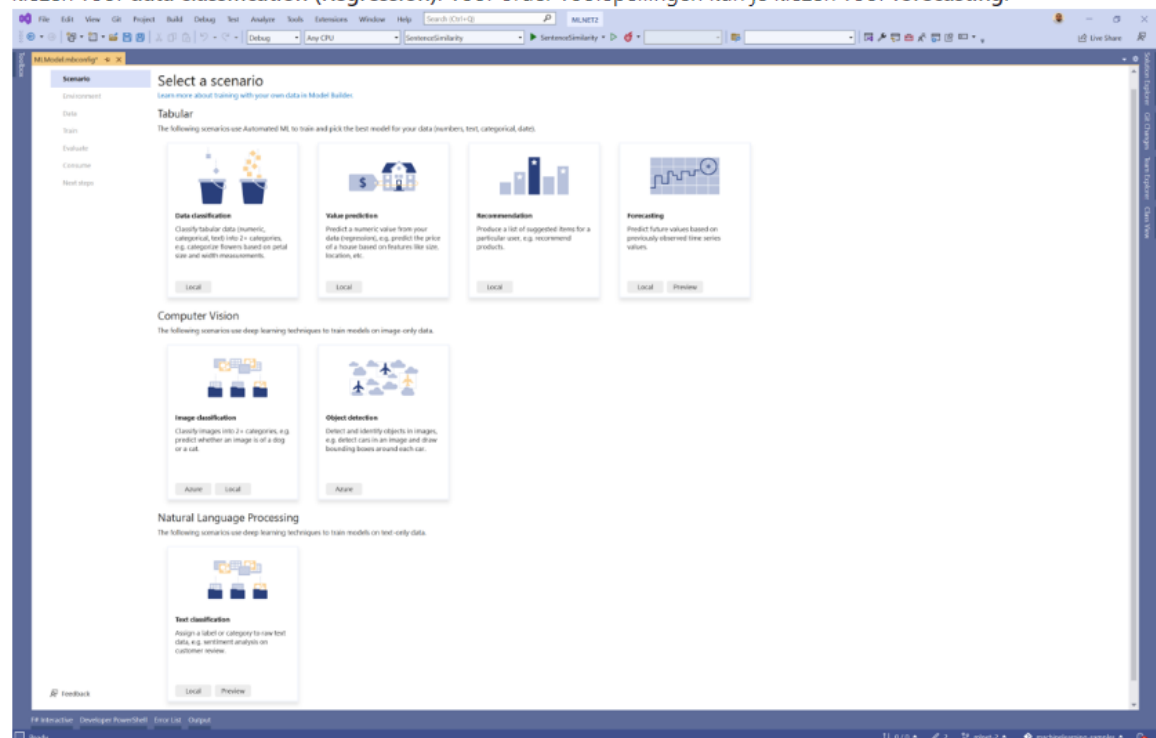


Model Builder.



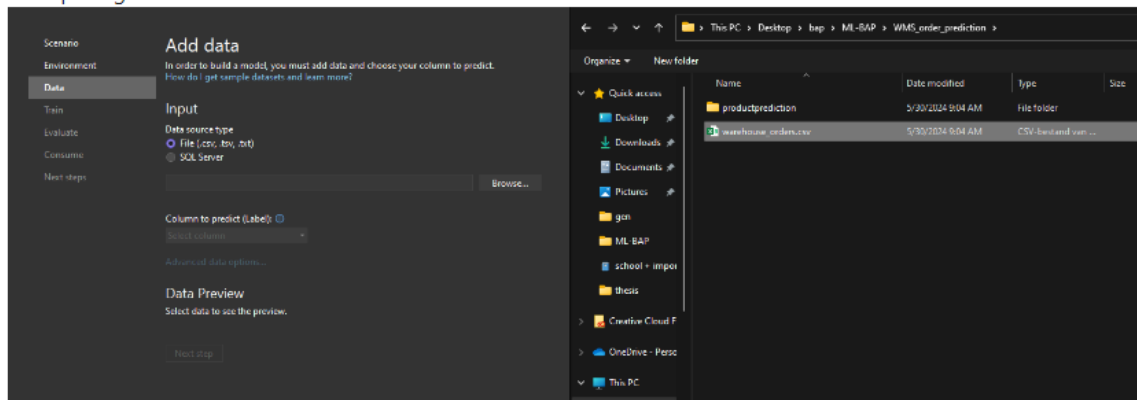
#### 4. Kies het scenario

Selecteer het scenario dat het beste past bij je voorspellingstaak. Voor locatie voorspellingen kun je bijvoorbeeld kiezen voor **data classification (Regression)**. Voor order voorspellingen kun je kiezen voor **forecasting**.



## 5. Voeg je data toe

Voeg je dataset toe. Dit kan een CSV-bestand zijn dat de historische gegevens bevat die je wilt gebruiken voor het trainen van je model. Zorg ervoor dat je dataset goed is voorbereid en relevante kolommen bevat voor je voorspelling.



## 6. Train het model

Kies hoe lang je Model Builder wilt laten trainen. Hoe langer het traint, hoe beter de prestaties van het model kunnen zijn. Model Builder zal verschillende algoritmen proberen en het beste model selecteren op basis van de prestaties.

```
Experiment output folder: C:\Users\aaaron\AppData\Local\Temp\AutoML-NNI\Experiment-CAK6MY
Forecasting
0, ForecastBySsa, RMSE: 12.6836072577456, MAPE: 0.30099734362301, 0.275
4, ForecastBySsa, RMSE: 12.6836072577456, MAPE: 0.30099734362301, 0.009
5, ForecastBySsa, RMSE: 10.3389426187351, MAPE: 0.306230585548937, 5.894
9, ForecastBySsa, RMSE: 12.6836072577456, MAPE: 0.30099734362301, 0.024
11, ForecastBySsa, RMSE: 13.2436767959948, MAPE: 0.300483986506878, 0.029
13, ForecastBySsa, RMSE: 48.7068927447803, MAPE: 1.41588327890343, 4.277
15, ForecastBySsa, RMSE: 7734.29516405595, MAPE: 230.172629744868, 3.47
17, ForecastBySsa, RMSE: 69.8062823549594, MAPE: 2.14043841304304, 4.291
18, ForecastBySsa, RMSE: 327.932732904102, MAPE: 9.84842964278469, 4.247
19, ForecastBySsa, RMSE: 12.7805440507612, MAPE: 0.360865366999021, 5.672

=====Experiment Results=====
|
| Summary
|
| ML Task: Forecasting
| Dataset: C:\Users\aaaron\Desktop\bap\ML-BAP\WMS_order_prediction\warehouse_orders.csv
| Label : Orders
| Total experiment time : 28.19 Secs
| Total number of models explored: 10
|
```



## 7. Evalueer het model

Bekijk de evaluatie resultaten. Model Builder zal je laten zien hoe goed je model presteert op basis van verschillende metrics zoals Mean Absolute Error (MAE), Mean Squared Error (MSE), en Root Mean Squared Error (RMSE).

### Evaluate

Results of training for your model can be found below.  
How do I understand my model performance?

**Best model:**

**RMSE:** 10.3389

**Model:** ForecastBySsa

**Try your model**

Horizon: 365

Predict

Prediction ☐ Table ☒ Graph

Orders	Orders_UB	Orders_LB
33.48481	80.23655	-13.26692
33.48312	80.23486	-13.26861
33.48145	80.23318	-13.27029
33.47975	80.23149	-13.27199
33.47805	80.22979	-13.27368

## 8. Gebruik het model

Na het trainen en evalueren, genereert Model Builder code om je model te gebruiken. Deze code kun je toevoegen aan je project om voorspellingen te doen.

```
// Load the model
MLContext mlContext = new MLContext();
IDataView dataView = mlContext.Data.LoadFromTextFile<ModelInput>(path: "your-data.csv", hasHeader: true);

// Train the model
var model = mlContext.Model.Load("model.zip", out var modelInputSchema);

// Use the model
var predictionEngine = mlContext.Model.CreatePredictionEngine<ModelInput, ModelOutput>(model);
var input = new ModelInput { ... };
var prediction = predictionEngine.Predict(input);

Console.WriteLine($"Predicted value: {prediction.PredictedLabel}");
```