# Voorspellingsmodel voor de fout in de voorraad

Dit document dient als leidraad voor het toepassen van het voorspellingsmodel dat een voorspelling doet met betrekking tot de aanwezig fout in de voorraad. Dit document is enkel bedoeld voor werknemers bij Euroma.

Thomas Hordijk

[thomasabelhordijk@gmail.com](mailto:thomasabelhordijk@gmail.com) +31614285316

1. Algemeen
   1. Werking

De algemene werking van het algoritme is hieronder beschreven door middel van een flowchart.

Het model importeert de data aangeleverd uit de datawarehouse. Dit zijn twee CSV bestanden, eentje bevat de ESA transacties, de andere de LN transacties. Het algoritme verwacht dat deze bestanden “ESA.csv” en “LN.csv” heten, als dit niet het geval is zullen de bestanden niet gevonden worden en het algoritme niet werken.

Na het in lezen van deze bestanden worden er eerste een aantal conversie stappen gemaakt om de data in het juiste format te hebben. Dit is nodig voor het samenvoegen en uitwerken van bepaalde velden.

De eerste stap is het omzetten van de locaties zoals deze in ESA genoteerd worden naar de magazijnen zoals deze in LN bekend zijn Vervolgens word de data ontdaan van alle artikelen die niet in het model worden meegenomen. Tot slot word de tijd notatie in de LN dataset omgebouwd naar een standaard notatie.

Wanneer alle ombouw vertaal stappen zijn voltooid word de data van de twee systemen samengevoegd. Na het samenvoegen worden de voorraadcorrecties verzameld. Deze dienen als start punt voor de fout analyse. Vanaf een correctie wordt de fout teruggerekend naar de correctie ervoor. Alle transacties van een artikel na de laatste correctie worden toegepast in het voorspellingsmodel. Het model wordt getraind op de data van voor de laatste correctie. De configuratie die gebruikt wordt bij het trainen is afgeleid uit de experimenten van het onderzoek. Na het trainen van het model word deze toegepast op de data na de laatste correctie.

A diagram of a flowchart

Description automatically generated

* 1. Documenten

De map van het algoritme ziet er als volgt uit. De code voor het algoritme bevind zich in de “.py” bestanden, de .docx is dit bestand, en daarnaast zijn er nog een aantal mappen.

A screenshot of a computer

Description automatically generated

* + 1. Ruwe Data verwerking

Het model maakt gebruik van twee documenten om de verwerking van de ruwe data te begeleiden deze documenten zijn te vinden in de map “Conversions”:

* ”sku\_loc\_list.csv”
* “esa\_ln\_locations.csv”

Het model is gefocust op artikelen in de buiten-, of binnen silo’s, of de miniload. De artikelen die worden meegenomen in het algoritme zijn te vinden in het bestand “sku\_loc\_list.csv”. Wanneer de ruwe data wordt binnengehaald word deze eerst gefilterd op transacties die betrekking hebben tot de artikelen in de locaties genoemd in dit bestand. Alle andere transacties in de datasets worden verwijderd uit het algoritme.

Voor de LN dataset is dit een simpele stap omdat hier de magazijnen aangegeven worden met de betreffende magazijn code. Voor de ESA dataset is dat niet zo, in ESA zijn de locaties verder gespecificeerd. Voor dat de data samen kan worden gevoegd dienen de locaties in ESA omgezet te worden naar locaties in LN, hiervoor wordt het tweede bestand gebruik, “esa\_ln\_locations.csv”. Hierin is beschreven wat het magazijn is voor elke locatie die in ESA is gevonden, voor de artikelen binnen de scope.

Wanneer het algoritme moet worden uitgebreid naar een grotere set artikelen dienen beide bestanden aangepast te worden. Het eerste bestand dient gewijzigd te worden om te zorgen dat het juiste artikel word uitgefilterd. Het tweede bestand zal moeten worden uitgebreid om de locaties waar dit artikel in ESA kan bestaan aan een LN magazijn te koppelen.

* 1. A table with numbers and letters

     Description automatically generatedOutput

Het algoritme schrijft het resultaat weg naar een CSV bestand in de map Output. Dit bestand heeft dezelfde structuur als het “sku\_loc\_list.csv” bestand en laat de voorspelde fout van een artikel in een magazijn zien. De eenheid van de voorspelling is kilogrammen, dit omdat de continue waarden in ESA en LN ook in kilogrammen worden weergegeven

Naast de voorspelling word ook het model dat resulteerde van de training geëxporteerd en opgeslagen in het mapje “Models”. Dit zou dan op een later moment opnieuw kunnen worden gebruikt voor het maken van een voorspelling. Deze functionaliteit zit niet in het algoritme. Er word een model getraind elke keer dat algoritme wordt aangezet.

Uit de simulaties in het uitgevoerde onderzoek is gebleken dat het sorteren op de grootste fout een goed begin punt is voor het kiezen van de artikelen om een telling op uit te voeren.

In het model wordt gebruik maakt van een “deminishing population”, dit houd in dat de set artikelen waar je uit kunt kiezen elke keer kleiner is. Artikelen die eerder in de cyclus zijn geteld worden pas opnieuw gekozen als de cyclus voorbij is.

Houd voor jezelf een bestand bij welke artikelen, wanneer geteld zijn om dit in stand te houden. In de simulatie is een cyclus van een week aangehouden en word er elke dag niet meer dan een uur aan tellen gespendeerd. De tijd die nodig is voor het tellen is per magazijn vastgesteld maar gebaseerd op een indicatie.

1. Gebruik

Het algoritme is geschreven door middel van Python. Om deze code te kunnen toepassen dient de gebruiker python 3.11.5 te installeren samen met de benodigde “libraries” met standaard functies. Python is gratis te installeren via de Microsoft store of van de python website.

Dit algoritme is ontwikkeld met Python 3.11.5, latere versies kunnen ook werken maar dit is niet gegarandeerd. Bij het tegenkomen van problemen bij de executie van het algoritme controleer of de Python versie overeenkomt. Dit simpel te doen door op start te drukken, te zoeken op “CMD”. In het prompt venster dien “python -V” ingevoerd te worden, dit geeft de geïnstalleerde python versie weer.

Naast Python zijn er een aantal sub-libraries die ook geïnstalleerd moeten worden. Python levert een standaard package manager mee bij installatie “pip”. Via pip zijn de andere vereiste libraries te installeren.

De libraries gebruikt zijn:

|  |  |  |
| --- | --- | --- |
| Library | Versie | Beschrijving |
| OS |  | Interactie met het systeem van de gebruiker word via OS geregeld, standaard mee geïnstalleerd met python |
| DateTime | 5.2 | Haalt systeem Datum/Tijd op voor tijdstamps |
| Numpy | 1.25.2 | Library voor arrays en wetenschappelijke berekeningen |
| Pandas | 2.1.0 | Library voor dataframes, vergelijkbaar aan SQL tabelen |
| Scikit-learn | 1.3.0 | Library voor machine learning modelen |
| Joblib | 1.3.2 | Library voor het opslaan van machine learning modellen om later weer te gebruiken |

* 1. Installatie

Het algoritme dient te worden afgetrapt vanuit een python IDE. De IDE gebruikt in dit onderzoek is PyCharm Community Edition 2023.2.1. Dit is open-source software die gedownload kan worden en geïnstalleerd. Onder aan de hieronder genoemde webpagina is de download link te vinden. Download de community edition zodat je niet gebonden bent aan een trial.

https://www.jetbrains.com/pycharm/features/

De IT afdeling heeft de installatie van dit programma, zoals alle andere, geblokkeerd op de Euroma computers. Neem contact op met IT om de installatie van dit programma te realiseren

Na het installeren van pycharm en het importeren van het project zijn de libraries die boven genoemd zijn nodig. Beweeg naar de terminal van de IDE Zie onderstaande afbeelding en voer de volgende regel in:

pip install numpy pandas scikit-learn joblib datetime

A screenshot of a computer

Description automatically generated

De achterliggende dependenties worden direct me geïnstalleerd. Na dat dit voltooid is kan het algoritme afgetrapt worden. Dit doe je door op het pijltje boven in de editor te drukken.

A screenshot of a computer program

Description automatically generated

* + 1. Fout meldingen

Mogelijk foutmeldingen zijn:

* Library niet geïnstalleerd – No module names xx
  + Zoek het pip install command voor de genoemde library

A black background with colorful lights

Description automatically generated with medium confidence

* Bestand bestaat niet
  + Afhankelijk van de getoonde fout controleer of de namen van de mappen en bestanden overeenkomen. Het model begint altijd met zoeken in de map waar “main.py” opgeslagen staat

A black background with blue and red lines

Description automatically generated

* Feature naam niet bekend, de naam van de kolomen in de code en de data komen niet ondereen, zoek de fout en verbeter deze

A blue and black background

Description automatically generated with medium confidence