4/10/2022

Daan Kleijn, Kasper Sikkema, Susan Bruggeling

Team 5

Advies Rapport Recommandatie Engine

In dit advies rapport wordt een aanbeveling gedaan voor een recommandatie engine. Het doel van de recommandatie engine is om de klanten van de op=op voordeelshop producten aan te bevelen waar ze ook in geïnteresseerd zijn.

Inhoudsopgaven

[**Inleiding** 2](#_Toc100568968)

[**Alternatieven** 3](#_Toc100568969)

[Samen gekochte producten 3](#_Toc100568970)

[Beschrijving van het Algoritme 3](#_Toc100568971)

[Logica 3](#_Toc100568972)

[Gender filtering 4](#_Toc100568973)

[Prijs filtering 4](#_Toc100568974)

[Datum filtering 5](#_Toc100568975)

[Logica 5](#_Toc100568976)

[Trending producten 6](#_Toc100568977)

[Logica 6](#_Toc100568978)

[**Onderzoek** 7](#_Toc100568979)

[**Conclusie** 8](#_Toc100568980)

[**Bronnen** 9](#_Toc100568981)

| **Onderdeel** | **Omvang** |
| --- | --- |
| [Titelpagina](https://www.scribbr.nl/stage/schrijf-een-perfect-adviesrapport/#titelpagina-van-je-adviesrapport) | 1 A4 |
| [Inhoudsopgave](https://www.scribbr.nl/scriptie-tips/automatische-inhoudsopgave-maken-met-word/) | 1 A4 |
| [Inleiding](https://www.scribbr.nl/stage/schrijf-een-perfect-adviesrapport/#inleiding-van-je-adviesrapport) | 1 A4 |
| [Onderzoek](https://www.scribbr.nl/stage/schrijf-een-perfect-adviesrapport/#onderzoek-in-een-adviesrapport) | 1 A4 |
| [Alternatieven](https://www.scribbr.nl/stage/schrijf-een-perfect-adviesrapport/#alternatieven-oplossingen-maatregelen-of-scenarios) | max. 2 A4 per alternatief |
| [Conclusie](https://www.scribbr.nl/stage/schrijf-een-perfect-adviesrapport/#conclusie-het-advies-in-je-adviesrapport) | 1 A4 |
| [Bronnen](https://www.scribbr.nl/apa-stijl/literatuurlijst-volgens-de-apa-regels/) | – |

# **Inleiding**

De op=op voordeelshop is een webshop met een groot en divers assortiment. Sommige producten zijn interessant voor klanten van de webshop, maar het overgrote deel niet.

De op=op voordeelshop wil graag meer producten verkopen. Wanneer ze goede producten aan hun klanten aanbevelen, zullen ze meer producten verkopen. Echter, wanneer er aan een klant veel aanbevelingen worden gedaan die niet interessant, loop je het risico die klant te verliezen. Met het oog op de grote variëteit van producten in de webshop, is het daarom belangrijk dat de aanbevelingen goed passend zijn.

In dit rapport hebben wij meerdere recommandatie engines ontworpen. Wij hebben deze ontwerpen gemaakt aan de hand van de verkregen data van de op=op voordeelshop.   
Wij hebben de recommandatie engines gevalideerd in een testomgeving om de juistheid van de recommandaties te bepalen. [hier komen resultaten]   
  
In dit rapport gaan we eerst in op de recommandatie engines. We beschrijven we per recommandatie engine:   
- hoe het ontwerp tot stand is gekomen

* de werking van het algoritme
* het logische framework
* de resultaten van de validatie
* de voordelen
* de nadelen

Daarna bespreken we het opzetten van de validatie test. Vervolgens bespreken we de resultaten hiervan. Daarna brengen we een advies uit aan de hand van de resultaten. Tot slot is er een discussie.

# **Alternatieven**

Wij hebben 5 verschillende recommandatie engines met elkaar vergeleken. Deze komen stuk voor stuk aan bod.

## Samen gekochte producten

Producten die vaak samen gekocht worden samen met een product, zouden goede aanbevelingen kunnen zijn voor klanten die dat product bekijken. Wij hebben dit algoritme ontwikkeld en in een testomgeving getest.

### Beschrijving van het Algoritme

Zoekt alle producten die samen met een gegeven product zijn gekocht. Telt per opgezochte product het aantal unieke bestelling met het gegeven product. Geeft de producten met de hoogste aantallen unieke bestellingen terug.

### Logica

Een korte toelichting van het logische framewerk achter dit algoritme.

#### Predicaten

I(x, y) = x vindt y interessant [heeft kopje nodig]  
B(x, y) = x bekijkt y  
KS(x, y, z) = x koopt y en z

#### Formules

ϕ (interesse bekeken product): (x, y): B(c, p) → I(c, p)   
ψ (interesse samen gekocht): (x, y, z, a): (I(c\_1, p\_1) /\ KS(c\_2, p, p\_2)) → I(c\_1, p\_2)

#### Logische Framework

Ω (interesse gekocht met bekeken): (B(c\_1, p\_1) /\ ϕ (c\_1, p\_1) ψ (c\_1, c\_2, p\_1, p\_2)) → I(c\_1, p\_2)

#### Verzamelingenleer

Venn diagram

Description automatically generated with medium confidence

De verzameling met aanbevolen producten bestaat uit OPS – P.

[Afwegingen? Of iets]

[resultaten]

[voordelen]

[nadelen]

## 

## Gender filtering

Hier komt het logische framework van gender filtering

[Afwegingen? Of iets]

Deze recommandatie is gebaseerd op de gender tag die de dataset van de op=op voordeelshop bevat. Het idee van dit algoritme is dat een gebruiker producten koopt en deze producten hebben een tag of het voor een man of vrouw is bedoeld. Met dit algoritme kijken we eerst of het huidige product voor een man of vrouw is bedoeld, vervolgens bieden wij een ander product uit het assortiment van de op=op voordeelshop. Dit is een basis algoritme waarmee we verder kunnen bouwen, een ander algoritme wat wij kunnen ontwikkelen is bijna gelijk met deze maar iets meer geavanceerd. Het idee is dat het gender nog steeds wordt uitgezocht en aanbevolen maar dan bevelen wij producten met dezelfde prijs met een marge van x% of die in dezelfde prijscategorie zitten. Nu is alleen nog besproken hoe wij iets gaan aanbevelen als de klant in een productpagina zit, waar wij ook naar gaan kijken is of een gebruiker meer mannelijke of vrouwelijke producten koopt waardoor wij ook op de voorpagina van de op=op voordeelshop producten kunnen aanbevelen zonder dat de gebruiker in een sessie zit.

[resultaten]

[voordelen]

[nadelen]

## Prijs filtering

Hier komt het logische framework van prijs filtering

[Afwegingen? Of iets]

Dit is een simpel algoritme wat in de product-pagina gebruikt kan worden, het zoekt andere producten in dezelfde prijscategorie. Deze producten moeten dan wel in dezelfde categorie zitten, we willen de juiste aanbevelingen geven en een willekeurig product maar wel met dezelfde prijs is geen goede aanbeveling. Om een nog betere aanbeveling te kunnen geven zullen wij dit algoritme combineren met andere algoritmes zoals Gender Filtering, hoe meer data wij kunnen verwerken hoe beter de aanbeveling die wij aanbieden past bij het huidige product.

[resultaten]

[voordelen]

[nadelen]

## Datum filtering

Dit algoritme zoekt producten die tijdens een maand significant meer verkocht worden dan tijdens andere maanden.   
Sommige producten zijn in bepaalde maanden van het jaar een interessant product terwijl ze dat de rest van het jaar niet zijn. Neem zonnebrand bijvoorbeeld. In de zomermaanden zullen mensen hier een stuk geinteresserder in zijn dan de rest van het jaar. Wij hebben het identificeren van deze producten geautomatiseerd.

### Beschrijving van het Algoritme

Het algoritme gaat na hoe vaak een product in een unieke bestelling besteld is. Het gemiddelde aantal bestellingen per maand wordt hier vervolgens mee berekent.

Vervolgens wordt het aantal bestellingen per maand geteld. Per maand wordt afgegaan of het aantal verkochte producten significant meer dan gemiddeld is in die maand.   
Een significant aantal meer is 140% + 50 van het orginele aantal. Er is een instapdrempel van 50 voor producten die extreem weinig worden verkocht.   
Alle maanden waarin het product significant meer wordt verkocht worden opgeslagen onder de column *popular\_month* in de *products* tabel in de database.

Om aanbevelingen te doen op basis van dit algoritme, wordt een aantal producten die populair zijn in de huidige maand uit de database opgehaald. Deze worden gebruikt als recommandatie aan de klant.

### Logica

Een korte toelichting van de logica waar dit algoritme op berust.

#### Predicaten

G (x, y) = over alle jaren heen gerekend is x per maand gemiddeld y keer verkocht  
GM (x, y, z) = over alle jaren heen gerekend is x, in maand z, gemiddeld y keer verkocht.   
M (x) = het is maand x.  
I (x) = x is interessant.  
PM (x, y) = tijdens maand x, geteld over alle jaren, wordt y meer dan gemiddeld gekocht.  
IM (x, y) = y is interessant tijdens maand y.

#### Formules

ϕ (significant verkeer) (m, p): (GM (p, a\_1, m) /\ (G (p, a\_2) /\ (a\_1 >= (a\_2 \* 1.4 + 50)))  
ψ (Populaire maand) (m, p): ϕ(m, p) → PM (m, p)   
ω (Interessante maand) (m, p): ϕ (m, p) /\ ψ (m, p) → IM (m, p)

#### Logische Framework

ζ (datum filtering) (m, p): M(m) /\ ϕ (, p) /\ ψ (m, p) /\ ω (m, p)) → I(p)

#### Verzamelingenleer

Shape

Description automatically generated

De verzameling met aanbevolen producten bestaat uit PM

[Afwegingen?

Dit algoritme kan waarschijnlijk nog veel verder geoptimaliseerd worden. Je wil veel producten, die uberhaubt niet veel verkocht worden liever niet meenemen. Hoe dit nu opgelost wordt, is door aan de maatstaaf + 50 toe te voegen. Je kan ook een betere selectie aan producten aan het algoritme voeden, waardoor de +50 overbodig zou worden en er minder producten gecontroleerd hoeven te worden. Dit zou het programma aanzienlijk sneller maken, al is het nu met 16 secondes geen ramp.

[resultaten]

[voordelen]

[nadelen]

## Trending producten

[beschrijf in een zin] [tot stand komen van het algoritme]

### Beschrijving van het Algoritme

Het algoritme gaat na hoe vaak een product in een unieke bestelling besteld is in het afgelopen jaar. Het gemiddelde aantal bestellingen per maand wordt hier vervolgens mee berekent.

Vervolgens wordt het aantal bestellingen over de afgelopen week geteld. Daarna wordt er afgegaan of het aantal verkochte producten significant meer dan gemiddeld is in de afgelopen week.   
Een significant aantal meer is 140% + 50 van het originele aantal. Er is een instapdrempel van 50 voor producten die extreem weinig worden verkocht.   
Een aantal product die significant meer zijn verkocht in de afgelopen week worden teruggegeven door het algoritme als recommandaties.

### Logica

Hier wordt het logische framework van trending producten beschreven.

#### Predicaten

G (x, y) = over het afgelopen jaar heen is x gemiddeld y keer per week verkocht.  
V (x, y, z) = x is y keer verkocht over week z.   
PW (x, y) = tijdens week x wordt y meer dan het jaarlijkse gemiddelde verkocht.  
W (x) = week x was afgelopen week.  
I (x) = x is interessant.

#### Formules

ϕ (Significant verkeer) (w, p): V (p, a\_1, w) /\ (G (x, a\_2) /\ (a\_1 >= (a\_2 \* 1.4 + 50))  
ψ (Populaire week) (w, p): ϕ (w, p) → PW (w, p)  
ω (interessante week) (w, p): PW(w, p) → I(p)

#### Logische Framework

ζ (datum filtering): (W (w) /\ ϕ (w, p) /\ ψ (w, p) /\ ω (w, p)) → I (p)

#### Verzamelingenleer

[Afwegingen? Of iets]

[resultaten]

[voordelen]

[nadelen]

# **Onderzoek**

Wij hebben onderzoek gedaan naar welke recommandatie algoritmes het beste passen voor de gebruikers met de data die beschikbaar is van de op=op voordeelshop webshop.

Wij trachten dit onderzoek te beantwoorden door vijf recommandaties te ontwikkelen die data van de producten en van de users ter hart nemen om de meest passende aanbevelingen te geven aan de klant. Vervolgens willen wij deze in een testomgeving testen zodat wij simulaties kunnen maken en onze aanbevelingen kunnen testen en de benodigde data verzamelen zodat we een juist advies kunnen ontwikkelen en een goed product kunnen leveren.

[Validatie resultaten komen hier]

[korte conclusie]

# **Conclusie**

# **Bronnen**