# Business rules voor Recommendation Engine

## Content-Filtering



Een recommendation engine die werkt met content filtering, bekijkt de items die een klant leuk vindt en de keywords die deze klant gebruikt om een item te beschrijven en baseert op deze data welk product de klant ook leuk kan vinden. Ook wordt gekeken naar de voorkeuren van de klant. De afbeelding hiernaast weergeeft dit op een simpele wijze. Klant ‘K’ heeft bijvoorbeeld een rode tas gekocht van een merk ‘X’. Mogelijk is bijvoorbeeld om nu andere producten van merk ‘X’ aan te bieden, andere rode producten of tassen van andere merken.

## Mijn recommendation

Mijn Recommendation Engine, ‘RE’, zal kijken tot welk categorie een product behoort. Als een profiel een product heeft aangeschaft, zal deze RE twee producten behorend tot dezelfde sub\_sub\_categorie aanraden, een product van dezelfde sub\_categorie, en een product van dezelfde categorie.

Diagram

Description automatically generatedHet onderstaande diagram beeld het uit.  
Voorbeeld: Klant ‘K’ heeft product ‘1’ gekocht, behorend tot een bepaald sub\_sub\_category, ‘X’. X = Nagellak. Klant ‘K’ krijgt vervolgens producten ‘2’ en ‘3’ uit ‘X’ aangeraden. Y = Make-Up (sub\_category). ‘K’ krijgt product ‘4’ aangeraden uit ‘Y’. Z = Make-up & geuren (category). ‘K’ krijgt ook product ‘5’ aangeraden uit ‘Z’.

Wel krijgen we te maken met een dataset-probleem: niet elk product behoort tot een sub\_sub\_category, en zelfs sommige producten niet tot een sub\_category en category. Deze recommendation engine zal dus niet werken voor deze producten.

Het voordeel van deze RE is dat we op verschillende ‘dieptes’ aanbevelingen kunnen doen aan de klant. Zoals te zien is in het bovenstaande plaatje, zijn er drie ringen, elk breder dan de ander, waarin naar producten gezocht worden. Zo kunnen we de klant een breed college aan aanbevelingen bieden.

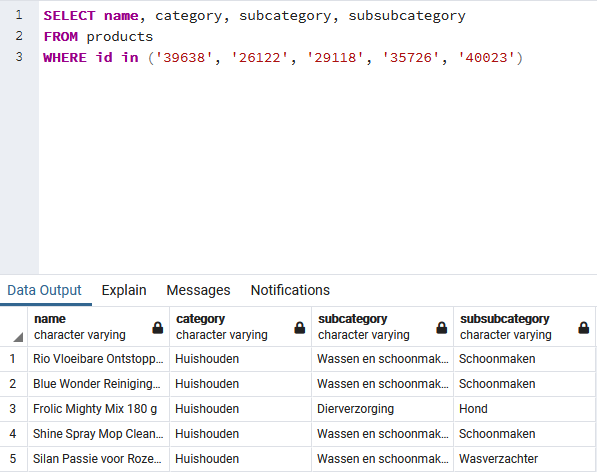
## Toepassing

Deze RE passen we toe op een product-detail pagina. Er wordt gebruik gemaakt van de tabel ‘products’ in de database. In de postgreSQL database wordt het product gezocht. Vervolgens worden van dit product de volgende gegevens geëxporteerd naar het python programma: de categorie, subcategorie en subsubcategorie. Als een van deze of deze allemaal niet bestaan, wordt er simpelweg ‘None’ geëxporteerd. Hier wordt in het algoritme rekening mee gehouden. Als we de categorieën hebben, kunnen we de database doorzoeken voor andere producten uit deze categorieën. Tot slot pakken we hier twee willekeurige producten uit het corresponderende subsubcategorie, een uit het subcategorie en een uit de categorie. Hieronder staat een voorbeeld, toegepast op een van de producten.

## Controle



Hier zie je de RE toegepast op het product ‘Shine Spray Mop Cleaning Towel 1 st’. Eronder staan onder het kopje ‘Soortgelijke producten’ vier producten aanbevolen. Deze zijn zoals boven beschreven uit dezelfde categorieën gehaald, zoals hieronder te zien is.



Uit deze afbeelding kunnen we zien dat het originele product, nummer 4 op de afbeelding, tot subsubcategorie ‘Schoonmaken’ behoort. Hier matchen we twee producten mee: product 1 en 2.

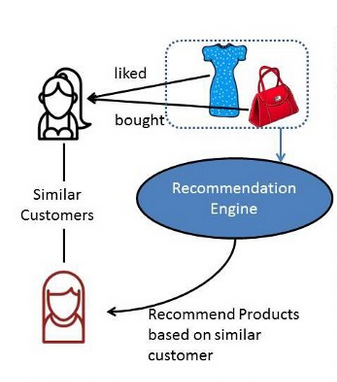
Daarnaast hebben we een product met dezelfde subcategorie: product 5. Echter behoort dit product tot subsubcategorie ‘Wasverzachter’, om de klant een bredere recommendation te kunnen bieden.

Tot slot hebben we product 3, een product met slechts dezelfde categorie. Dit product behoort tot ‘Huishouden’, maar is nog breeder. In dit geval gaat het om hondenvoer.

## Problemen

Het probleem met deze RE is dat niet elk product een categorie heeft, laat staan subsubcategorie. Als dit het geval is laat de RE momenteel geen producten zien. Daarnaast is deze RE soms te breed vanwege de willekeurige keuze uit de categorieën.

## Collaborative-filtering



Een recommendation engine die werkt met collaborative-filtering, bekijkt het gedrag, de activiteiten of voorkeuren van een klant en voorspelt met deze data wat de klant leuk zal vinden. Dit gebeurt door deze data te vergelijken met andere klanten. De afbeelding hiernaast weergeeft dit op een simpele wijze. De klant vond een jurk en tas leuk, die een vergelijkbare klant ook leuk vond. De RE kijkt vervolgens wat de vergelijkbare klant nog meer leuk vond en beveelt dit aan aan de klant.

## Mijn recommendation

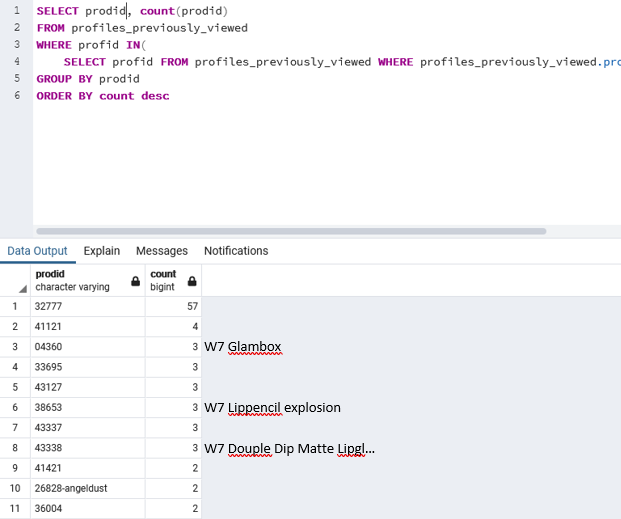
Mijn RE maakt gebruik van alle ‘eerder bekeken’ data. Op het moment dat een klant op een product klikt, moet er in de profiles\_previously\_viewed tabel alle profielen verzameld worden die op dat product hebben geklikt. Vervolgens moeten van al deze verzamelde profielen alle eerder bekeken producten verzameld worden. Deze producten moeten elk geteld worden en de meest bekeken producten moeten voorgesteld worden aan de klant.

## Toepassing

Deze RE passen we ook toe op de product-detail pagina. Er wordt gebruik gemaakt van de tabellen ‘profiles\_previously\_viewed’ en ‘profiles’ in de database. In de postgreSQL database wordt het product gezocht. Vervolgens worden van dit product de profielen die daarop hebben geklikt verzameld. Deze profielen worden vervolgens nogmaals in een sql-query gezet, om van alle verzamelde profielen de kliks te verzamelen. Nu we alle producten hebben, hebben we nog een counter nodig. Zo kunnen we zien welk product het meest is geklikt en een top 11 ervan verzamelen. Het enige wat nog nodig is, is om het originele product uit deze lijst te halen. Dat laat ons met een top 10 producten, waar andere gebruikers het meest op hebben geklikt.

## Controle

In dit geval gebruiken we een willekeurig product. Dit is W7 Lip Legend Matte Topcoat. Dit is een lipstick. In de volgende afbeelding kun je het product zien met de Recommendation Engine in werking. 4 andere, erg soortgelijke producten, worden vervolgens aangeboden. In de query eronder is te zien hoe de lijst eruitziet waaruit de producten gekozen worden.



## Problemen

Het probleem met deze RE is dat niet van elk product er klikgegevens zijn. In dat geval wordt teruggevallen op de Content-Filter. Als de Content-Filter geen producten doorgeeft, wordt er momenteel niets weergeven.