Verslag 1 – Collectieve Intelligentie

Groep 4: Dennis de Buck, Cesar Groot Kormelink, Enzo Delaney-Lamour en Maik Larooij

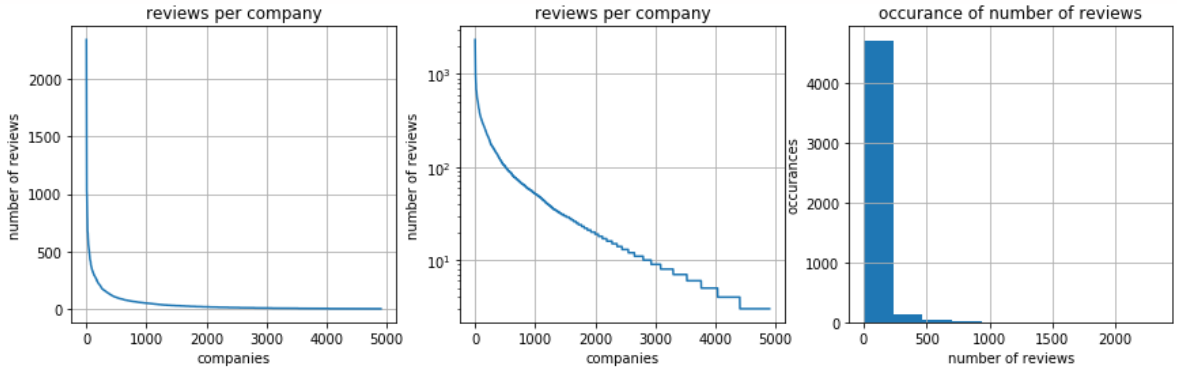
1.0 Taakverdeling

Er is besloten om elke week minstens één keer contact op te nemen via Zoom. Op deze manier houden we elkaar een beetje op de hoogte van de vorderingen en kunnen we werken aan het verslag van die week. Voor het programmeren en evalueren van de algoritmes zijn er twee subgroepen gemaakt. Enzo en Dennis storten zich op het content based algoritme. Maik en Cesar gaan aan de slag met het collaborative filtering algoritme. Wel is afgesproken om elkaar waar nodig zo goed mogelijk te helpen bij problemen en om elkaar goed op de hoogte te houden van vorderingen. Voor ons betekent dit dus volgende week het werken aan het algoritme en het schrijven van het voortgangsverslag. De laatste week zal dan aan de evaluatie worden besteed.

1.1 Structuur en verdeling van de data

Review data

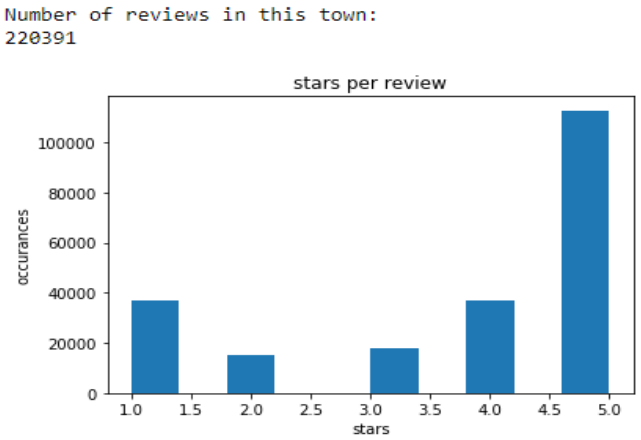
De review data bestaat ten eerste uit een unieke review id en de datum van publicatie van de review. Er wordt aangegeven welke gebruiker op welke business een review geeft door middel van een user en business id. De inhoud van deze review is opgeslagen in tekst en aantal sterren. Daarnaast wordt voor iedere review aangegeven hoeveel andere gebruikers de review als ‘useful’, ‘funny’ of ‘cool’ beoordelen.



Figuur 1. Verdeling van de review data

De meest linker plot geeft aan hoeveel reviews er per bedrijf zijn. De middelste plot laat hetzelfde zien op een logaritmische schaal. We hebben hier te maken met een ‘long tail’. Als je naar de logaritmische plot kijkt, zijn er relatief veel bedrijven (>2000) met minder dan 10 reviews. Over deze bedrijven weten we op basis van reviews weinig.

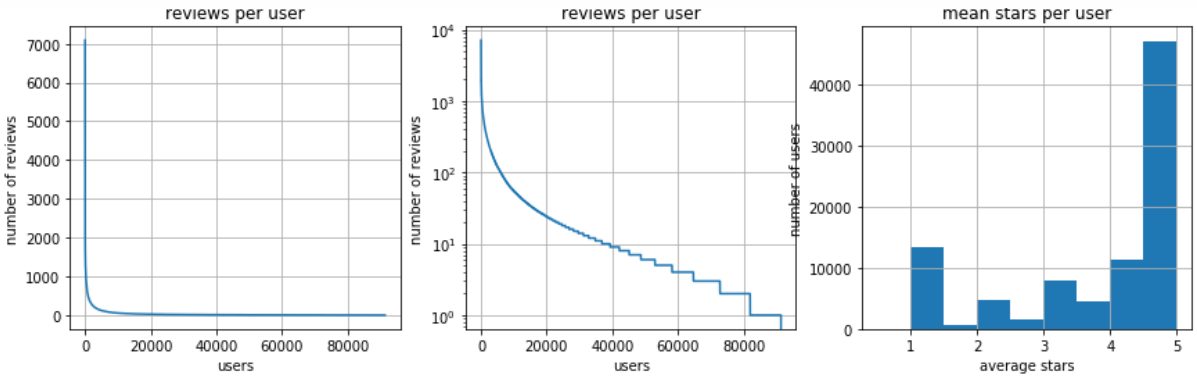
In deze plot is te zien hoe vaak een bepaald aantal gegeven sterren voorkomt. Hier is dus te zien dat ongeveer de helft van alle reviews 5 sterren heeft. Met beide problemen moet rekening worden gehouden.



Figuur 2. Verdeling van de review data

User data

De user data bestaat uit user ID's en namen met het aantal reviews geschreven door de user. Er wordt aangegeven sinds wanneer de user actief is op yelp en een lijst van users als user ID's die zijn toegevoegd als vriend. Ook is er aangegeven hoe vaak reviews als 'useful' 'funny' en 'cool' zijn beoordeeld. Daarnaast zijn er andere features te zien zoals het aantal 'fans' van de user, de datums dat de user 'elite' was, zijn/haar gemiddelde rating van alle reviews en allerlei soorten complimenten zoals 'hot', 'profile', 'cool' en 'funny', die de user heeft ontvangen van anderen.

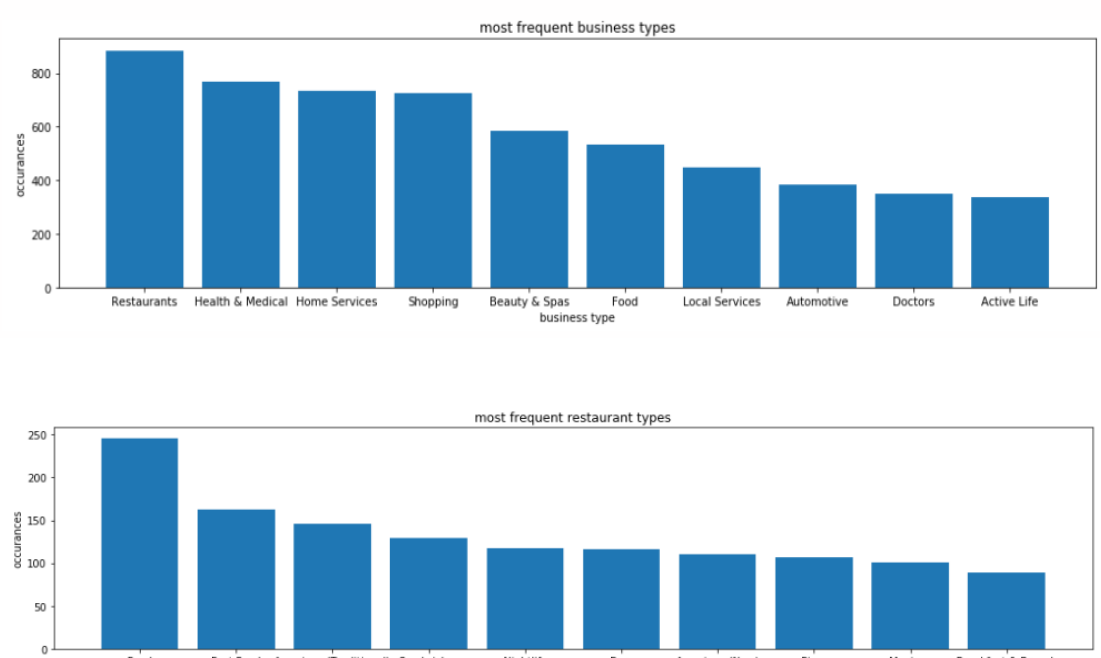


Figuur 3. Verdeling van de user data

De linker plot geeft aan hoeveel reviews er zijn per user. Op de middelste logaritmische schaal is te zien dat ongeveer de helft van de users meer dan 10 reviews heeft. Zo'n 30.000 users hebben minder dan 5 reviews. Ook is op de rechter plot te zien dat het aantal gebruikers met een gemiddelde rating van 5 erg hoog ligt. Deze gegevens kunnen van invloed zijn op de nauwkeurigheid van ons systeem, en is het handig om ook met deze 2 problemen rekening te houden.

Business data

Iedere business heeft een eigen rij in een tabel. In het tabel staat de naam, business-id en de locatie opgeslagen. Verder heeft het tabel nog meer locatiegegevens en de gemiddelde rating, het aantal ratings en een binaire waarde of het bedrijf open is. Ook staan in het tabel specifieke features van het restaurant met een True False waarde of het bedrijf dit heeft. Ook staan er categorieën van ieder bedrijf opgeslagen zoals restaurant, flowers of shopping. Als laatste staat er opgeslagen wat de openingstijden zijn. Dit tabel heeft voor iedere kolom een feature en als rijen staan er de verschillende bedrijven.



Figuur 4. Verdeling van de business data

De bovenste plot laat zien wat de meest voorkomende categorieën zijn van alle bedrijven in deze stad. Te zien is dat de meeste bedrijven vallen onder de categorie ‘restaurants’, om nog wat dieper te kijken naar de categorieën laat de tweede plot zien wat de meest voorkomende type restaurants zijn. Het totaal aantal restaurants waar de tweede plot op is gebaseerd is dus het aantal wat uit de bovenste plot komt, 885. De categorieën zijn dus gebaseerd op 885 restaurants. Wel hebben restaurants vaak meerdere categorieën. Restaurants met geen categorieën of maar 1, bijvoorbeeld alleen ‘restaurant’ zijn niet zo bruikbaar. De vraag rees dus hoeveel van dit soort restaurants voorkwamen in de gehele dataset. Na een test zijn we er achter gekomen dat er slecht 4 van de 885 restaurants niet bruikbaar zijn. Deze worden op dit moment al automatisch weggehaald door ons algoritme: de lege categorieën worden niet meegenomen met het filteren van restaurants en ook de categorie restaurant wordt in het proces weggehaald om alleen te focussen op de typen restaurants.

```
Number of restaurants:
885
Number of restaurants with 0 or 1 categories:
4
```

Figuur 5. Aanvulling verdeling business data

1.3 Verwachtingen

Onze verwachting is dat collaborative filtering lastig is op deze dataset. Dat komt omdat er veel verschillende bedrijven in de set zitten. Zo kan het voorkomen dat iemand die een nagelsalon zoekt een restaurant aangeraden krijgt omdat andere gebruikers waar diegene ‘op lijkt’ vooral restaurants hoog hebben beoordeeld. Hierom zal item based naar verwachting beter werken dan user based. Item based filtering is wel lastig te implementeren, omdat er veel bedrijven zijn met minder dan 10 reviews. Wanneer de data gesplitst wordt in bijvoorbeeld alleen restaurants, zal user based CF waarschijnlijk beter werken. Er zijn relatief veel gebruikers met reviews wat de implementatie van dit algoritme makkelijker maakt. Door alleen de focus te leggen op de subset van restaurants is het hierboven omschreven probleem deels verholpen.

De verwachting is dat content based filtering goed zou werken omdat we eerder hebben uitgelegd dat collaborative filtering, vooral user-based, wat minder zou werken omdat we in de dataset te maken hadden met verschillende categorieën aan bedrijven. Een nagelsalon en een kiprestaurant kunnen dan een hoge similarity vertonen omdat ze toevallig ongeveer dezelfde ratings hebben gekregen. Via content based filtering kan dit probleem verholpen worden aangezien hier juist gefocust wordt op categorieën. Op deze manier worden er nagelsalons aangeraden als de gebruiker positieve interesse toont in nagelsalons. Bij content based wordt er dus vergeleken op basis van de inhoud van het bedrijf, wat in het geval van de verschillende categorieën van deze dataset een voordeel is.

1.4 Plannen

De volgende twee algoritmes zullen worden uitgewerkt:

- Content based filtering op basis van op basis van categorieën (Jaccard similarity) en locatie. Dus de restaurants die een gebruiker goed vindt zullen worden vergeleken met andere bedrijven op de aangegeven aspecten en de bedrijven die hier het meest op lijken zullen aanbevolen worden. Daarnaast zal een bedrijf alleen aanbevolen worden als het ook geopend is aangezien het anders helemaal geen relevante optie is.
- Item-based collaborative filtering, waarbij op basis van reviews een neighborhood van het bedrijf wordt gevormd met gelijke bedrijven. De bedrijven in deze neighborhood worden verrekend met de ratings van de gebruiker waar voorspelt voor wordt om zo een voorspelde rating te krijgen. Op basis van deze rating en een treshold kan dan worden beslist om dit bedrijf aan te raden of niet.

Als je een content based filtering algoritme toepast moet je rekening houden met het feit dat er van veel restaurant niet is aangegeven of een feature van toepassing is, waardoor restaurants soms niet worden meegerekend voor een bepaalde feature. We moeten dus gaan bepalen welke features een positieve inpact hebben op ons recommendation system. Bij het implementeren van een item-based collaborative filtering algoritme zal er moeten worden gekeken hoeveel reviews en gebruikers nuttig zijn om mee te nemen. Willen we bijvoorbeeld gebruikers met maar één review of gebruikers met een gemiddelde van vijf sterren eruit filteren.

Om het item-based collaborative filteren te evaluaren zal het algoritme worden gemaakt met behulp van een trainingsset en getest worden op een testset. Met de evalatiematen precision en recall zullen ten eerste verschillende similarity maten (in ieder geval cosine, Euclidian en de nieuwe **Minkowski distance**) met elkaar worden vergeleken. Ook zal er een baseline moeten worden gecreëerd die aan kan geven of de algoritmes überhaupt beter scoren dan simpel gemiddelden of willekeurige getallen pakken. De optimale treshold kan worden bepaald door het plotten van een precision-recall curve. Op deze manier zullen de meest optimale resultaten van het algoritme moeten worden getoond aan de gebruiker.