Eindverslag Yelp Recommendation System

Tom Handgraaf, Vincent Kleiman & Winston Lam
Student number: 11852615, 11884622 & 11844078
University of Amsterdam Faculty of Science
Essay: Collectieve Intelligentie, Informatiekunde
Datum: 22-05-2019

Yelp

Yelp is een online stadsgids die mensen helpt bij het vinden van geweldige plekken om te eten, winkelen, iets te gaan drinken en zich te ontspannen. En dat allemaal gebaseerd op de onderbouwde meningen van een levendige en actieve community van lokale kenners. Yelp is een leuke en gemakkelijke manier om je mening te geven over alles wat geweldig (en minder geweldig) is in jouw omgeving.

1. Bespreek in detail wat voor systeem je wil bouwen en welke informatie uit de data je hiervoor wel of niet kunt gebruiken.

Wij denken dat het handiger is om een recommender systeem te bouwen aan de hand van content-based filtering. Rekening houdend met de gegeven data hebben we drie argumenten voor onze keuze.

1. Indien we gebruik maken van een collaborative-based filtering is het mogelijk dat gebruikers vaker een restaurant aangeraden krijgen waar ze geen affiniteit mee hebben. Dit komt vaker voor met collaborative-based filtering gezien gebruikers met elkaar vergeleken worden, wat de een lekker vindt hoeft niet te gelden voor de ander ongeacht van of ze vergelijkbaar met elkaar zijn qua eetpatroon of niet.
2. Door gebruik te maken van content-based filtering zijn we niet afhankelijk van de feedback van de users. Dit is voor ons belangrijk gezien per stad niet evenveel users zijn. Als we een collaborative based filtering systeem zouden maken, zou dit recommender systeem volgens ons niet even accuraat zijn per stad. Zo heeft de ene stad veel gebruikers en de ander slechts weinig. Dit

betekent dat een recommender system, dat zou werken met collaborative-based filtering inconsistent zou zijn wat betreft accuratie. Gezien het voor de ene stad meer data heeft om mee te werken dan voor een andere stad, vanwege verschil in users per stad. Met een content-based filtering systeem zou dit probleem voorkomen kunnen worden gezien we kijken naar hoe vergelijkbaar de restaurants zijn kijkend naar de kenmerken. Hierbij zijn we niet afhankelijk van het aantal gebruikers.

3. Als je de data gaat analyseren, dan kan er worden opgemerkt dat er per business relatief veel data wordt verzameld. Zie hier een overzicht van alle data die wordt verzameld:

   Business: business_id, name (business), address, city, state, postal_code, latitude, longitude, stars, review_count, is_open, attributes (Restaurantstakeout, restaurantreservations, goodforkids, restuarantpricerange2, wifi, alcohol, businessacceptscreditcards, ambience, restauarantsdelivery, bikeparking, noiselevel, outdoorseating, caters, drivethru, restaurantstableservice, restaurantsgoodforgroups, restaurantsattire, hastv, businessparking), categories, hours.

   Doordat er zoveel data per business wordt verzameld, kan het algoritme op veel verschillende/combinerende manieren de data van verschillende businesses vergelijken en hierdoor aanbevolen opties genereren. Verder is het van belang dat het algoritme rekening houdt met de waardering en de categorie-restaurant. Als een gebruiker geen affiniteit heeft met een bepaalde categorie/cultuur eten, dan moet het algoritme hier rekening mee houden.

   Note: Om nog wel diversiteit van categorie/cultuur eten te behouden, zal het algoritme andere categorieën ook aanbevelen. Op deze manier zal niet één soort eten overheersen en zal het algoritme proberen te achterhalen met welk soort eten de gebruiker nog meer een affiniteit heeft.

   Welke features zijn nuttig?

- De data per business met de content features/attributen en de kenmerken van de business (bijv. locatie)
- De data van de users, voornamelijk de reviews per user (met bijbehorende restaurant-categorie)
- De totale rating rekening houdend met het aantal gegeven ratings

Hoe is de data gedistribueerd?

De data die wij gaan gebruiken is per Amerikaanse stad gedistribueerd. Per stad worden de businesses met elkaar vergeleken en hieruit moeten de aanbevelingen naar voren komen. Hiervoor wordt er per stad gebruikt gemaakt van 5 bestanden: business.json, checkin.json, review.json, tip.json en user.json.

2. Bespreek welke technieken je hebt gebruikt bij het maken van een prototype recommender system en waarom deze van toepassing zijn.

Bij het maken van een prototype recommender system hebben wij de volgende technieken van toepassing:

- Utility Matrix
- Randomizer
- Content-based (item features)
- Similarity Matrix
- Neighborhood

Utility Matrix
Op de hoofdpagina van de Yelp applicatie worden er tien advertenties aangeboden voor businesses van een stad. Deze stad kan op twee manieren gekozen worden: (1) een gebruiker logt in en de stad die de gebruiker heeft geregistreerd wordt meegegeven of (2) de stad wordt willekeurig gekozen.
Als een gebruiker een gekozen/willekeurige stad meegeeft, moet de hoofdpagina de zes hoogst gereviewde (met minimaal twintig reviews) businesses van die stad weergeven in de onderste 2 x 3 tabel van de hoofdpagina. Er wordt een utility matrix gemaakt van de businesses in de stad en vervolgens wordt deze aflopend gesorteerd op de ‘stars’. Hierdoor kunnen de zes hoogst gereviewde business worden weergeven op de pagina.
Daarnaast wordt de utility matrix ook gebruikt bij het content-based filteren op de business pagina. Content-based filteren wordt later besproken in het verslag.

Randomizer
Om het probleem van de ‘power law’ zoveel mogelijk te beperken, is er bij het prototype gebruik gemaakt van een randomizer. Op de hoofdpagina en de businesspagina worden zes van de tien businesses aanbevolen d.m.v. een recommendation process en de resterende businesses worden aanbevolen d.m.v. een randomizer.
In de slideshow van de hoofdpagina worden de vier willekeurig gekozen businesses van de meegegeven/willekeurige stad aanbevolen. Op de businesspagina worden de willekeurig gekozen businesses (van de stad van de

gekozen business) op de onderste vier advertenties weergeven. Hierdoor zal de gebruiker andere content aangeboden krijgen om voor variatie te zorgen in de content die de gebruiker zoekt.

Content-based (item features)
Op de businesspagina van een gekozen business worden de zes businesses aanbevolen d.m.v. een content-based flitering recommendation process. De content-based filter maakt gebruik van de categorieën die in de data van een business wordt meegegeven. Hierdoor ontstaat er een utility matrix van de businesses van de stad (van de gekozen business) en de categorieën.
Per business wordt er aangeven welke categorieën zijn meegegeven in de data. Als er bij een business een categorie hoort, wordt dit aangegeven met een 1. Als deze categorie zich niet bevindt bij de business, dan wordt dit aangegeven met een 0.
Door het maken van een utility matrix, kan de volgende stap gemaakt worden en dat is de similarity matrix.

Similarity Matrix
Vanuit een utility matrix kan er gewerkt worden naar een similarity matrix. In de similarity matrix wordt er vergeleken in hoeverre twee businesses op elkaar lijken op het gebied van overeenkomende categorieën. Deze similarity matrix wordt weer gebruikt voor de volgende functie: de neighborhood

Neighborhood
Bij de techniek neighborhood wordt op basis van de gekozen business, de top zes best overeenkomende businesses uit de similarity matrix gehaald. Op basis van de gekozen business zal de gebruiker de meest overeenkomende businesses ook aanbevolen krijgen. Dit kunnen de businesses die volgens de recommendation processes (gebaseerd op business-categorie) relevant voor de gebruiker zijn.

3. Bespreek de resultaten van je experimenten met het prototype en de kwaliteit van de geleverde aanbevelingen.

Om het prototype te testen, hebben wij in de praktijk geëxperimenteerd met data. Hierbij hebben wij een random stad meegegeven (en waar nodig een random business) en hebben wij alle stappen van deel 2 doorlopen. Elke stap hebben wij geprint en de resultaten hebben wij vergeleken met de echte data. Een voorbeeld: op de hoofdpagina moesten de zes hoogst gereviewde steden worden laten zien. In de data zijn wij nagegaan of inderdaad de hoogst gereviewde steden werden laten zien. Ditzelfde proces hebben wij toegepast op de business pagina. Daarbij werden de random gekozen businesses ook onderzocht in hoeverre deze klopten.
Door dit meerdere keren te herhalen, hebben wij dit kunnen experimenteren en vervolgens kunnen bevestigen dat dit op de juiste manier werkte. De aanbevelingen werden op de juiste manier aangegeven.

De kwaliteit van ons prototype hebben wij dus geëvalueerd en hieruit hebben wij een aantal conclusies getrokken.
Om te beginnen is ons systeem robuust. Het recommendation system is gebaseerd op content based, waardoor het onafhankelijk is van de users. De recommandations zijn gebaseerd op de relatie tussen eigenschappen van de items. Onze aanbevelingen houden geen rekening met de input met de users. Voorbeeld:

Restaurant A is 'Thais' en heeft een average rating van 4.5 met 50 ratings. Restaurant B is ook 'Thais' en heeft een average rating van 1.5 met 10 ratings. Ondanks het feit dat deze restaurants qua ratings niet gelijkwaardig zijn. Worden ze door onze similarity matrix toch vergelijkbaar gevonden door content-based filtering (omdat ze allebei 'Thais' zijn). We houden met onze huidige prototype dus nog niet rekening met user input. De kwaliteit van de restaurants volgens de users worden dus nu niet meegenomen in de recommendation.

Ook zal het lastig zijn om een aanbeveling te doen als er nieuwe users zijn die nog geen ratings hebben gegeven aan items. Dit zijn problemen die mogelijk verholpen kunnen worden met behulp van collaborative filtering.
In de huidige situatie zullen de restaurant aanbevolen worden op basis van content, dus bijvoorbeeld eigenschappen van het restaurant. Als we ook collaborative filtering toepassen kunnen we onze aanbevelingen zowel op content als op ratings baseren. De users zullen dan aanbevelingen krijgen die aansluiten op hun eigen persoonlijke interesses. We spreken dan van een hybride based filtering.
Bij collaborative filtering moet wel goed opgelet worden voor veel voorkomende problemen, zoals 'cold start problem' en first rating problem'. Het 'cold start problem' betekent dat er niet genoeg data is om een match in het systeem te vinden. Er zijn dus een hoop users nodig om een accurate aanbeveling te doen. Bij het 'first rating problem' geldt dat het systeem een item niet kan aanbevelen als deze nog nooit een rating heeft gehad. Bijvoorbeeld als er een nieuw item op de markt komt.

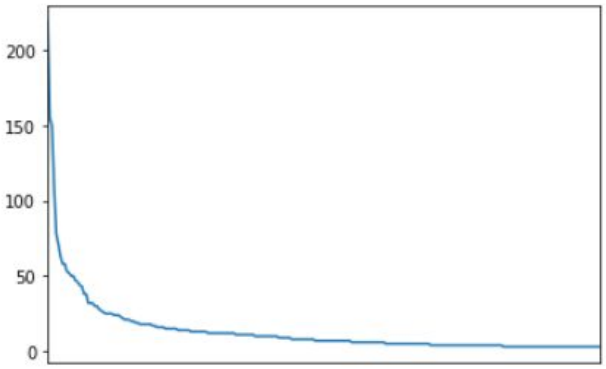
4. Geef een uitgewerkt voorstel voor een recommender system, onderbouwd met je kennis over het vakgebied en de resultaten van je experimenten uit de exploratiefase.

Het voorstel overzicht
Ons voorstel voor een recommender system doet aanbevelingen gebaseerd op de content van verschillende businesses. Elke business wordt gekenmerkt door een aantal categorieën. Aan de hand van deze categorieën hebben we gekeken hoe erg businesses op elkaar lijken, dit is gedaan aan de hand van een similarity matrix en content based filtering. Daarnaast is er gebruik gemaakt

van een randomizer en een dataframe om de businesses op beoordeling te kunnen filteren.
In deel 2&3 is behandeld hoe deze data en aanbevelingen verwerkt worden in op de pagina's.

Het voorstel inhoudelijk

Power law
De data van de steden ervaren een power law probleem. Hier volgt een plot, waarbij het aantal reviews van de businesses worden weergeven.



Om dit probleem te omzeilen, gebruikt het recommendation system een randomizer. Hierbij wordt ervoor gezorgd dat businesses die minder reviews hebben, alsnog aanbevolen kunnen worden.

Evaluatie content based filtering
Om een recommendation system in cijfers te evalueren zijn metrieken nodig die het verschil tussen de geschatte waardes en daadwerkelijke waardes berekend. Voor het testen van een specifieke geschatte rating moet de werkelijke waarde van de voorkeur van een user vergeleken worden. Echter bestaat deze voorkeur waarde niet, omdat niemand exact weet wat een user in de toekomst leuk vindt. Dus wordt de data opgesplitst in twee delen. Een training set, waarmee het model gebouwd wordt, en een test set, waar het model op getest wordt. Het recommendation system zal de missende values moeten voorspellen. Achteraf zijn deze voorspellingen vergeleken met de daadwerkelijke values uit de test set.
Wij hebben uiteindelijk zowel de MSE en RMSE over een collaborative filtering methode als over onze content based filtering methode berekend, omdat beide methodes erg accuraat zijn om de effectiviteit van het recommendation system te testen. Door de errors te kwadrateren en hier de root van te nemen van de mean zullen de grote values meer mee tellen.

Wij hebben de predicted ratings die gelijk zijn aan nul gefilterd, zodat deze niet mee worden genomen in de berekening. Het grootste deel van de

predicted ratings was gelijk aan nul, waardoor we uitkwamen op een extreem hoge MSE en RMSE. Uiteindelijk kwam er uit onze mse en rmse het volgende resultaat:

```
mse for item based collaborative filtering 2.75
mse for content based filtering: 2.00
rmse for item based collaborative filtering 1.68
rmse for content based filtering: 1.31
```

Zoals hierboven te zien is, is zowel de MSE als de RMSE voor content based filtering lager, wat betekent dat de verschillen tussen de normale ratings en de predicted ratings kleiner is bij content based filtering dan bij collaborative filtering.

Toekomstige verbeteringen

Door collaborative filtering in de toekomst nog toe te passen op ons huidige recommendation system kunnen we niet alleen onze aanbevelingen op content baseren, maar ook op ratings. Door content based filtering en collaborative filtering beide toe te passen spreken wel van een hybride based filtering.

Beide methodes vullen elkaar goed aan, maar toch moet er op beide mankementen worden gelet. Bij bijvoorbeeld collaborative filtering moet wel goed opgelet worden voor een aantal populaire voorkomende problemen, zoals ‘cold start problem’ en first rating problem’.