

Project Information and Knowledge Engineering

Eindrapport

18-01-2012

Groep 4

Ruben Visser 4081641



Olaf Maas 4086864



Marieke van der Tuin 4079299



Begeleiders

1^e docent: Marcel Reinders

2^e docent: Arjen de Vries

1^e student-assistent: Samur Cardoso de Araujo

2^e student-assistent: Raynor Vliegendhart

Inhoudsopgave

Inleiding	3
Implementatie	4
Initialisatie van de graaf	5
Via Yahoo	5
Via de gebruikerswaarderingen	6
Verbeteren van de graaf	6
Positieve waarderingen	6
Negatieve waarderingen	7
Het verwijderen van suggesties	8
Het geven van de suggesties	8
Het eerste algoritme: het bouwen van een MST	8
Het tweede algoritme: sorteren op score	9
Het derde algoritme: de combinatie van beiden	9
Het vierde algoritme: de laagste waarderingen.....	9
Het geven van een suggestie: een voorbeeld	9
Design	10
Evaluatie	12
Opzet van de test	12
Resultaten tests	13
Evaluatie test met 2 ^e algoritme	14
Evaluatie test met het volledige algoritme	14
Evaluatie van de evaluatie.....	15
Discussie	16
Verbeteringen voor de implementatie	16
Verbeteringen in het proces.....	16
Communicatie onderling	17

Inleiding

In het 2^e kwartaal hebben wij, Ruben, Olaf en Marieke, samengewerkt om een systeem te maken waarmee een gebruiker nieuwe muziek kan ontdekken en kan uitvinden welke genres hij leuk vind.

Hiervoor is een product gemaakt dat gebruikmaakt van diverse databronnen: DBPedia, MusicBrainz, Last.FM en Yahoo.

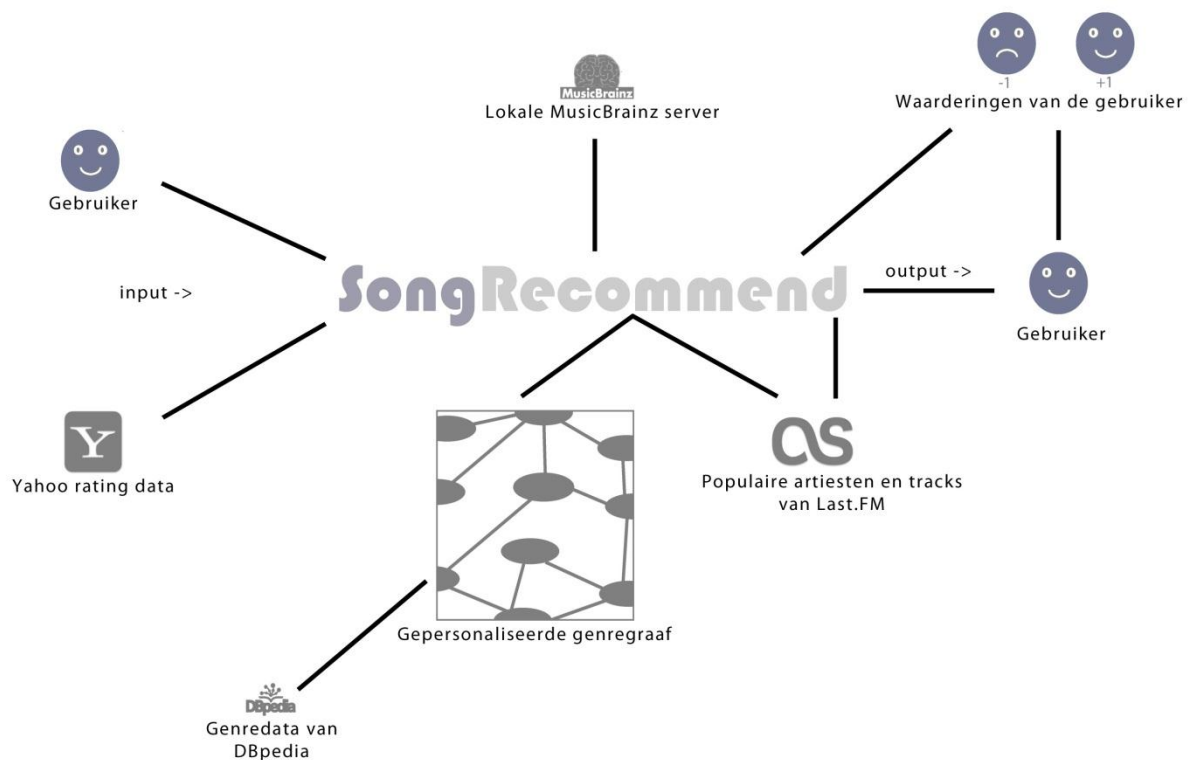
In dit eindverslag is te lezen hoe de implementatie van het systeem is gemaakt en hoe het uiteindelijke product eruit ziet. Daarnaast is het systeem ook geëvalueerd. De resultaten daarvan zijn ook opgenomen in dit eindverslag. Tenslotte valt in het laatste hoofdstuk 'Discussie' te lezen wat er nog verbeterd kan worden aan ons systeem en hoe wij het project de volgende keer aan zouden pakken.

Implementatie

Het systeem dat wij hebben gemaakt, werkt met diverse databronnen en methodes om deze te bereiken. Globaal werkt het systeem als volgt:

- De gebruiker vult zijn basis genrevoorkeuren in. Samen met de algemene Yahoo rating set vormt dit de initiële input voor het systeem.
- Vervolgens worden er favoriete genres bepaalt. Dit gebeurt aan de hand van een genregraaf, welke is opgebouwd door middel van de data op DBpedia.
- Met deze lijst favoriete genres worden populaire bijbehorende artiesten van Last.FM opgevraagd. Van deze artiesten wordt het op dit moment meest populaire nummer als track getoond.
- Ondertussen wordt er ook nog data van de MusicBrainz server opgehaald, zoals links naar pagina's behorende bij de artiest.
- De suggesties worden getoond aan de gebruiker. De gebruiker kan deze muziek beoordelen. Aan de hand hiervan wordt de genregraaf geüpdatet, waarmee vervolgens weer betere suggesties kunnen worden bepaald.

Zie ook het onderstaande diagram, dat het geheel grafisch weergeeft:



Om muziekvoorkeuren voor verschillende gebruikers te kunnen berekenen, wordt een graaf gebruikt. In deze graaf worden de relaties tussen verschillende muziekgenres weergegeven, eerst met gegevens die een gemiddelde muzieksmaak vertegenwoordigen, maar na meer gebruik van het systeem past deze graaf zich steeds meer aan, aan de gebruiker.

Initialisatie van de graaf

De graaf wordt op twee manieren geïnitieerd: via de rating data uit de Yahoo dataset en via de voorkeuren die de gebruiker zelf doorgeeft.

Via Yahoo

In de Yahoo dataset zit een klein gedeelte van de waarderingen die Yahoo gebruikers hebben gegeven aan muzieknnummers. Elke gebruiker uit de dataset heeft minstens 10 songs een waardering toegekend. De waardering bestaat uit een getal tussen de 1 en 5. Een 1 geeft hierbij een lage waardering aan, een 5 juist een hele hoge.

De songs zelf hebben willekeurige nummers gekregen, waardoor deze niet zijn te herleiden naar een daadwerkelijke artiest en nummer of MBID. Wel is er een koppeltabel toegevoegd waarin de songs zijn gekoppeld aan een bepaald genre. Deze informatie kan worden gebruikt om de genres uit de graaf te initialiseren.

Hiervoor is eerst een gedeelte van de data (30 miljoen waarderingen) uit de tekstbestanden van Yahoo via een Java programma ingelezen. Deze data is vervolgens in een database gezet. In de database worden de volgende gegevens opgeslagen:

- Genre id
- Genre naam
- Het aantal gebruikers dat per genre heeft gestemd
- De waarderingen van alle gebruikers voor een bepaald genre

De gemiddelde waardering wordt hiermee per genre bepaalt door het algoritme dat IMBD gebruikt:

$$(v \div (v+m)) \times R + (m \div (v+m)) \times C$$

Met:

R = gemiddelde van het genre, ofwel alle waarderingen gedeeld door het aantal gebruikers

v = het aantal gebruikers dat heeft gestemd

m = minimum aantal stemmen die nodig zijn (wij hebben gekozen voor 1500)

C = de gemiddelde waardering van alle genres bij elkaar

Door het bepalen van het gemiddelde op deze manier, worden genres waar maar een beperkt aantal mensen op stemmen (vaak de wat alternatievere) weggelaten. Ook krijgen de genres die juist door heel veel mensen worden beoordeeld, relatief gezien een hogere beoordeling.

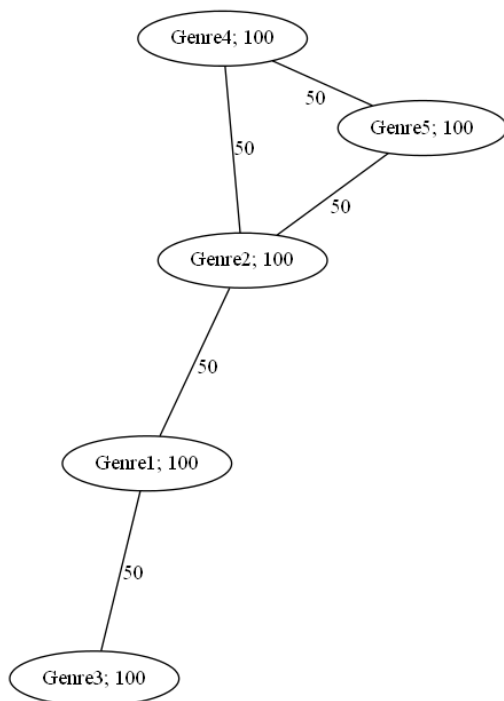
De gemiddelde waardering die hiermee wordt berekend, is uiteraard een score tussen de 1 en 5. Deze score wordt met de factor 30 gewogen en vervolgens afgerond. Van al deze gemiddelde waarderingen wordt het gemiddelde genomen. Dit wordt vervolgens van de score afgetrokken.

De wegen die daaruit komen (ongeveer variërend tussen de +15 en -30) worden opgeteld bij de nodes in de graaf, welke een standaard waarde van 100 hebben.

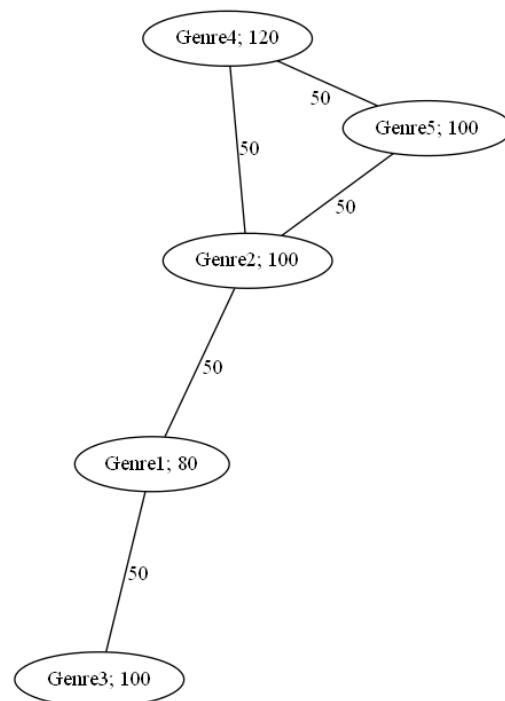
Via de gebruikerswaarderingen

Na de registratie bij het systeem, komt de gebruiker als eerste een pagina tegen waarop hij gevraagd wordt om zelf aan enkele vooraf gedefinieerde genres een score toe te kennen. Deze scores vormen naast de geïmporteerde data uit de Yahoo-dataset een zeer algemene indruk van wat een gebruiker graag zou luisteren.

De scores die een gebruiker kan geven zijn op een schaal van 1 tot 5: een 1 staat voor niet leuk, een 3 voor neutraal en een 5 voor zeer leuk. Bij het opbouwen van de graaf worden deze omgezet in een percentage tussen de 80 en 120 (80 voor een score van 1, 120 voor een score van 5). De waarde van de node waarop deze van toepassing is wordt aangepast naar dit percentage ten opzichte van de originele waarde.



Figuur 1: standaardgraaf



Figuur 2: graaf met gebruikersvoorkeuren

In dit voorbeeld is te zien hoe dat in de praktijk toegepast wordt: in figuur 1 is de situatie voor het toepassen van de scores weergegeven, in figuur 2 is er een score van 5 gegeven aan Genre 4, een score van 1 aan Genre 1 en aan de overige genres een score van 3.

Verbeteren van de graaf

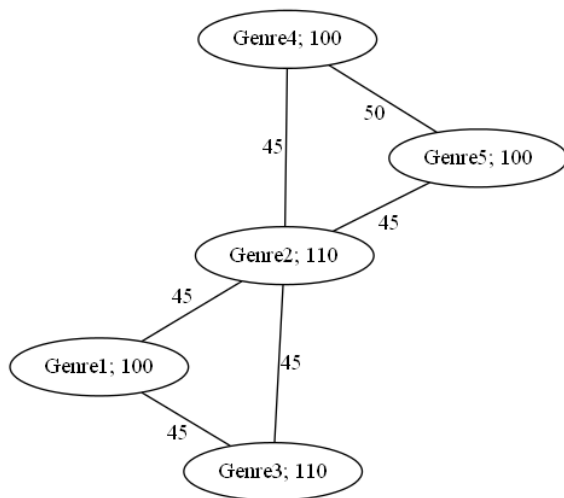
Om de gebruiker meer invloed te geven op de suggesties die door de graaf gegeven worden, wordt de graaf aangepast aan user input.

Positieve waarderingen

Bij een positieve waardering voor een suggestie, worden er 3 dingen aangepast in de graaf. Als eerste worden de scores van alle nodes die corresponderen aan de suggestie verhoogd.

Vervolgens wordt er naar de edges gekeken die aan de nodes gekoppeld zijn. Alle edges die aan een veranderde node vastzitten worden verkort.

Als laatste wordt er gekeken of er directe verbindingen tussen alle nodes die bij deze suggestie horen zijn. Als een verbinding nog niet bestaat, wordt deze aangemaakt en krijgt een score van 90% van de standaardlengte 50.



Figuur 3: voorbeeld van positieve waardering

In figuur 3 staat een tekening van de graaf, nadat er een stem is uitgebracht op een suggestie die gekoppeld was aan Genre2 en aan Genre3.

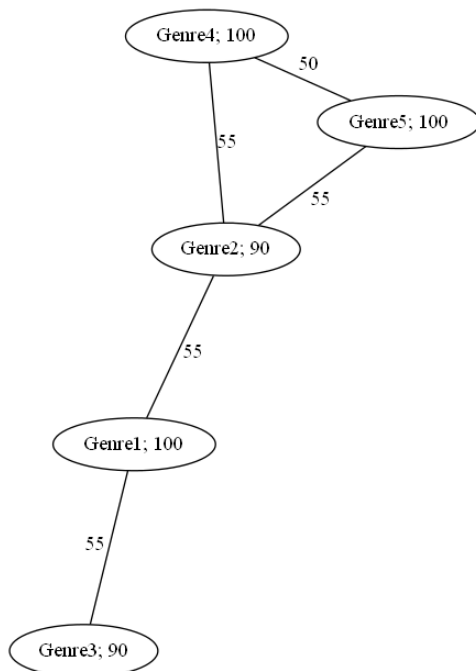
Zoals er te zien is, zijn de scores op Genre2 en Genre3 verhoogd met 10%, en is er een nieuwe edge ontstaan tussen deze 2 nodes met lengte 45.

Negatieve waarderingen

Bij een negatieve waardering voor een suggestie, worden er net als bij een positieve waardering 2 dingen aangepast.

De score op een node wordt in dit geval verlaagd, en de edges tussen de nodes die bij deze suggestie horen worden langer. Waar er echter bij een positieve waardering ook edges toegevoegd werden, gebeurt dat hier niet, maar ook worden er geen edges verwijderd.

Hiervoor is gekozen omdat het niet erg zinnig is om een negatieve relatie uit te drukken in een graaf waar gezocht wordt op positieve relaties, en ook omdat het weghalen van edges een veel grotere impact heeft op de structuur van de graaf dan het toevoegen ervan.



Figuur 4: voorbeeld van een negatieve waardering

In figuur 4 is te zien wat er gebeurt ten opzichte van de situatie in figuur 1 wanneer er een negatieve waardering wordt gegeven aan een suggestie die gekoppeld is aan de nodes Genre2 en Genre3. De nodes krijgen een lagere score en enkele edges worden langer.

Het verwijderen van suggesties

Suggesties kunnen ook worden verwijderd uit het systeem. Als een suggestie verwijderd wordt, worden er 2 acties ondernomen. Zo wordt er als eerste voor gezorgd dat deze suggestie niet meer terug komt in de zoekresultaten.

Als tweede worden er ook in dit geval acties ondernomen op de graaf. Een verwijderde suggestie wordt in de database opgeslagen als een negatieve waardering (een -2 om precies te zijn), en dat vertaalt zich in de graaf naar exact dezelfde actie als dat er een negatieve waardering gegeven zou zijn.

Het geven van de suggesties

Nu er duidelijk is hoe de graaf opgebouwd wordt en hoe gegevens daarin verwerkt worden, is het laatste punt wat de graaf kan doen het geven van genresuggesties. In de volgende stukjes worden de 3 algoritmes besproken die dit mogelijk maken. Verder wordt er ook nog een algoritme besproken dat losstaat van het geven van suggesties. De algoritmes worden besproken in de volgorde waarop ze aan het systeem zijn toegevoegd: het oudste algoritme komt eerst, gevolgd door latere.

Het eerste algoritme: het bouwen van een MST

Het algoritme om een Minimum spanning tree door de boom te maken, lijkt het meeste op een Prim's algorithm. Er wordt een startpunt gekozen, namelijk de node met de hoogste score, en van daaruit wordt er gekeken naar welke node er het 'beste' toegevoegd kan worden aan de boom. Hiervoor wordt er gekeken naar de scores van alle nodes die met minstens 1 edge aan de boom verbonden is, waar de lengte van de edge als 'straf' wordt afgetrokken. Om te kunnen vergelijken met andere algoritmen, is er besloten om deze straf te compenseren door de gemiddelde lengte van alle edges weer op te tellen bij de score.

Na x iteraties van dit algoritme, is het resultaat een lijst van x+1 nodes die voldoen aan de voorwaarden die gesteld zijn.

Het tweede algoritme: sorteren op score

Het eerste algoritme is de meest simpele: van alle nodes worden de scores opgevraagd, en de nodes worden hierop aflopend gesorteerd. De eerste x nodes uit de resulterende lijst worden teruggegeven (waarbij x een gewenst aantal resultaten is).

Het derde algoritme: de combinatie van beiden

Beide eerder besproken algoritmes hebben hun eigen nadelen: het tweede algoritme verandert het resultaat snel na kleine wijzigingen en geeft een gespreid resultaat, terwijl het eerste algoritme heel erg op elkaar lijkende suggesties geeft en wat minder snel verandert na veranderingen aan de graaf.

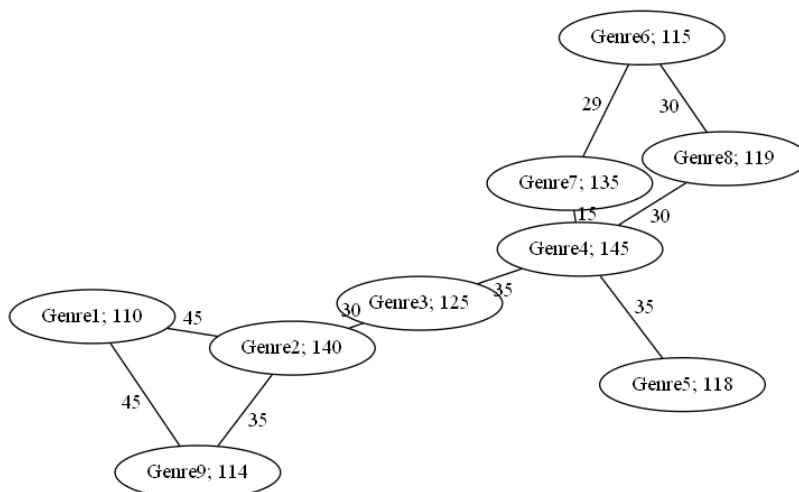
Het derde algoritme is een poging om deze nadelen een beetje uit te middelen en als voordeel in te zetten: beide algoritmes wordt gevraagd om een suggestie voor een nieuwe node, en er wordt vervolgens gekeken naar welke methode de beste suggestie opleverde.

Het vierde algoritme: de laagste waarderingen

Het vierde algoritme is een rare eend in de bijt. Dit algoritme werkt net als het tweede algoritme, maar dan andersom. Het teruggeven van de minst gewaardeerde nodes wordt gebruikt voor de statistiekenpagina en is niet van invloed op de suggesties die gemaakt worden.

Het geven van een suggestie: een voorbeeld

In de praktijk wordt het geven van de best scorende genres afgehandeld door algoritme 3. Dit komt neer op algoritme 1 waar aanpassingen aan gemaakt zijn om algoritme 2 ook toe te kunnen passen.



Figuur 5: voorbeeldgraaf

In dit voorbeeld is te zien in welke volgorde de nodes in deze graaf gesorteerd worden. De eerste stap wordt vastgelegd door het derde algoritme (wat erop neerkomt dat die de oplossing van het tweede algoritme forceert). Vervolgens laat het derde algoritme de andere 2 algoritmes een suggestie voor de volgende node doen, wat als 2^e node Genre oplevert die aangedragen is door het eerste algoritme.

In dit voorbeeld is er doorgedaan met het blijven uitvoeren van algoritme 3 totdat alle nodes gesorteerd zijn, in de praktijk zal hier een limiet op worden toegepast: van een lijst van honderden nodes zouden anders toch slechts enkelen gebruikt worden.

Tabel 1: uitkomst algoritme 3

Rang	Naam	Algoritme
1	Genre4	3 / 2
2	Genre 7	1
3	Genre 2	2
4	Genre 3	2
5	Genre 8	2
6	Genre 6	1
7	Genre 5	2
8	Genre 9	2
9	Genre 1	2

Design

Het eerste scherm waar je op terecht komt na het inloggen in het systeem, ziet er uit zoals hiernaast is weergegeven. Dit scherm krijgt de gebruiker enkel te zien indien hij nog nooit eerder heeft ingelogd.

In dit scherm kan de gebruiker zijn genrevoorkeuren aangeven. De sliders kunnen van links naar rechts worden verschoven. Als een slider helemaal links staat (gelijk aan waardering 1), betekent dit dat de gebruiker het genre niet leuk vindt. Als de slider helemaal aan de rechterkant staat (gelijk aan waardering 5), betekent dit dat de gebruiker het genre juist helemaal te gek vindt.

Nadat de gebruiker zijn voorkeuren heeft aangegeven, klikt hij op de knop verder, waarna de suggesties getoond zullen worden zoals in onderstaand scherm te zien is.

Blues

Classical:

Country:

Dance:

Electronic:

Hip-Hop

House

Jazz:

Pop

Punk

Reggae

Rock

Soul

Techno

World

SongRecommend Bewerk voorkeuren

delete

Blitzkrieg Bop
RAMONES
american, classic pop
-1 +1 i

London Calling
The Clash
british, classic pop
-1 +1 i

All The Small
blink-182
american,
-1 +1 i

Dear Maria,
All Time Low
-1 +1 i

Sex Killer
Jeffrey Lee
-1 +1 i

Let's Get Killed
The Immortal Lee
-1 +1 i

Jockey Full Of
Tom Waits
american, blues,
-1 +1 i

Tighten Up
The Black Keys
blues, blues rock,
-1 +1 i

Run To The Hills
Iron Maiden
british, english, hard
-1 +1 i

Princess Of The
Saxon
british, hard rock,
-1 +1 i

Add Recommendations

Hier zijn diverse elementen te zien. Het grootste gedeelte van het scherm wordt in beslag genomen door de suggesties. Bij elke suggestie staat bovenaan het tracknummer. Dit is altijd één van de meest populaire tracks van de bijbehorende artiest. Onder het tracknummer staat een afbeelding van het bijbehorende album. Daaronder staat de artiest vermeldt. De regel daaronder zijn genres weergegeven waarbij de suggestie hoort. Verder zijn er onderaan het suggestie blokje 3 knopjes te vinden: -1, +1 en i. Met de -1 en +1 knoppen kan een nummer gewaardeerd worden. Indien je het nummer al een keer eerder hebt beoordeeld, staat er een zwarte rand om de +1 of -1. Door op het i'tje te klikken, komt er meer informatie tevoorschijn. De volledige naam van de artiest en het nummer worden dan getoond. Ook worden er allerlei linkjes naar internetpagina's behorend bij de artiest tevoorschijn, zoals Wikipedia.



Tom Waits - Jockey Full Of Bourbon

- [wikipedia](#)
- [fanpage](#)
- [microblog](#)
- [purchase for download](#)
- [biography](#)
- [BBC Music page](#)
- [discogs](#)
- [IMDb](#)
- [myspace](#)
- [official homepage](#)

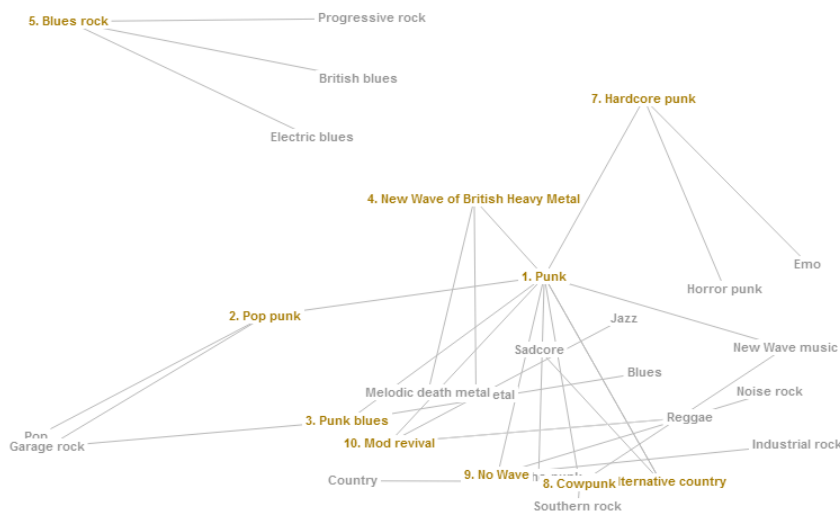


Verder kan een nummer uit de suggesties verwijderd worden door het te slepen naar de kolom aan de linkerkant waar 'delete' bij staat. Uiteraard kunnen er ook suggesties worden toegevoegd. Dit kan door op de knop Add recommendations te klikken. Vervolgens kan er een keuze worden gemaakt tussen een suggestie die je wellicht leuk zou vinden, maar wel van een ander genre is, of een suggestie welke veel wegheeft van de muziek die je leuk vind.

Ook is het mogelijk om het basisprofiel van genrevoorkeuren aan te passen door op de knop Bewerk Voorkeuren te klikken. Hiermee kom je weer uit op het scherm zoals beschreven is aan het begin van dit hoofdstuk.

In het systeem is ook een statistiekenpagina beschikbaar. Op deze pagina valt een top 10 van de specifieke genres die je leuk vind, bepaalt aan de hand van de input die je zelf hebt gegeven bij het waarderen van de muziek. Ook is er een flop 10 te zien, met daarin de genres die je absoluut niet leuk vind.

Aan de hand van de top 10 is een grafische weergave gemaakt. Hierin vallen de 10 favoriete genres te zien en hun connecties. Ook worden er genres weergegeven welke sterk gerelateerd zijn aan jouw top 10. Alle genres zijn te verslepen binnen het scherm om zelf een goed overzicht te kunnen krijgen van de opgebouwde graaf. De statistieken pagina is hieronder weergegeven:



Links ziet u uw muzieksmaak, de top 10 genres die u volgens onze gegevens het meest bevallen zijn bruin gekleurd. De andere genres zijn lichter gekleurd en zijn gerelateerd aan uw top 10, deze genres kunt u ook leuk vinden. Hieronder ziet u uw top 10 en flop 10.

Top 10

1. Punk
2. Pop punk
3. Punk blues
4. New Wave of British Heavy Metal
5. Blues rock
6. Alternative country
7. Hardcore punk
8. Cowpunk
9. No Wave
10. Mod revival

Flop 10

1. Electronic
2. World
3. Rock
4. Rock and roll
5. Ambient music
6. Alternative rock
7. Funk
8. Indie rock
9. Hard rock
10. Speed metal

Evaluatie

Om te weten te komen of het ontstane systeem daadwerkelijk goede voorspellingen kan geven aan de gebruikers, is het nodig om het muziekrecommandatiesysteem te testen.

Daarnaast wilden wij graag weten in hoeverre de 2 gebruikte algoritmes invloed hebben op de graaf. Hiervoor hebben wij eerst de test uitgevoerd door enkel gebruik te maken van het 2^e algoritme (sorteren op score). Vervolgens hebben we dezelfde test uitgevoerd met gebruikmaking van het 3^e algoritme, welke het 1^e en 2^e algoritme combineert.

Opzet van de test

Bij de test wordt er gekeken of de voorkeuren van een bepaald persoon naar voren komen na het invoeren van zijn voorkeuren in het systeem. Aangezien het idee is dat ons systeem steeds betere suggesties geeft naarmate er meer waarderingen zijn gegeven aan de tracks, zullen er meerdere malen waarderingen van de gebruiker ingevuld moeten worden. Uiteraard zal ook het basisprofiel van de testgebruiker ingevoerd worden.

Aangezien wij zelf aan de hand van de genres de artiesten selecteren, en aan de hand van deze artiesten hun beste track selecteren, ligt het voor de hand om te werken met de lijst van favoriete artiesten van de testgebruiker. Via Last.FM is dit gemakkelijk haalbaar door middel van 'user.getTopArtists'. Daarom hebben we ervoor gekozen om testgebruikers van Last.FM te halen.

Dit resulteert in de volgende concrete opzet van de test:

1. Er wordt een willekeurige gebruiker van Last.FM gekozen, die minstens 3000 artiesten heeft beoordeeld.
2. De favoriete genres van deze gebruiker worden via user.getTopTags opgehaald. Deze worden ingevoerd bij de basis genrevoorkeuren van een nieuwe testgebruiker in ons systeem.
3. Aan de hand hiervan worden er een aantal suggesties gegeven door ons systeem. De artiesten behorend bij de suggesties worden opgezocht in de top 1000 van favoriete artiesten van de testgebruiker via user.getTopArtists. Vervolgens wordt geregistreerd welke suggesties wel en niet voorkomen in de top 1000 van de testgebruiker. Aan de hand hiervan wordt een percentage juiste suggesties bepaald. Ook werd de positie van de artiest uit de top 1000 van de gebruiker genoteerd, en hiervan een gemiddelde bepaald.
4. De juiste suggesties krijgen een +1 waardering, de rest een -1 waardering.
5. Vervolgens worden stap 3 en 4 vier maal herhaald. Door het geven van de waarderingen zal de lijst suggesties steeds veranderen.
6. Om nog extra variatie aan te brengen, zullen er nu 15 suggesties toegevoegd worden door middel van de 'iets nieuws toevoegen' optie. Ook hiervan wordt gekeken in hoeverre er suggesties wel of niet voorkomen in de top 1000 artiesten van de gebruiker, en worden er +1 en -1 waarderingen toegekend.
7. Vervolgens wordt nog een laatste maal stap 3 en 4 uitgevoerd. De resultaten hiervan zijn de eindresultaten van de test.

Resultaten tests

De willekeurige gekozen gebruikers zijn ikea, irviniebrown en lucasm2.

“ikea” houdt vooral van Rock. Daarnaast vind “ikea” het Techno, Pop, Punk en Reggae genre ook wel aardig. Over de andere genres is geen mening bekend.

Gebruiker “irvinebrown” houdt van Rock, Soul en Jazz. Daarnaast is er voor bijna alle andere genres bekend hoe leuk hij deze vind.

De derde gebruiker, “lucasm2”, is een typische Amerikaan. Hij houdt van Rock, Punk en Techno. Daarnaast vind hij Reggae en Electronic ook erg leuk.

Deze 3 gebruikers kregen de volgende resultaten tijdens de twee tests:

Ikea

	2 ^e algoritme			1 ^e + 2 ^e algoritme		
Actie	Juist	Percentueel	Gemiddelde	Juist	Percentueel	Gemiddelde
1: Suggesties beoordelen	4	4/19 = 21%	759.75	4	4/19=21%	545
2: Suggesties beoordelen	6	6/24 = 25%	642.17	9	9/22=41%	457.33
3: Suggesties beoordelen	8	8/19 = 42%	602.25	12	12/24=50%	405.92
4: Suggesties beoordelen	9	9/21 = 43%	624.44	12	12/22=55%	405.92
5: 15x iets nieuws	4	4/15 = 27%	392.5	2	2/15=13.33%	323
6: Suggesties beoordelen	13	13/24=54%	553.08	15	15/24=63%	410.87

Irvinebrown

	2 ^e algoritme			1 ^e + 2 ^e algoritme		
Actie	Juist	Percentueel	Gemiddelde	Juist	Percentueel	Gemiddelde
1: Suggesties beoordelen	5	5/19 = 26%	305.2	5	5/19=26%	177.4
2: Suggesties beoordelen	6	6/19 = 32%	275.83	6	6/19=32%	169.33
3: Suggesties beoordelen	9	9/24 = 38%	300.44	9	9/19=47%	277.67
4: Suggesties beoordelen	10	10/24=42%	305	9	9/24=38%	229.44
5: 15x iets nieuws	0	0/15 = 0%	-	3	3/15=20%	703.33
6: Suggesties beoordelen	12	12/24= 50%	308.9	13	13/26=50%	296.54

Lucasm2

	2 ^e algoritme			1 ^e + 2 ^e algoritme		
Actie	Juist	Percentueel	Gemiddelde	Juist	Percentueel	Gemiddelde
1: Suggesties beoordelen	13	13/19 =68%	265	7	7/19=37%	224
2: Suggesties beoordelen	17	17/21 =81%	264.42	10	10/22=45%	270.5
3: Suggesties beoordelen	21	21/24= 88%	279.05	15	15/22=68%	262.6
4: Suggesties beoordelen	21	21/25= 84%	279.05	16	16/23=70%	248.88
5: 15x iets nieuws	4	4/15 = 27%	335.25	6	6/15=40%	409.33
6: Suggesties beoordelen	26	26/29= 90%	315	23	23/28=82%	287.65

Evaluatie test met 2^e algoritme

Bij het 2^e algoritme wordt er enkel gekeken naar de waarden van de nodes (welke genres representeren), waarna de hoogste genres worden teruggegeven als resultaat. Hiermee worden vervolgens de suggesties gegenereerd.

Door enkel gebruik te maken van deze manier, zullen de connecties tussen de nodes niet gebruikt worden, waardoor de graaf niet optimaal wordt benut. Het gevolg hiervan is, is dat de voorkeuren van de gebruiker niet optimaal worden meegenomen in het bepalen van de suggesties. Daardoor worden er over het algemeen niet heel veel nieuwe suggesties gegeven van gerelateerde genres. Bij de test met "lucasm2" kwam er in shift 4 bijvoorbeeld slechts 1 (foutieve) suggestie bij, de rest bleef ongewijzigd.

Het lijkt erop dat het systeem op deze manier vooral perfect is voor de typische Amerikaan ("lucasm2"). Aangezien de voorkeuren van de gemiddelde Amerikaan vooraf worden ingeladen aan de hand van de Yahoo data, is de keuze van lucasm2 al snel goed zichtbaar in de suggesties.

Verder is te zien dat er aan het einde respectievelijk 50, 54 en 90% van de suggesties goed zijn. Dit zijn naar ons idee op zich al prima percentages voor het suggestiesysteem.

Evaluatie test met het volledige algoritme

In de 2^e test worden de beide algoritmes geïmplementeerd, ofwel er wordt zowel gekeken naar het Minimum Spanning Tree algoritme, welke kijkt naar de lengtes van de verbindingen tussen de genres, als naar de hoogste waarden van de genres zelf.

Hierdoor is er meer variatie zichtbaar bij de suggesties. Ook worden er meer nieuwe suggesties aangedragen. Een nadeel hiervan is wel dat het vaak langer duurt voordat het 'genre-profiel' van een gebruiker goed zichtbaar wordt in de suggesties. Uiteindelijk is het geleverde profiel echter wel vollediger dan door enkel gebruik te maken van het 2^e algoritme.

Bij gebruiker "ikea" leidt dit tot een betere inschatting van het aantal goede genres, waarschijnlijk omdat er voor deze gebruiker vooraf niet een duidelijk basisprofiel kon worden ingevoerd. Hierdoor bleven de resultaten eerder rond hetzelfde genre hangen, terwijl met het volledige systeem het resultaat veel meer divers was.

Bij de gebruikers "Irvinebrown" en "Lucasm2" leidt het volledige systeem echter niet tot een verbetering, zelfs een verslechtering bij "Lucasm2". Bij beiden stijgt het aantal goede suggesties echter nog steeds. Dit zou erop duiden dat, naarmate er vaker gebruik wordt gemaakt van het systeem en er vaker waarderingen worden gegeven aan de muziek, er uiteindelijk een breder en beter genre-profiel zal ontstaan met meer juiste suggesties.

Verder valt er te zien dat in de 2^e versie van de test, er overal lagere gemiddeldes bij de eindscores worden gehaald. Dit betekent dat de suggesties over het algemeen beter passen bij de gebruiker dan de eerder aangedragen suggesties met de 1^e test. Ook dit is een verbetering van het systeem.

Uiteindelijk heeft de test 50%, 63% en 82% juiste suggesties aangedragen. Dit zijn naar ons idee prima percentages voor het systeem dat wij hebben gemaakt.

Evaluatie van de evaluatie

Tijdens deze evaluatie zijn er slechts de gegevens van 3 willekeurige gebruikers gebruikt. Hoewel dit een redelijk inzicht geeft in hoe ons systeem presteert, kunnen we niet met absolute zekerheid zeggen dat het altijd zal werken, aangezien de testgroep nogal klein is. Het testen viel echter niet heel gemakkelijk te automatiseren, waardoor de tests handmatig zijn uitgevoerd. Dit is een redelijk tijdsintensieve manier om te testen, waardoor ervoor is gekozen om slechts te testen met 3 gebruikers en de verdere tijd te steken in verbetering van ons systeem.

Verder is het 1^e algoritme (de MST variant) niet apart getest. Voor een volledig beeld van de werking van de 2 verschillende algoritmes zou dit nog handig zijn om toe te voegen aan de evaluatie.

Discussie

Hoewel wij uiteindelijk een product hebben neergezet waar we erg blij mee zijn, zijn er altijd nog verbeterpunten mogelijk aan het product zelf, maar ook aan het gehele ontwikkelproces. Deze punten worden in dit hoofdstuk behandeld.

Verbeteringen voor de implementatie

In het opgeleverde systeem is de graaf met genres 1 van de belangrijkere onderdelen die het mogelijk maakt om suggesties te geven. Er komen suggesties uit die zeker niet verkeerd zijn, maar verbetering is uiteraard mogelijk.

Een van de punten die verbeterd kan worden, is wat er gebeurt met edges als er een stem uitgebracht wordt. Op dit moment worden de waardes van alle aangrenzende edges veranderd indien er op een nummer met een bepaald genre wordt gestemd. Hierdoor kunnen er tientallen wijzigingen ontstaan als het genre met veel andere genres is verbonden, zeker als de gewaardeerde song tot meerdere genres behoort. Er zou onderzocht kunnen worden of er een meer gedetailleerd beeld over de voorkeuren van de gebruiker zou ontstaan, als enkel de edges tussen de genres behorend tot hetzelfde muzieknnummer verhoogd worden.

Een ander punt is dat er behoorlijk wat 'magic constants' in de code staan. Veel van deze getallen zijn wegingsfactoren die veel invloed uitoefenen op de suggesties die gegeven worden. Ondanks dat deze getallen hun werk blijkbaar aardig doen, en ook wel uit te leggen is dat ze dat doen, ontbreekt het bij deze getallen vaak aan harde onderbouwing waarom deze zo zijn.

Verbeteringen in het proces

In het begin van het project was het erg onduidelijk wat nu precies de bedoeling was van het project. We werden erg vrij gelaten in wat we gingen doen, wat op zich erg fijn was, maar eigenlijk net iets te vrij. Het was redelijk onduidelijk wat er elke 2 weken precies opgeleverd diende te worden. Zeker de inhoud van het rapport was niet heel erg duidelijk.

Voor het project is er gewerkt in een scrum achtige opzet waarbij elke 2 weken een werkend product moest worden opgeleverd. Dit zorgde ervoor dat er vooral in het begin vrij weinig tijd beschikbaar was om goed uit te denken hoe het systeem uiteindelijk zou gaan werken. Hierdoor zijn implementaties niet altijd even goed doordacht van te voren. Zo is er lang gewerkt aan het maken van de graaf, zonder dat er daadwerkelijk was bedacht hoe deze graaf ons gaat helpen met het voorspellen van suggesties. Dit kwam pas in een veel later moment aan de orde. De korte fases zorgden er ook voor dat het product wat we konden tonen tijdens de 2-wekelijkse meetings, pas in de 3^e fase echt begon te lijken op hetgeen wat wij voor ogen hadden. Daarvoor was er weliswaar op de achtergrond wel veel gebeurd, maar dat kon nog niet getoond worden.

Als we dit project opnieuw zouden moeten doen, dan zou het wellicht handig zijn om te werken met iets langere fases, zodat er iets meer tijd is om goed na te denken over bepaalde keuzes. Hierdoor zou het uiteindelijke systeem waarschijnlijk beter in elkaar zitten.

Verder was de aanwezige begeleiding ook niet optimaal naar ons idee. De wekelijkse meetings met de student-assistent hebben bijvoorbeeld eigenlijk nooit plaatsgevonden. Sowieso gaf de student-

assistent naar ons idee te weinig input om mee aan de slag te kunnen. Deze input was er gelukkig wel van de docenten, wat als erg handig is ervaren.

Communicatie onderling

De communicatie in ons groepje verliep erg goed. Het was erg simpel om met een zo kleine groep dingen af te kunnen spreken en even snel te overleggen.

We hebben voor het eerst met dit project met GIT gewerkt. Dit was zeker in het begin lastig opstarten, aangezien het toch net iets anders is dan SVN. Het is regelmatig voorgekomen dat iemand bijvoorbeeld een bepaalde wijziging enkel had gecommit en vergeten was deze daarnaast ook nog eens via een Push op de server te zetten. Echte voordelen van GIT boven SVN hebben we niet echt kunnen ontdekken.