Ontwerpen van een aanbevelingssysteem voor films

Gabriëla de la Cruz, Mick van Hulst

en Yoeri van Bruchem

2017

1. Inhoud

2. Inleiding

Het kijken van films gebeurd tegenwoordig veelal online. Mensen hoeven niet meer naar de bioscoop, maar maken gebruik van services als Netflix en Amazon Prime. Deze services maken gebruik van beoordelingen van hun gebruikers om aanbevelingen te genereren. De vraag van de projectgroep is op basis van welke factoren aanbevelingen worden gegenereerd en hoe kan er gebruik worden gemaakt van deze factoren om een aanbevelingssysteem te creëren.

Dit rapport behandelt de doorloop van de projectgroep bij het maken van een aanbevelingssysteem. De groep gebruikt theorie van zowel interne als externe bronnen van de Hogeschool Rotterdam om het systeem te realiseren. Er bestaat veel diversiteit in de projectgroep waardoor er onderling veel van elkaar kon worden geleerd.

2.1. Hoofdstukindeling

Het document is opgedeeld in zeven hoofdstukken. Hoofdstuk drie beschrijft de definitie van het project. Hoofdstuk vier beschrijft de technieken welke tijdens het project zijn geütiliseerd. Hoofdstuk vijf beschrijft de resultaten van het project. Hoofdstuk zes beschrijft de verantwoording van de resultaten van het project (op zowel groeps- als individueel niveau). Hoofdstuk zeven bestaat uit de bijlagen.

3. Projectdefiniëring

Dit hoofdstuk beschrijft het uitgangspunt en de doelstellingen van het project. Het gaat in op het doel en de aanpak van het project. Dit hoofdstuk geeft daarnaast een uitwerking van de eisen die zijn gesteld aan het eindproduct.

3.1. Aanleiding Project

De Minor Data Science bestaat naast de regulieren lessen ook uit een project. Dit project, uit te voeren door een projectgroep tussen de drie en de vijf personen, gaat in op de theorie die in deze reguliere lessen zijn behandeld.

De projectgroep heeft ervoor gekozen het onderwerp ‘film’ als uitgangspunt voor dit project te kiezen. De keuze is gebaseerd op het feit dat er hiervoor veel achtergrondinformatie te vinden is en omdat de groep zelf ook uit filmliefhebbers bestaat. Informatie voor dit soort projecten is ruimschoots beschikbaar op sites as Kaggle, IMDB en MovieLens.

Na de initiatie fase van het project trok de aandacht van de projectleden naar aanbevelingstechnieken voor films. In het verleden zijn er (onder andere door Netflix) verschillende wedstrijden georganiseerd voor het aanbevelen van films. Het idee is om een soortgelijk systeem te ontwikkelen.

3.2. Projectgroep

De projectgroep bestaat in totaal uit vijf personen: Mick van Hulst, Gabriëla de la Cruz, Bilal Aarabe en Joep Hoogsteden. Mick, Yoeri en Bilal studeren Business IT & Management aan de HAN in Anrhem, Gabriëla studeert Wiskunde aan de Haagse Hogeschool en Joep studeer Logistiek aan de Hogeschool van Rotterdam.

3.3. Doel

Het doel van het project is een web-applicatie realiseren welke gebruikers kunnen gebruiken om films aanbevolen te krijgen op basis van de historie van de gebruikers en kenmerken van films.

3.4. Aanpak

De tijdsduur van het project is 20 weken. In deze weken wordt in ieder geval elke vrijdag gewerkt aan het project. Naast deze vaste dagen wordt van alle projectleden verwacht in eigen tijd aan het project te werken. Het volgende stuk gaat in op de opbouw en planning van het project.

Planning

Week 1 tot 4

Deze weken dienen vooral als initiatie fase van het project. In deze fase wordt het project gespecificeerd, de scope vastgesteld en bronnen verzamelen die aansluiten bij het gekozen onderwerp. In deze weken wordt daarnaast ook gekeken naar welke methoden en technieken voor dit project het meest geschikt zijn.

Week 4 tot 8

In deze weken onderzoekt de projectgroep de beschikbare data. Dit gaat vooral in op de bruikbaarheid en kwaliteit van deze data. Ook wordt de basis van de infrastructuur opgezet en wordt de eerste data verwerkt.

Week 8 tot 12

De projectgroep onderzoekt deze weken welke algoritmes het beste aansluiten bij de uiteindelijke implementatie van het systeem. De eerste algoritmes worden ontwikkeld en getest. De infrastructuur voor de front-end wordt opgebouwd.

Week 12 tot 16

In deze weken werkt de projectgroep de algoritmes verder uit. De algoritmes worden getest op bruikbaarheid en met elkaar vergeleken. De front-end wordt in deze weken verder uitgewerkt.

Week 16 tot en met 20

De laatste weken van het project staan voor het implementeren en afronden van het systeem. De algoritmes worden samengevoegd en geïmplementeerd in de web-applicatie. De projectgroep maakt daarnaast het eindproduct klaar voor oplevering en bereidt te presentatie voor.

3.5. Eisen en wensen

De applicatie wordt ontwikkeld aan de hand van de volgende eisen:

Hoogste prioriteit

- Gebruiker moet na inloggen, films beoordelen totdat de projectgroep genoeg informatie heeft verzameld om aanbevelingen te maken.
- De applicatie moet films aanbevelen op basis van beoordelingsgedrag van een gebruiker en de kenmerken van films.
- De applicatie moet films aanbevelen op basis van gelijkenissen in beoordeling gedrag van gebruikers.

Middelmatige prioriteit

- Gebruiker moet registreren via de web-applicatie.
- Gebruiker moet inloggen via de web-applicatie.

Lage prioriteit

- De applicatie maakt gebruik van een herhalend proces waardoor gebruikers aanbevelingen kunnen beoordelen en de applicatie nieuwe aanbevelingen genereerd.
- De applicatie biedt informatie over films zodra de gebruiker hierop klikt (e.g. visualisaties).

4. Technieken

Dit hoofdstuk beschrijft de technieken en methoden welke tijdens het project zijn gebruikt en waarvoor.

4.1. Verwerken en analyseren van data

Programmeertaal R

R is een programmeertaal welke veelal voor statistische doeleinden wordt gebruikt. Binnen R is het mogelijk om verschillende pakketten toe te utiliseren, welke het mogelijk maken om functionaliteiten toe te passen om data te analyseren.

Programmeertaal Python

Python is een programmeertaal welke veelzijdig toegepast kan worden. Door gebruik te maken van verschillende pakketten kunnen er onder andere spellen en websites mee worden ontwikkeld. Er kan ook data-analyse mee worden uitgevoerd.

4.2. Beheren van data

MySQL

MySQL is een managementsysteem die wordt gebruikt om databases op te zetten. Het is een relationeel systeem wat betekent dat het via een relationeel model is opgebouwd.

4.3. Ontwikkelen Front-End

Opmaaktaal HTML

HTML is een opmaaktaal welke wordt gebruikt om structuur te creëren voor onder andere websites.

Scripttaal PHP

PHP is een scripttaal welke is bedoeld om websites dynamisch te maken. Het is een web-taal welke hoofdzakelijk via de server wordt aangeroepen.

Stylingtaal CSS

CSS is een styling taal welke wordt gebruikt om websites vorm te geven (e.g. kleur & grootte van objecten binnen webpagina's).

5. Resultaten

Het hoofdstuk resultaten is opgedeeld in twee gedeeltes. Het eerste gedeelte beschrijft de technische resultaten. Hier worden de resultaten toegelicht welke in de applicatie worden gebruikt, maar niet direct terug te zien zijn. Het tweede gedeelte beschrijft het functionele resultaat, wat bestaat uit de functionele kant van het project (de applicatie).

5.1. Technische Resultaten

Deze paragraaf beschrijft de technische resultaten van het project. Hier wordt elk resultaat welke in de applicatie wordt gebruikt, maar niet direct is terug te zien, besproken.

Infrastructuur

Om de informatie op te slaan en toegankelijk te maken voor de gehele projectgroep, is er een Raspberry Pi gebruikt om een Ubuntu server op te zetten. Op deze server is MySql geïnstalleerd. Omdat deze server toegankelijk was vanaf het internet, had dit als resultaat dat elk projectlid toegang had tot dezelfde data. Hier is voor gekozen om te voorkomen dat verschillende projectleden, verschillende varianten van de datasets gebruikten.

Voor de demo is er gebruik gemaakt van een lokale installatie van MySql, omdat de projectgroep het IP van de server niet vrij wilde geven (deze moet namelijk geregistreerd worden in de scripts en het IP is een IP-adres van een van de projectleden).

Voorbereiding data

De data is afkomstig van de MovieLens database[1]. Deze dataset biedt informatie over films en beoordelingen van users over films.

De dataset bevatte geen algemene informatie over de films (e.g. plots). Om deze reden is er gebruik gemaakt van een API welke de projectgroep in staat stelde om de plots op te slaan in de database. Deze plots zijn gebruikt om steekwoorden te vinden voor elke film. Dit wilde de projectgroep later gebruiken in het ‘content-based’ algoritme.

Ook had de projectgroep geen afbeeldingen van films. Deze waren nodig om de gebruiker een beeld te geven van films. Hiervoor is ook een API gebruikt welke het mogelijk maakte om de afbeeldingen op te halen en op te slaan in de server.

[1] https://movielens.org

Aanbeveling algoritme: Collaborative user-filtering

Dit algoritme is toegepast om gelijkenissen te zoeken tussen gebruikers. Hier wordt bijvoorbeeld bepaald dat gebruiker A film één, twee en drie beoordeeld als goede films. Gebruiker B heeft film één en drie als goede films beoordeeld. In dit geval lijkt gebruiker B op gebruiker A, omdat twee van de drie beoordelingen hetzelfde zijn. Dit zal als resultaat hebben dat gebruiker B, film twee als aanbeveling krijgt.

Het algoritme is in eerste instantie in R geïmplementeerd. Later in het project is er besloten om het algoritme alsnog in Python te implementeren om het algoritme te versnellen. (Het R-script bevindt zich in Bijlage 7.3)

Er is een soortgelijk algoritme toegepast welke was gebaseerd op een pakket van Python. Deze maakte gebruik van verschillende technieken om gelijkenissen tussen gebruikers te meten. Helaas kon de projectgroep het pakket niet toepassen op de structuur van de database. Het pakket maakte namelijk gebruik van een speciaal gestructureerde dataset. (Het Python script bevindt zich in Bijlage 7.4)

Content-based filtering

Content-based filtering wordt gebruikt om aanbevelingen te genereren aan de hand van eerder beoordeelde films van een bepaalde gebruiker. Content-based filtering kijkt aan de hand van deze films welke genres het beste bij een bepaalde gebruiker passen en voegt hier een weging aan toe. Vervolgens vergelijkt dit algoritme deze voorkeur-genres met alle andere films in de dataset. Films die in totaal het beste voldoen aan deze eis en dus het meeste wegen worden door het algoritme teruggegeven als aanbeveling.

Aanbeveling algoritme: K-Nearest Neighbours

K-Nearest Neighbours (oftewel KNN) wordt gebruikt om de afstand tussen vectoren te meten. In dit geval werd het gebruikt om de afstand tussen gebruikers te meten. Een voorbeeld kan zijn dat de afstanden tussen de gebruiker welke ingelogd is en de resterende gebruikers. De gebruikers welke dan het meest overeenkomen met de ingelogde gebruiker zijn dan ‘buren’. Door films op te halen van deze buren welke de ingelogde gebruiker nog niet heeft gezien, worden er aanbevelingen gevonden.

Aanbeveling algoritme: K-means (niet toegepast in applicatie)

K-means is een algoritme welke wordt gebruikt voor het clusteren van data. Hierin geeft de gebruiker het aantal clusters mee als parameter. Hier wordt voor elk datapunt, iteratief bepaald tot welke cluster deze behoord. Zodra deze punten zijn toegewezen worden de clusters opnieuw berekend door het gemiddelde te nemen. Op deze manier veranderden de coördinaten van clusters na elke iteratie. Het proces

stopt zodra er na een iteratie geen verandering plaatsvindt in de coördinaten van de clusters.

Het algoritme is in R geïmplementeerd en omdat de projectgroep geen inzicht had in het aantal clusters is ervoor gekozen om dit algoritme niet toe te passen. (Het R-script bevindt zich in Bijlage 7.2). Na advies van een van de leraren is ervoor gekozen om DB-Scan te realiseren.

Aanbeveling algoritme: DB-Scan (niet toegepast in applicatie)

DB-Scan is een ander soort clusteralgoritme dat gebruik maakt van andere parameters dan k-means. Hierbij geeft de gebruiker aan hoeveel data punten nodig zijn om een cluster te maken en de afstand tussen de data punten. DB-Scan zou op dezelfde wijze worden gebruikt als het k-means algoritme. Gedurende de lessen kwam de projectgroep erachter dat DB-Scan erg traag was zodra er veel informatie verwerkt moest worden. De projectgroep vond het geen goed idee om de gebruiker een lange tijd te laten wachten voor zijn of haar aanbevelingen. Om deze reden is het algoritme niet toegepast in de applicatie. (Het Python script bevindt zich in Bijlage 7.5)

Applicatie

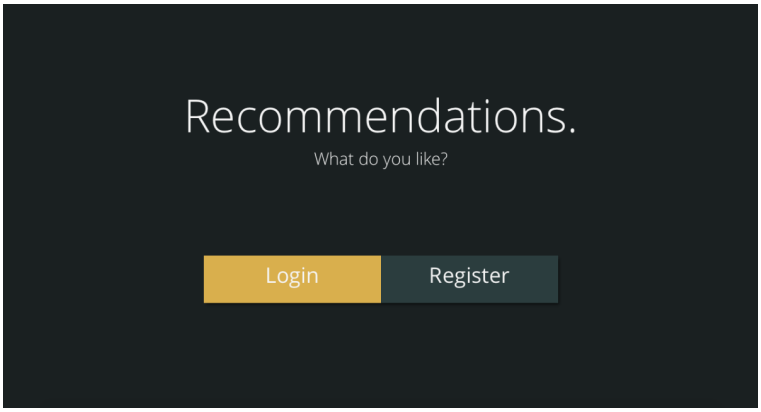
De applicatie is het eindproduct waar de gebruikt uiteindelijk mee gaat werken. Hoe deze werkt en hoe deze eruit ziet is beschreven in hoofdstuk 5.2. Door gebruikt te maken van PHP kan de applicatie gekoppeld worden met Python. Op deze manier heeft de gebruiker interactie met het systeem en is het systeem een gebruiksvriendelijke weergaven van de resultaten van de algoritmes.

5.2. Functionele Resultaten

Het functionele resultaat is een werkende web-applicatie die gebruikers films aanbevelen op basis van hun filmvoorkeuren. In dit hoofdstuk wordt het programma weergegeven door middel van schermafbeeldingen. Elke schermafbeelding wordt toegelicht met daarbij de mogelijke acties voor de gebruiker.

Hoofdscherm

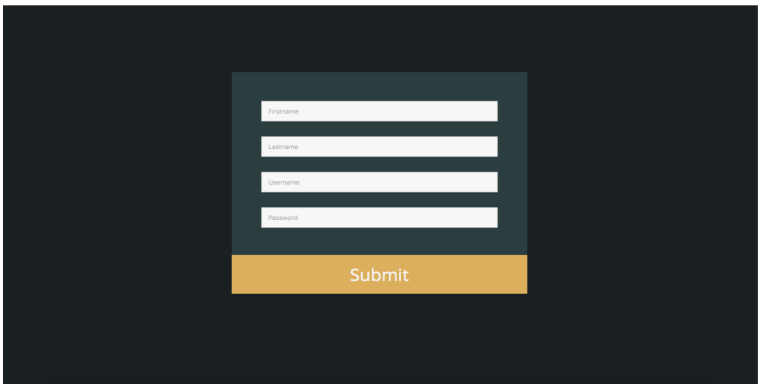
In de onderstaande afbeelding is het hoofdscherm weergegeven. Hierbij kan de gebruiker ervoor kiezen om in te loggen (bij gebruik van een bestaand account) door op de knop “Login” te klikken. Anders kan de nieuwe gebruiker zich registeren door op het knopje “Register” in te klikken.



Register-scherm

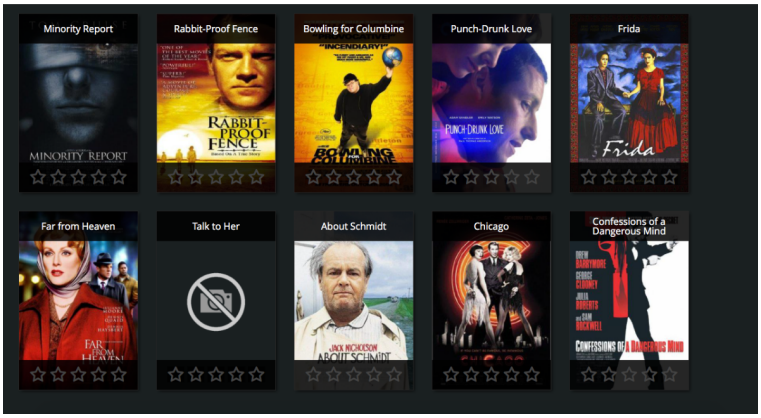
In de onderstaande afbeelding is het “register”-scherm weergegeven. De gebruiker komt in dit scherm terecht als hij/zij geen account heeft in de web-applicatie. De vereiste gegevens voor het maken van een account zijn “Firstname”, “Lastname”, “Username” en “Password”.

De gebruiker kan met het gemaakte account inloggen door middel van de “Username” en “Password”. Bij het klikken op “Submit” wordt een account aangemaakt en opgeslagen. De gebruiker keert terug naar het hoofdscherm.



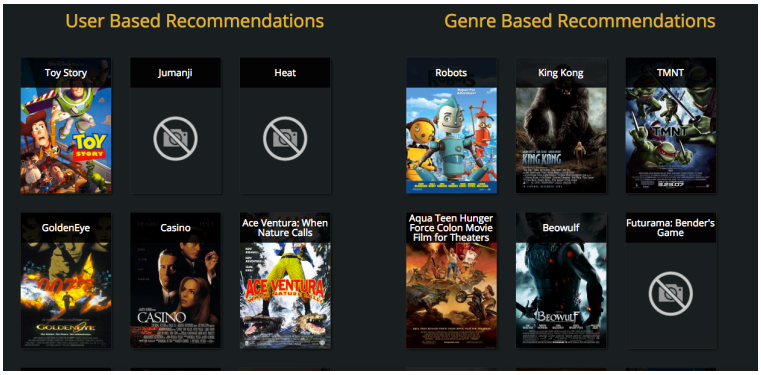
Rating-scherm

Na het inloggen, vraagt het programma aan de gebruiker om x aantal films te beoordelen. Deze beoordelingen vormen de input voor de algoritmes achter het programma die de aanbevelingen genereren. Zoals in figuur 3 staat weergegeven, kan het voorkomen dat sommige films geen poster hebben. Dit komt doordat de poster niet te verkrijgen was via de gebruikte dataset. Als een gebruiker de film niet kent, dan hoeft de gebruiker geen sterren aan te vinken. Het programma blijft films genereren tot genoeg films zijn beoordeeld om aanbevelingen voor te stellen. Het aantal vereiste films staat op vijftien.



“Recommendations”-scherm

Na het hebben beoordeeld van films, komt de gebruiker op het onderstaande scherm terecht. Bij dit scherm zijn twee soorten aanbevelingen weergegeven, namelijk “User-based” en “Genre-based”. Per algoritme zijn twaalf films weergegeven. De gebruiker kan zo zelf beoordelen welke lijst beter aansluit op zijn/haar voorkeuren. Hierna volgen er geen schermen meer. Als de gebruiker weer inlogt, dan kan weer input verzameld worden om de aanbevelingen aan te scherpen.



6. Projectverantwoording

Dit hoofdstuk beschrijft hoe het project is verlopen. Het gaat hierbij in op de individuele bijdrage, de obstakels die ontstonden en de eisen die uiteindelijk niet geïmplementeerd konden worden.

6.1. Individuele verantwoording

Deze paragraaf beschrijft de activiteiten van elk projectlid. Deze dienen als individuele projectverantwoording, omdat het een overzicht geeft in de activiteiten per groepslid. Deze zijn per groepslid in chronologische vorm beschreven.

Yoeri

Tijdens het project is Yoeri vooral verantwoordelijk geweest met de Front-End applicatie en een aantal andere programmeertaken.

Allereerst is hij bezig geweest met het bouwen van een scraper voor het scrapen van filmposters. Doordat deze niet direct beschikbaar waren op het internet moesten ze via een API en een Python script worden gedownload. De API die hiervoor gebruikt werd is de API van de OMDB (Open Movie DataBase). Door gebruik te maken van het IMDB-ID konden deze via de API gescraped worden van het internet.

Daarnaast heeft Yoeri het content-gebaseerde algoritme geprogrammeerd. Dit algoritme genereert een aanbeveling aan de hand van de beoordelingen van de gebruiker en de metadata van de verschillende films. Samen met Mick heeft hij nog gekeken naar mogelijkheden om dit algoritme verder te verbeteren en efficiënter te laten werken. Uiteindelijk heeft hij deze geïmplementeerd in de applicatie.

Een groot deel van de tijd heeft Yoeri gewerkt aan de applicatie. Allereerst heeft hij ervoor gezorgd dat deze een verbinding had met de eerder beschreven databronnen. Daarnaast is hij bezig geweest met het uitdenken en programmeren van de applicatie. Hierna is hij verantwoordelijk geweest voor het koppelen van de programmeertaal PHP (waarin de applicatie is gebouwd) met de Python scripts. Zo konden deze algoritmes worden uitgevoerd tijdens het gebruik van de applicatie.

Samen met Mick heeft Yoeri aan het eind nagedacht over de samenvoeging en implementatie van alle uitgewerkte onderdelen. Deze heeft hij uiteindelijk geïmplementeerd. Samen met de andere projectleden heeft Yoeri aan het eind gewerkt aan het schrijven van een projectverantwoording (dit verslag) en het uitwerken van een demonstratie van de applicatie (de presentatie).

Gabriëla

Aan het begin heeft Gabriëla de taak gekregen om twee algoritmes te programmeren, namelijk "Collaborative User-based filtering" en "K-means". Deze twee algoritmes zijn in R geprogrammeerd, omdat zij geen ervaring had met Python. Met de gedachte dat R en Python later zouden samengevoegd, vormde het op dat moment geen probleem om de algoritmes in R te programmeren. Echter kwamen de leden later erachter dat de algoritmes in R nogal traag waren door de hoeveelheid data. Om deze reden is 'Collaberative User-based filtering' in Python geïmplementeerd. In Bijlage 7.2 is de code van de twee algoritmen weergegeven.

Verder heeft Gabriëla in het project getracht om verschillende talen te leren zoals, Python, HTML, PHP en CSS. Ook werkte ze mee aan de verslaglegging van het project.

Mick

Gedurende het project is Mick verantwoordelijk geweest voor een groot gedeelte van het programmeer werk.

Zijn eerste taak was het correct verkrijgen van de data. Hierbij hoorden taken zoals het opschonen van de data en het toevoegen van data via onder andere API's. Een voorbeeld hiervan is het toevoegen van een samenvatting van elke film via een API en deze samenvatting verwerken tot steekwoorden. Deze steekwoorden konden dan gebruikt worden voor het inhoud gebaseerde algoritme.

Zijn tweede taak was het inrichten van een Ubuntu server (Raspberry Pi) zodat deze als MySQL database gebruikt kon worden. Deze database heeft hij samen met Bilal ingericht, zodat de projectleden gebruik konden maken van de verkregen informatie. Het inrichten van de database is in het begin van het project gerealiseerd, zodat de projectleden bij de benodigde informatie konden.

Mick zijn derde taak was het realiseren en samenvoegen van de algoritmes die aanbevelingen genereren voor gebruikers van de applicatie. In totaal heeft Mick vier algoritmes gerealiseerd, waarvan twee uiteindelijk werden gebruikt voor de applicatie (Collaberative User filtering & KNN). Dit omdat sommige algoritmes erg traag waren door de grootte van de dataset.

Mick heeft gedurende het project veelal een ondersteunende rol gehad. In eerste instantie wilde hij graag een groter aantal algoritmes realiseren (bijvoorbeeld een Deep Learning algoritme), maar dit is niet gelukt doordat een aantal projectgenoten ondersteuning nodig hadden bij het uitvoeren van hun taken. De laatste taken bestonden uit het meewerken aan de verslagligging van het project.

6.2. Obstakels

Gedurende het project heeft de projectgroep een aantal obstakels ondervonden. De grootste obstakels waren:

- Ervaring met programmeren;
- Uitval van projectleden;

De ervaring met programmeren binnen de projectgroep was één van de obstakels. Sommige hadden ervaring met programmeren, maar niet op het niveau van een informaticastudent. Naarmate het project vorderde kwam dit obstakel steeds meer naar voren. Het had als resultaat dat een aantal projectleden veel tijd hebben gestoken in het begeleiden van andere projectleden en niet verder konden met hun eigen taken.

In het begin van het project is de student Joep Hoogsteden uitgevallen. Joep voelde zich niet op zijn plek en had zelf niet verwacht dat een groot gedeelte van de minor bestond uit programmeren. In de tweede periode van het project viel Bilal Aarabe uit. Bilal had niet het gevoel dat de minor bij hem paste en heeft na overleg met de projectgroep besloten om te stoppen. De uitval van twee projectleden zorgde voor een verhoging van de werkdruk onder de overblijvende projectleden.

Deze obstakels zorgde ervoor dat de studenten met kennis van programmeren veel bezig waren met het helpen van de andere projectleden. Door de uitval van meerdere projectleden en het ondersteunen van anderen moesten er een aantal doelen worden geschrapt.

6.3. Vervallen eisen

De obstakels hadden als resultaat dat een aantal eisen niet gerealiseerd zijn. De volgende eisen zijn niet gerealiseerd:

- Een gevarieerdere set van algoritmes om aanbevelingen te maken (zoals een Deep Learning algoritme)
- De applicatie biedt informatie over films zodra de gebruiker hierop klikt (hier is echter wel een ontwerp voor gemaakt, zie Bijlage 7.1)
- De applicatie maakt gebruik van een herhalend proces waardoor gebruikers aanbevelingen kunnen beoordelen en de applicatie nieuwe aanbevelingen genereerd.

6.4. Reflectie

Het succesvol afronden van het project was een uitdaging voor de projectleden. De groep was beperkt in mankracht en technische voorkennis.

Voor het project gaven een aantal van de projectleden aan dat deze wel in aanraking waren gekomen met programmeren, maar dit bleek niet op het niveau te zijn wat benodigd was voor een project als deze. De projectgroep had duidelijker moeten zijn in de benodigde kennis voor het project. Dit had als resultaat gehad dat er beter inzicht was in het kennen en kunnen van de gehele projectgroep. Het opnemen van een Informaticastudent in de projectgroep had het resultaat kunnen zijn van deze bevinding. Deze had Yoeri en Mick extra technische ondersteuning kunnen bieden.

De projectgroep is van mening dat het project succesvol is afgerond. De web-applicatie werkt met de geïmplementeerde algoritmes en genereert aanbevelingen voor gebruikers. Het wel of niet succesvol afronden is gebaseerd op het wel of niet realiseren van de eisen met een hoge prioriteit en de leercurve welke elk projectlid heeft ondervonden. De projectgroep is van mening dat elk projectlid op technisch gebied erg veel heeft geleerd. Een voorbeeld hiervan is dat één van de projectleden in het begin van het project geen ervaring had met Python en aan het einde van het project een aantal algoritmes heeft opgeleverd (in Python).

7. Bijlagen

Dit hoofdstuk geeft een uitwerking van de verschillende bijlagen.

7.1. Bijlage 1 – Dashboard

Een tool die gebruikt maakt van een dataset bevat veel informatie. In het filmproject is data beschikbaar bijvoorbeeld over het jaartal, plot en budget van de films. Om de gebruiker een beeld te geven met wat voor soort films in de database beschikbaar zijn, heeft het projectgroep besloten om een dashboard te construeren. Een dashboard is de manier om informatie overzichtelijk weer te geven.

Methodiek

Een dashboard kan gemaakt worden met diverse programmeertalen. De projectgroep heeft besloten om het dashboard te maken binnen PHP, omdat dit gemakkelijk te implementeren in de aanbevelingstool. Verder wordt HTML en CSS gebruikt om de lay-out te maken. Er is voor gekozen om Gabriëla de taak te geven om een dashboard te bouwen. Het maken van een interactief dashboard is een uitdaging voor iemand zonder HTML, CSS en PHP kennis, dus het begin draaide vooral om het leren van de talen.

Ontwerp

De tool heeft informatie over twee groepen, namelijk over de gebruikers en de films. In het ontwerp heeft de dashboard drie categorieën, namelijk “Movies visualisations”, “Top 5’s” en “Users visualisations”. De motivatie achter deze keuze ligt in het soort data dat de projectgroep en hoe dit gevisualiseerd zou kunnen worden.

In “Movies visualisations” worden vooral grafieken en diagrammen weergegeven dat diverse factoren in de data tegenover elkaar zet. Een voorbeeld van een visualisatie dat hierbij hoort is een bardiagram dat het budget en land tegenover elkaar zet.

Verder wordt in de “Top 5’s”-blok, zoals de naam al zegt, top 5’s binnen de dataset gepresenteerd. Denk hierbij bijvoorbeeld aan top 5 acteurs/actrices die in films voorkomen.

Ten slotte is een blok met “Users visualisation” data gepresenteerd die betrekking heeft op de gebruikers. Hierin staat onder andere hoe vaak een gebruiker films heeft beoordeeld en top 5 gebruikers met de meeste beoordelingen.

Verantwoording grafieken

Lijngrafiek

De projectgroep gebruikt lijngrafieken bij gegevens die weergegeven worden door de tijd heen. Het is zo gemakkelijk voor de lezer om het verschil te zien tussen tijdperioden.

Bardiagram

De bardiagrammen worden voornamelijk gebruikt om gegevens weer te geven per categorie. Een voorbeeld hierbij is het aantal films tegenover de afkomst qua land. Het resultaat is dat per land een balk is ter hoogte van het aantal films dat zijn uitgebracht in het betreffende land.

Spreiddiagram

Het spreiddiagram wordt gebruikt om de verdeling in de data weer te geven. Op deze manier kan de lezer zien of de data bijvoorbeeld zich naar een bepaalde verdeling voor doet, zoals de normale verdeling. Er kan ervoor worden gekozen om in deze visualisatie een regressielijn te weer te geven.

Taartdiagram

De taartdiagram gebruikt de projectgroep om de lezer in een oogopslag de verhoudingen van de meetwaarde in te nemen. Een voorbeeld van het gebruik van een taartdiagram is het weergeven van het geslacht van de acteurs/actrices in de database.

7.2. Bijlage 2 - K-means - R

De hoofdmap van het ingeleverde zip bestand bevat een folder met naam ‘unused_algo’. Deze map bevat een bestand met naam ‘Kmeans.R’. Dit is het bestand waarnaar wordt verwezen.

7.3. Bijlage 3 – Collaborative user-based filtering - R

De hoofdmap van het ingeleverde zip bestand bevat een folder met naam ‘unused_algo’. Deze map bevat een bestand met naam ‘Collaborative Filtering.R’. Dit is het bestand waarnaar wordt verwezen.

7.4. Bijlage 4 - Collaborative user-based filtering vanuit package - Python

De hoofdmap van het ingeleverde zip bestand bevat een folder met naam ‘unused_algo’. Deze map bevat een bestand met naam ‘lightFm.py’. Dit is het bestand waarnaar wordt verwezen.

7.5. Bijlage 5 - DBSCAN - Python

De hoofdmap van het ingeleverde zip bestand bevat een folder met naam ‘unused_algo’. Deze map bevat een bestand met naam ‘dbscan.py’. Dit is het bestand waarnaar wordt verwezen.