Big Movie

Projecthandleiding
van het Data Science project van
HBO-ICT 2e jaar

3 februari 2020 – 9 april 2020

versie 4 - 30/1/2020

Inhoud

Inleiding

De laatste tijd wordt meer en meer gebruikt gemaakt van grote verzamelingen gegevens om patronen te analyseren en te herkennen. Dit wordt “Big Data” of “Data Science” genoemd. Het is dus belangrijk dat een student Informatica hands-on ervaring krijgt met het hanteren van grote verzamelingen gegevens, de analyse daarvan, het opstellen van modellen of patronen in die gegevens en het presenteren van visualisaties en modellen.

Deze projectbeschrijving beschrijft het project dat in periode 2 door de 2e jaars studenten van de opleiding HBO-ICT wordt gedaan. In dit project komt de stof terug van de vakken uit deze periode (Analytics, Databases 2, Design Patterns, Statistiek, PPO) en ook van eerdere periodes.

Beschrijving basis

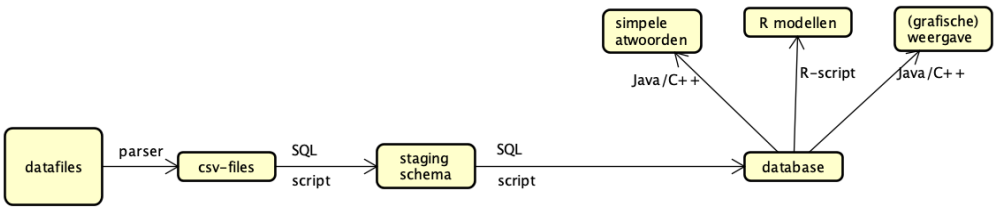
In het project “Big Movie” gaan jullie, studenten, van de gegevens van de Internet Movie Database (IMDB) een mooi gestructureerde database maken om de gegevens te kunnen presenteren. Vragen aan de database worden in een R-model gepresenteerd of een grafiek getoond, of rechtstreeks, bv '42'.

Daarvoor moeten de gegevens van IMDB (films, acteurs, regisseurs e.d.) geparsed worden om de fouten eruit te halen. De datafiles zijn niet (altijd) goed gestructureerd en er staan fouten in. Om de gegevens netjes in tabellen te krijgen moeten ze eerst in tijdelijke tabellen ingelezen worden (dit wordt ook wel staging genoemd), immers de data waarmee je begint zijn geen tabellen met goed gedefinieerde primaire en foreign keys. Daarna kunnen ze in goed gestructureerde tabellen ge-insert worden.

Vervolgens moeten er met behulp van R analytics-modellen gemaakt worden om trends uit de gegevens te halen of modellen op te stellen. Daarbij moet worden getoetst of de antwoorden statistisch betrouwbaar zijn. Tot slot moeten die trends of geaggregeerde gegevens op een aansprekende manier gepresenteerd worden.

Er is een pool (verzameling) met vragen over de filmdatabase. Een vraag uit de pool is bijvoorbeeld: “Welke acteur heeft het vaakst met Nicole Kidman in een film gestaan?”. Tenminste 10 van die vragen moeten in het project beantwoord worden. Je mag zelf kiezen welke vragen je wil beantwoorden. Zie voor meer informatie het hoofdstuk “Vragen”.

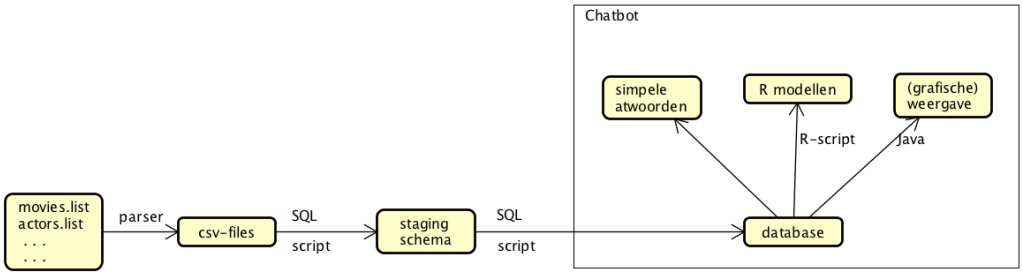
In Figuur 1 zie je een vereenvoudigd schema van het te bouwen systeem:



Figuur 1: basisschema

Extra

Je kunt een hoger cijfer halen als je het geheel van vragen en antwoorden in een chatbot presenteert, waarbij de vragen aan de database in een natuurlijke taal gesteld kunnen worden:



Figuur 2: schema met chatbot

Technische randvoorwaarden

Om het overzichtelijk te houden worden er een aantal technische randvoorwaarden gesteld aan het te bouwen systeem:

1) De parser van de ruwe IMDB-files moet geschreven worden in Java of C++. Je kan extra punten verdienen door hiervoor design patterns in te zetten.
2) De database moet worden gemaakt in PostgreSQL, MySQL of een ander RDBMS.
3) Indien code nodig is voor visualisatie van resultaten (bijvoorbeeld grafieken of geïllustreerde kaarten) moet gebruik worden gemaakt van Java.
4) Voor Analytics moet gebruik worden gemaakt van R.

Alternatief

Het kan zijn dat jullie met je groep een heel ander idee hebben en graag je eigen dataverzameling willen ontsluiten. Dat mag! Je moet dan een plan maken met de volgende elementen:

1) Welke dataverzameling wil je gaan gebruiken?
2) Wat zou de doelgroep moeten zijn waarvoor je de data wil analyseren.
3) Op wat voor een manier wil je de dat analyseren? (Wil je verbanden aantonen tussen de gegeven data?, wil je voorspellingen doen? Wil je trends aannemelijk maken?)
4) Welk platform wil je daarvoor gebruiken?

Dat plan moet dan door de modulecoördinatoren goedgekeurd worden.

Basisonderdelen

Het werk valt dus in een aantal stukken uiteen:
Parser (Java, C++)
SQL-script(s) voor het inlezen van .CSV files in de database
SQL-script(s) voor omzetten van raw-data naar goed gestructureerde tabellen
Relationeel databaseschema, inclusief ontwerp (klassendiagram, strokendiagram)
Maken van de database (PostgreSQL/ MySQL / ?).
R-script
SQL-queries
java/C++/R code voor grafische weergave (visualisatie code)
Het toetsen van een eigen hypothese (statistiek)
(extra:) java/C++/R code voor de interface met de chatbot.

Het maken van een analyse van dit systeem mbv het '4+1'-model (Kruchten 1995) is onderdeel van de opdracht. (zie toetsing en verslag)

Bemensing

groep
De groepen die de opdracht uitvoeren bestaan uit 5 of 6 studenten. 4 Daarvan zijn SE studenten en 1 of 2 BIM. Je mag zelf je groep samenstellen zolang je je zorgt dat de aantallen kloppen.

tutor
Elke groep krijgt een tutor toegewezen die het groepsproces begeleidt. Zij of hij spreekt de groep tenminste 1 keer per week. De tutor beoordeelt aan het eind van het project de samenwerking in de groep en (als nodig) de individuele bijdrage van iedere student aan het groepsproces.

expert
Er zijn bij dit project een aantal docenten die je specifiek vragen kunt stellen:
Wouter Brinksma: design patterns / java / C++
Gert Meijer: design patterns / java / C++
Karel Pieterson: design patterns
Wouter vd Ploeg: UML / statistiek
Martin Bosgra: databases
Martin Molema: parser
David Schweizer: analytics

Vragen

Als je het systeem hebt gemaakt moet je laten zien dat je met het systeem een aantal film-gerelateerde vragen kan beantwoorden. Daarvoor geldt het volgende:

- je moet van te voren bepalen welke vragen je gaat beantwoorden (en welke datafiles je daarvoor nodig hebt)
- je moet minimaal 10 vragen kunnen beantwoorden
- er moeten tenminste 2 modellen met R gemaakt worden
- er moet tenminste één hypothese getoetst worden
- er moeten tenminste 2 visualisaties gemaakt worden
- van die 10 moet je minimaal 5 geschikte vragen zelf (met je groep) bedenken (en beantwoorden).

Een paar voorbeelden van vragen zijn:

- Welke film heeft het langst geduurd om op te nemen?

- Welke acteur/actrice heeft de meeste dubbelrollen in één film (en welke film is dat)?

- Maak een interactief kaartje (b.v. google maps / openstreetview) met geboorteplaatsen van acteurs. Zodat op de kaart te zien is wie waar geboren is.

- Hypothese voorbeeld: Naarmate een actrice ouder wordt speelt ze in minder films.

De lijst van vooraf bedachte vragen is te vinden op Blackboard.

IMDB-bestanden

Er zijn twee verzamelingen data
1) (al een beetje gestructureerd): www.imdb.com/interfaces
2) (meer, maar minder gestructureerd): blackboard
(P3 Data science -> Project data science -> Content -> data files IMDB)
Dit zijn de begindata die je moet downloaden.
Voor een goed werkend systeem hoef je overigens niet ALLE bestanden te gebruiken. Uiteraard moet je zorgen dat je in elk geval de gegevens haalt die je nodig hebt (voor de vragen die je wilt beantwoorden). Misschien kan je in de lijst bestanden ook inspiratie vinden voor nieuwe (zelf bedachte) vragen.

Overige bronnen

Het kan zijn (afhankelijk van welke vragen je wilt beantwoorden) dat je andere gegevens wilt koppelen aan je filmdatabase. Denk bijvoorbeeld aan sites als Rotten Tomatoes (filmrecensies) of Box Office Mojo (commerciële gegevens). Voor het maken van kaartjes zul je gebruik moeten maken van bv Google Maps of OpenStreetMap. Verder heb je dan locatie-gegevens nodig in de vorm van coördinaten.

Als je dit soort verrijking wilt gebruiken in je systeem zul je wel zelf moeten onderzoeken hoe je daaraan komt en hoe de data moet worden gekoppeld aan de IMDB-data. De projectleiding beoordeelt dit overigens wel positief: de kans op een extra hoog cijfer stijgt als je ook gegevens uit andere bronnen gebruikt.

Verder zul je voor visualisaties (grafieken en dergelijke) een geschikt pakket moeten opspeuren en in je programmatuur opnemen.

Copyright

Gebruik van IMDB-data is alleen toegestaan onder bepaalde voorwaarden. De juridische details kun je vinden op http://www.imdb.com/help/show_leaf?usedatasoftware.
Zorg dat je je hieraan houdt!

Toetsing

In dit project word je op een aantal onderdelen getoetst.

1) De ***tutor*** van je groep bekijkt hoe het in jullie groep gegaan is. Is de samenwerking goed verlopen, is de planning en werkverdeling op een goede manier gedaan, hoe was de sfeer?
2) Elke groep krijgt een ***technisch-assessment.*** De groep wordt ge-assessed over technische zaken; beoordeeld worden de geschreven code, het ontwerp van het systeem als geheel en het ontwerp van de systeemonderdelen,
3) Tijdens het ***technisch-assessment*** worden ook individuele vragen gesteld. Dit leidt tot een individueel cijfer voor de code die door elke student aangeleverd wordt, zie ook de rubrics voor dit project.
4) Er moet een ***verslag*** gemaakt worden met
    a. de analyse van het sysyeem mbv het '4+1'- model (Kruchten 1995).
    b. Een verantwoording van de statistische toets
    c. de databasestructuur

Het totaalcijfer dat je op het project krijgt wordt dus bepaald door het **assessment-cijfer** (voor de groep), het cijfer voor de bijdrage aan de **code** (individueel), het **tutoroordeel** (groep en individueel) en het cijfer voor de **verslag** (groep). Let op: Hoewel het in beginsel groepscijfers zijn, kan het tutoroordeel en het assessment-cijfer per student individueel afwijken (naar boven of naar beneden) als daar aanleiding voor is.

Systeemoverzicht

Om het project uit te voeren moeten er veel verschillende soorten producten worden gemaakt:

Parser: de Java/C++ code om van de (“vervuilde”) IMDB-files ‘schone’ .CSV files te maken

UML-analyse: De UML-analyse moet bestaan uit de 5 elementen van het '4+1' model.

SQL-scripts: het ene SQL-script moet de schone .CSV files in de database zetten (de raw data, de ruwe gegevens), het andere script moet er voor zorgen dat van die raw data goed gestructureerde tabellen gemaakt wordt. Ook kun je scripts maken om views in je database te definiëren.

Visualisatiecode: de java/C++/R code waarmee je de gegevens uit de database grafisch presenteert. Omdat het over veel gegevens gaat moet je een 'vertaling' maken door middel van grafieken, of trends weergeven. Duidelijkheid en visuele aantrekkelijkheid zijn belangrijke criteria hiervoor.

SQL-queries: de queries (en eventueel PL/pgSQL scripts) die je gebruikt om gegevens uit de database te halen voor de grafische weergave en voor het beantwoorden van de (minimaal) 10 vragen.

R-scripts: Zoals bij analytics gebeurt moet je de modellen die je in R maakt opslaan in de vorm van een script (in feite dus de verschillende commando's in R op een rijtje). Het is toegestaan om vanuit een ander programma R aan te roepen. Zoek zelf uit hoe dat moet.

Databasestructuur: het relationeel database schema, zeg maar de tabellen, met daarbij natuurlijk de primaire en referentiële sleutels (primary keys en foreign keys) en indices. Dit laat je zien in een geschikte weergave zodat de structuur van de database goed te zien is.

(extra) chatbot-interface: De interface tussen de natuurlijke taal chatbot en de database.

Assessment

Het assessment is een technisch assessment en duurt voor een groepje 60 minuten. In het assessment komen aan de orde:

- de Java code,
- de SQL-scripts en -queries,
- de R-scripts,
- (extra) de chatbot
- de databasestructuur (tabellen en hun onderlinge verbanden),
- het gemaakte systeem als geheel.

Het is verplicht om bij elk stuk code aan te geven wie dat stukje code geschreven heeft. Tijdens het assessment wordt daar per groepslid ook persoonlijk naar gevraagd.

Jullie krijgen een cijfer voor de manier waarop de groep het systeem heeft geïmplementeerd, en een cijfer voor je eigen bijdrage aan de code. Het eerste cijfer kan dus hoger (of lager) zijn dan het tweede cijfer.

De code moet tijdig, dat is uiterlijk eind van de week vóór de assessments worden ingeleverd via Blackboard. Zie de planning verderop in dit document.

Uiteraard moeten de vragen, de resultaten en de manier waarop je het gedaan hebt aan bod komen.

Verslag

In het verslag moet staan:

- Het '4+1-model' (Kruchten 1995) met per diagram een kleine toelichting.
- De gestelde vragen met antwoorden.
- Het toetsen van de statistische hypothese.
- De databasestructuur.

Tutoroordeel

Zoals gewoonlijk heeft elke groep een tutor. Deze tutor moet in staat zijn om jullie groepsproces te zien en te beoordelen. Daarom is het handig om haar of hem zo snel mogelijk te betrekken bij jullie werkzaamheden.
Let op: Als de tutor niet in staat is geweest om zien hoe jullie aan het project gewerkt hebben, of niet kon zien welke bijdrage je op individuele basis geleverd hebt[1] kan de tutor een onvoldoende geven aan de groep of aan een of meer personen uit de groep.

Weging

Uiteindelijk komen er voor elke student vijf cijfers. Zie hiervoor onderstaande tabel:

	Onderdeel	Wijze toetsing
1	Technische kwaliteit gemaakte producten	Technisch assessment (groep)
2	Code kwaliteit en hoeveelheid	Technisch assessment (individueel)
3	Kwaliteit verslag (PDF)	Beoordeeld door projectleiding
4	Groepsproces en groepsbijdrage	oordeel tutor (groep/individueel)

Deze vier cijfers worden gemiddeld tot een eindcijfer (alle onderdelen wegen dus voor 25% mee).
Elk van deze cijfers moet voldoende zijn om een voldoende op het project te kunnen halen.

Minumum

Wat moet je minmaal doen om op elk onderdeel een voldoende te halen?
1) Technisch: het moet werken en je moet kunnen uitleggen hoe elk onderdeel werkt.
2) Code: moet objectgeoriënteerd zijn, voldoen aan code-conventie, niet te veel nesting, niet te lange methoden, leesbaar, van commentaar voorzien etc.
3) De tutor en de andere groepsleden zijn niet ontevreden.

Bonuspunten

Je kunt op verschillende manieren bonuspunten krijgen. (het krijgen van bonuspunten betekent niet dat je op een ander onderdeel een onvoldoende mag halen)
1) door een bijzonder mooi geintegreerde chatbot/app te maken
2) door een mooi creatief filmpje te maken

[1] Tip: dit kan ook blijken uit rapportages / screenshots vanuit bv. GitHub

3) door op een andere manier een eigen speciale invulling aan de opdracht te geven

Op te leveren producten (deliverables)

De onderstaande producten moeten worden opgeleverd gedurende dit project:

Parsercode (Java/C++)	20 maart
SQL-scripts (.txt of .rtf of .docx)	27 maart
Verslag (PDF)	1 april (nee geen grap)
Code voor eindproduct: SQL-queries+antwoorden visualisatie-code (Java/C++/R) analytics- (R-scripts)	1 april
Demo-film	1 april
Interface chatbot (eventueel!)	1 april

Alle producten kunnen ALLEEN worden ingeleverd via **BLACKBOARD**! Je moet op tijd inleveren, anders kan de projectleiding je producten niet tijdig beoordelen en kan het gebeuren dat je geen cijfer krijgt voor het project. De deadlines staan in de planning hieronder en ook bij de assignments op Blackboard.

Let op: bij het inleveren van code (en daaronder verstaan we ook SQL en R-scripts) moet je ervoor zorgen dat uit comments blijkt wie welk stukje code heeft geschreven. Dit is nodig om te kunnen beoordelen of iedereen voldoende heeft bijgedragen aan het eindproduct.

Planning

datum	**week**	**wie**	**wat**	**resultaat**
3/2	1/6	allen	kick-off	groepen
13/2	2/7	allen	workshop parser	
26/2	4/9	allen	workshop onderzoek	
	5/10		*geen* workshop	
16/3	6/12	groep	daadwerkelijke start project	planning
		groep + tutor	bijeenkomst	procesmeting
20/3		***groep***	***Parsercode (Java/C++) (Blackboard)***	deelcijfer
		groep + tutor	bijeenkomst	procesmeting
27/3	7/13	***groep***	***SQL-scripts***	deelcijfer
		groep + tutor	bijeenkomst	procesmeting
1/4		allen	aftrap project IDP/robotica	
1/4	9/14	***groep***	***Verslag***	deelcijfer
		groep + tutor	bijeenkomst	procesbeoordeling
1/4		***groep***	***inleveren code SQL-queries+antwoorden visualisatie-code (Java/C++/R) analytics- (R-scripts) [evt. Chatbot]***	deelcijfers
7/4	10/15	allen	gezamelijke afronding	deelcijfer
		expert+ tutor	Beoordelen	deelcijfers
7/4		allen	technisch assessment	eindcijfer

Bibliogafie:

- Kruchten, Philippe Architectural *Blueprints — The "4+1" View Model of Software Architecture.*, IEEE Software 12 (6), pp. 42-50, 1995.
- van der Ploeg, W. - *Het maken en controleren van een UML-analyse*, versie 2.0, eigen beheer, Leeuwarden, 2018