# Analyzing recurring events

Arno Moonens
Richting: 3e Bachelor Computerwetenschappen
Email: arno.moonens@vub.ac.be
Rolnr.: 500513

Promotor: Yann-Michaël De Hauwere
Begeleiders: Maarten Devillé en Peter Vrancx

Academiejaar: 2014-2015

# Inhoudsopgave

# 1 Abstract

Men gaat hier naar relaties zoeken tussen informatie over *tweets*, films en hun metadata en box office informatie over films. Allereerst werd er data verzameld van Twitter via hun API. Ook metadata van films werd opgezocht m.b.v. een API voor *Rotten Tomatoes* en *IMDb*. Box office data werd daarentegen verzamel d.m.v. het scrapen van de site *Box Office Mojo*.
Voor het verwerken van tekst uit *tweets* werd *sentiment analysis* uitgevoerd m.b.v. een *naive Bayes classifier*. Hierbij gaat men een training voorbeeld het label geven dat de grootste kans heeft om voor te komen gegeven bepaalde waarden voor de features. Deze *classifier* is uiterst geschikt voor het classificeren van tekst en kan gebruikt worden voor tekst van *tweets* door de woorden van deze tekst als features te gebruiken. Er word ook lineaire regressie gebruikt bij het verwerken. Hier gaat men naar een rechte met als functie $f(x) = ax+b$ zoeken met bepaalde waarden voor $a$ en $b$ zodat er de kleinst mogelijke fout gemaakt wordt van voorspelde output t.o.v. werkelijke output.
Bij het bekijken van het aantal geplaatste *tweets* per dag van een film zag men dat er een piek was tijdens de dagen voor en na de release van die film. Men kan ook zien dat *tweets* met geolocatie vooral geplaatst worden in Noord-Amerika en Europa, wat te verklaren is door het feit dat daar het meeste mensen Twitter gebruiken en er vooral data van westerse films verzameld is.
Zoals verwacht daalt de opbrengst van een film en het aantal *tweets* over die film per week, om na een aantal weken constant te blijven. Er is echter geen direct verband tussen het gemiddeld aantal *tweets* per dag een week voor de release van een film en de opbrengst in de week na de release van die film. Ook tussen de *sentiment* en de *Rotten Tomatoes* score, de *sentiment* en de *IMDb* score en tussen de *Rotten Tomatoes* score en de box office is geen verband te merken.
Na het bekijken van de *sentiment* voor en na de release van een film bleek dat er veel *retweets* in de database aanwezig waren en dat dit veel invloed heeft op het resultaat omdat elke groep van *retweets* ook hetzelfde geclassificeerd werd. Daarom werden deze gefilterd. Lineaire regressie op het resultaat hiervan leidde echter niet tot goede voorspellingen. Men kan wel zien a.d.h.v. de impact van *retweets* van trailers dat sociale media als communicatieplatform een grote rol spelen bij het promoten van films door filmproducenten.

# 2 Inleiding

Films zijn steeds een bron voor veel emotie: Men heeft al dan niet hoge verwachtingen. Achteraf zijn deze verwachtingen echter niet altijd ingelost.
Een van de middelen waarmee mensen hun gevoelens over films uiten is Twitter. We gebruiken Twitter dan ook als voornaamste bron om de relatie tussen films en mensen en het succes van film te analyseren.

In de voorbereiding van mijn bachelorproef werd vooral clustering besproken, samen met de werking van, de performantie en de voor- en nadelen van K-Means clustering.
Clustering is een vorm van unsupervised learning. Men gaat geen waarden gaan voorspellen, maar wel reeds aanwezige data gaan groeperen. Bij de bachelorproef bleek het echter nuttiger te zijn om aan supervised learning te doen. Hierbij heeft men voorbeelden met input- en outputwaarden waarvan men kan leren en moet men nadien van ongeziene data de output gaan voorspellen m.b.v. de input.
Omdat clustering hier niet nodig bleek te zijn is er dan ook geen gebruik van gemaakt.

Allereerst wordt er besproken hoe de data die men gebruikt voor analyses verzameld werd.
Men begint bij het bespreken van data van Twitter. Er wordt uitgelegd hoe men data van het internet haalt, hoe deze gemanipuleerd wordt en tenslotte ook hoe men ze opslaat.
Er was ook een manier nodig om metadata over een film te verzamelen, zoals bijvoorbeeld de scores die gebruikers geven, de acteur die meedoen,. . . Hiervoor gebruikt men 2 sites als bronnen: *IMDb* en *Rotten Tomatoes*. Er wordt ook besproken hoe de data van deze sites verzameld werd, maar niet hoe ze opgeslagen wordt omdat dit ook niet nodig bleek te zijn.
Er moest ook nog verzameld worden hoeveel een film heeft opgebracht in een bepaalde week. Hiervoor werd de site *Box Office Mojo* gebruikt. Ook hier werd alleen besproken hoe de data verzameld werd en niet hoe ze opgeslagen werd.

Vervolgens wordt er uitgelegd welke onderdelen men van machine learning gebruikt die geholpen hebben bij het analyseren van informatie.
Een van deze onderdelen is *sentiment analysis*. We wilden graag bepalen of een bepaald stuk tekst, bij ons de tekst van een *tweet*, eerder positief of negatief is. Men wil de tekst dus classificeren. Hiervoor wordt een *naive Bayes classifier* gebruikt. Dit algoritme wordt dan ook uitgelegd, om nadien te kijken hoe dit getraind en toegepast werd.
Een ander onderdeel is lineaire regressie. Bij regressie willen we numerieke waarden voorspellen die continu van aard zijn. Ook hier bespreken we de werking van dit algoritme.

Tenslotte gaan we verzamelde data gaan analyseren.
We beginnen eerst met simpelweg te kijken hoeveel *tweets* er per dag worden geplaatst en wanneer dit aantal maximaal is. Hierna bekijken we ook waar *tweets* met geolocatie zoal verstuurd worden.
Vervolgens gaan we *sentiment analysis* gebruiken en bekijken we het percentage positief geclassificeerde *tweets* per dag. Deze *sentiment analysis* wordt ook ge-

bruikt om te kijken of men positiever of negatiever is na de release van een film dan ervoor.
Hierna werd de *sentiment analysis* van alle *tweets* (zonder *retweets*) van een film gecombineerd met de gebruikersscores van allereerst *IMDb* en daarna *Rotten Tomatoes* om te kijken of er een relatie is tussen beide.
Vervolgens werd informatie i.v.m. de opbrengst van films gebruikt. Men begint met te bekijken hoe het aantal *tweets* per week en de opbrenst van een film per week zich verhouden. Hierna gaat men opnieuw *sentiment analysis* gebruiken en kijken we of we de opbrengst van een film kunnen voorspellen a.d.h.v. het percentage positief geclassificeerde *tweets* die geplaatst zijn voor de release van een film. Nadien probeert men dit opnieuw met het gemiddeld aantal *tweets* per dag in de week voor de release van een film.
Tenslotte werden de *tweets* eens niet gebruikt en werd er gezocht naar de relatie tussen de gebruikersscore op *Rotten Tomatoes* van een film en de opbrengst van die film in de eerste week na de release.

# 3 Het verzamelen en opslaan van data

Om enige analyse te verrichten i.v.m. *tweets* over films moeten deze *tweets* uiteraard eerst verzameld worden en moet men ook metadata over films verzamelen. Deze 2 onderdelen worden hier besproken.

## 3.1 Twitter

Op Twitter kan men berichten, genaamd *tweets*, van maximaal 140 tekens plaatsen die standaard voor iedereen te lezen zijn.
Een *tweet* bevat buiten de tekst zelf nog hele reeks aan metadata. Het bevat onder andere het moment van plaatsen, de auteur van de *tweet*, de locatie waarop de *tweet* is verstuurd,... Deze informatie is echter makkelijk te verkrijgen via de *Twitter API* [1], meerbepaald met het *Search API* gedeelte [2]. Het antwoord van een aanvraag aan deze *API* is steeds in *JSON*-formaat.
Voor elke film werden *tweets* opgezocht a.d.h.v. een zogenaamde *hashtag*. Dit is een woord voorafgegaand door een hekje(#). Het woord was voor de meeste films gewoon de titel van deze film zonder spaties. De zoekterm voor de film *Beyond The Lights* bijvoorbeeld werd dan *#BeyondTheLights*.

Voor het opslaan van de *tweets* werd er een *NoSQL* database gebruikt, namelijk *MongoDB* [3]. In tegenstelling tot een *RDBMS* is er hier geen schema dat restricties oplegt aan de structuur of de types van de data [4]. Data kan echter wel gegroepeerd worden m.b.v. een collectie. Dit komt overeen met een tabel in een *RDBMS*. De data zelf noemt men documenten.
Er werd voor *MongoDB* gekozen omdat men alleen *tweets* moest opslaan een geen relaties moest maken tussen andere *tweets*, ... *MongoDB* bevat daarbuiten ook een functie om te zoeken in documenten waardoor het aan alle

vereisten voldeed om de data correct op te slaan om later makkelijk te kunnen verkrijgen.
Elke film kreeg zijn eigen collectie in de database en *tweets* werden dan ook in een bepaalde collectie geplaatst afhankelijk van de film waarvoor ze opgezocht werden. Ondanks dat de *Python* module voor *MongoDB* toelaat om rechtreeks *JSON* objecten aan een collectie toe te voegen, werd er eerst nog een aanpassing gedaan.
Elke *tweet* die men terugkrijgt van de *Search API* bevat een *id* veld [5]. De naam van dit veld werd veranderd van 'id' naar '_id'. Dit werd gedaan omdat *MongoDB* zijn data opslaat in *BSON* (*Binary JSON*) formaat waarbij het '_id' veld van een document gebruikt wordt als een sleutel die uniek is voor elk document in de database [6]. Als dit veld nog niet aanwezig is in het toe te voegen document, wordt dit eerst automatisch gegenereerd en toegevoegd. Dit is echter niet nodig omdat het 'id' van een *tweet* reeds uniek is. Het volstaat dus om het 'id' veld van naam te veranderen.
Buiten deze kleine aanpassing wordt alle ontvangen data exact zo toegevoegd aan de database.

De *Twitter API* had echter wel een aantal beperkingen. Zo was het maar mogelijk om met de *Search API tweets* van maximaal 7 dagen terug op het moment van zoeken te verkrijgen. Hierdoor moest men op tijd beginnen met *tweets* te verzamelen als men genoeg data wou hebben.
Een andere limiet was het aantal tweets dat men kon verzamelen in 15 minuten. Zo kan men namelijk 'maar' 180 aanvragen doen in 15 minuten en wordt elke aanvraag beantwoord met maximaal 100 *tweets*. Dit resulteert in 18000 *tweets* per kwartier. Als men *tweets* van films opvraagt waar veel over gesproken wordt op *Twitter* kan het in het slechtste geval dus enkele uren duren om de *tweets* ervan te verzamelen.
Dit probleem werd echter opgelost door een *SSD server* te gebruiken die elke dag met een *Python* script *tweets* verzamelde. Hierdoor moest men zelf geen computer laten open staan.

## 3.2 Rotten Tomatoes en IMDb

Om meer info van films te verzamelen en te kunnen linken, werd data van 2 verschillende sites gebruikt: *Rotten Tomatoes* [7] en *IMDb* [8]. Deze tonen allebij een korte inhoud, trailer, acteurs, director, ...
Er kunnen op beide sites ook een score gegeven worden aan deze films. Ze hebben beide een score die gegeven wordt door iedereen en een gegeven door *critics*. Dit zijn mensen die professioneel bezig zijn met films en er vaak over berichten in de media [9].
De kijkersscore op *IMDb* die op IMDb te zien is, is simpelweg het gemiddelde van de scores die door niet-*critics* gegeven werden. De *critics* score, op *IMDb* de *metascore* genoemd, is gelijkaardig aan de vorige, maar maakt uiteraard

gebruik van de score die *critics* hebben gegeven.
Bij *Rotten Tomatoes* werkt het iets anders. De kijkersscore is hier het percentage van mensen die de film minstens een score van 3,5/5 hebben gegeven. De *critics* score, genaamd de *Tomatometer*, is dan weer het percentage van *critics* die de film een positieve beoordeling hebben gegeven. Werknemers van *Rotten Tomatoes* zelf beslissen of een beoordeling al dan niet positief is [10].

*IMDb* biedt zelf geen *API* aan, dus was het nodig om een onofficiële *API* [11] te gebruiken. Deze *API* biedt echter voldoende info aan om te kunnen gebruiken voor de experimenten die hier uitgevoerd worden, zoals bijvoorbeeld de scores, de release datum in de Verenigde Staten van Amerika, de *cast*, een samenvatting,... Het resultaat zelf is ook in *JSON*-formaat.
*Rotten Tomatoes* biedt daarentegen wel zelf een *API* aan [12]. Deze geeft gelijkaardige informatie terug als de *IMDb API*, maar uiteraard met andere scores en met een andere structuur. Het resultaat zelf is eveneens in *JSON*-formaat.

De info van beide sites wordt niet opgeslagen in een database, maar wordt rechtstreeks opgehaald via de nodige *API*. De data werd niet opgeslagen omdat de scores van de films steeds kunnen veranderen naar mate er meer mensen de film in kwestie beoordelen. Het was voor mijn experimenten ook niet nodig om de score op verschillende momenten in de tijd op te vragen en op te slaan.

## 3.3 Box Office Mojo

Buiten *tweets* over films en de algemene metadata van films was het ook nodig om meer specifieke data te verkrijgen van een film, namelijk de *box office* van een film per week. Hiermee bedoelt men hoeveel een film heeft opgebracht per week en wordt gemeten a.d.h.v. de opbrengst van de kaartenverkoop in de cinema's.
Voor deze informatie is er geen *API* beschikbaar dus moest deze data verkregen worden via *scraping* van de site *Box Office Mojo* [13]. Bij dit proces wordt een *URL* geladen met als resultaat een *HTML*-pagina en wordt er in deze code, eventueel m.b.v. reguliere expressies, gezocht naar de data die men nodig heeft. Dit is anders dan het gebruik van een *API* omdat men hier geen duidelijke en consistente structuur heeft als antwoord en het ook niet duidelijk is wat men altijd van antwoord kan verwachten. Zo is het moeilijker te detecteren of de pagina of nodige data wel geladen en dus te vinden is.

Om de *box office* data te verkrijgen werd de site *Box Office Mojo* gebruikt. voor elke film werd de pagina van deze film geladen die de *box office* data van verschillende weken bevatte in de vorm van een tabel. M.b.v. de *scraper* werden de weeknummers en de bedragen in Amerikaanse dollar uit deze tabel gehaald.
De *box office* data is slechts die in de Verenigde Staten van Amerika, omdat

de release datum van films ook steeds diegene uit dat land is en beide data tesamen gebruikt worden bij het analyseren.
Deze data werd ook niet opgeslagen, omdat men niet alleen box office resultaten van de vorige week kan ophalen, maar van alle weken dat de film in de cinema's is geweest. Omdat er elke week informatie bijkomt moeten de resultaten ook telkens opnieuw opgehaald worden.

# 4 Hulpmiddelen voor analyse

## 4.1 Sentiment analysis

De datum en tijdstip waarop een *tweet* geplaatst en het aantal tweets geplaatst (in een bepaalde periode) is reeds nuttige informatie om bepaalde conclusies te kunnen trekken, maar een *tweet* heeft uiteraard als doel om een bepaalde boodschap te verkondingen m.b.v. tekst. Het is dan ook nuttig om deze tekst te gaan ontleden en analyseren.
Een van de zaken die we kunnen zeggen over een de tekst van een *tweet* is of deze een positieve boodschap verkondigt of eerder een negatieve. Doordat mijn database een paar miljoen *tweets* bevat is het natuurlijk onbegonnen werk om elke *tweet* apart te beoordelen en deze eigenschap toe te voegen. Daarom maakt men gebruik van supervised learning en meerbepaald een classificeringsalgoritme om de tekst van een *tweet* te beoordelen.
Bij supervised learning heeft men verschillende training voorbeelden met input waarden (features) en output waarden (labels) waar men op kan trainen. Men kan dit zien als een tabel waarbij de rijen de training voorbeelden zijn en de kolommen de features en (een) label(s). Het is dan na het leren de bedoeling om het label van een ongezien voorbeeld te voorspellen a.d.h.v. de waarden van de features.
Bij classificering zijn, in tegenstelling to regressie, maar een beperkt aantal waarden als label mogelijk. Deze waarden kunnen numeriek zijn, maar dit is niet verplicht en is hier ook niet het geval.

Het classificeringsalgoritme dat gebruikt wordt is een *naive Bayes classifier*. Dit algoritme is vrij goed in het classificeren van tekst en wordt daar dan ook veel voor gebruikt [17].
We beginnen eerst met het algoritme zelf uit te leggen alvorens het algoritme toe te passen op ons classificeringsprobleem.

### 4.1.1 Uitleg naive Bayes classifier

Bij het Bayesiaans classificeren van een voorbeeld gaat men als label de waarde toewijzen die het meest waarschijnlijk is ($v_{MAP}$), gegeven de waarden van de

features van dat voorbeeld. We kunnen dit schrijven als volgt [18]:

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j | a_1, a_2 \dots a_n)$$

Met $v_j \in V$ bedoelt men elke mogelijke output waarde en $a_1, a_2 \dots a_n$ zijn waarden van features van het te classificeren voorbeeld. M.b.v. het Bayes theorema ($P(A|B) = \frac{P(B|A)P(A)}{P(B)}$) kunnen we dit herschrijven als volgt:

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(a_1, a_2 \dots a_n | v_j) P(v_j)}{P(a_1, a_2 \dots a_n)} \tag{1}$$

$$= \underset{v_j \in V}{\operatorname{argmax}} P(a_1, a_2 \dots a_n | v_j) P(v_j) \tag{2}$$

Het is makkelijk om $P(v_j)$ te berekenen omdat dit simpelweg de frequentie is waarmee de output waarde $v_j$ voorkomt in de training data. Het probleem is echter dat het totaal aantal termen in het product gelijk is aan het aantal mogelijke voorbeelden vermenigvuldigd met het aantal mogelijke output waarden. We moeten daarom elk mogelijk training voorbeeld meerdere keren zien om betrouwbare schattingen te maken. De *naive Bayes classifier* gaat er echter van uit dat features conditioneel onafhankelijk zijn gegeven de output waarde. Bijgevolg is $P(a_1, a_2 \dots a_n) = \prod_i P(a_i | v_j)$. We kunnen daarom de formule als volgt herschrijven:

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j) \prod_i P(a_i | v_j)$$

De leerstap van dit algoritme bestaat er uit om de termen in bovenstaand product te schatten. Deze schattingen vormen dan de hypothese die men kan gebruiken om ongeziene voorbeelden te classificeren. Het label voor de tekst ($v_j$) is simpelweg datgene dat $P(v_j) \prod_i P(a_i | v_j)$ kan maximaliseren.

### 4.1.2 Toepassing

De *naive Bayes classifier* werd toegepast in Python op 2000 film reviews, geschreven in het Engels, uit de *nltk.corpus* module [19]. 1500 van deze reviews werden gebruikt om te trainen en 500 om het resultaat na het trainen te testen. Deze reviews bevatten elk een lijst van woorden en een veld waarin staat of ze al dan niet positief zijn. Van elke review werd dan een bruikbaar training voorbeeld gemaakt door als features de woorden van in de review te gebruiken en als waarde *True* om aan te duiden dat ze wel degelijk in de review staan. Als output waarde werd 'pos' of 'neg' ingevuld om aan te duiden of de review respectievelijk positief of negatief was. Men heeft dus wel degelijk met classificering te maken waarbij men slechts 2 mogelijke output waarden heeft. Het *naive Bayes classifier* algoritme kan dus gebruikt worden als men zegt dat $v_j \in \{'pos', 'neg'\}$ en voor $a_i$ het i'de woord van de review in kwestie.
Na het trainen werd het algoritme getest op de overige 500 voorbeelden. Dit

betekent dat men deze voorbeelden gaat voorspellen en dat men gaat kijken of het voorspelde label overeenkomt met het echte label van dat voorbeeld. Na het trainen was mijn algoritme voor 72,8% accuraat op de test data. Dit betekent dat het algoritme 72,8% van de voorbeelden uit de test data juist heeft kunnen voorspellen.

Met het resultaat na het trainen van dit algoritme kan men dus de tekst van tweets gaan classificeren als zijnde positief of negatief. Hiervoor moet men eerst features uit de tekst halen. Dit doet men door eerst de tekst te splitsen in woorden en ze aan een *dictionary* toe te voegen (met opnieuw als naam het woord zelf en als waarde *True*) indien het woord minstens 3 tekens lang is.
Het resultaat van deze extractie wordt dan aan de *classifier* gegeven als input, waarna men dan dezelfde labels kan terugkrijgen als bij de training voorbeelden, namelijk 'pos' of 'neg'.
*Sentiment Analysis* wordt alleen uitgevoerd op *tweets* geschreven in het Engels, omdat de features en dus woorden van de film reviews waarop getraind is ook Engelstalig zijn.

## 4.2 Lineaire regressie

Bij lineaire regressie is het de bedoeling om te kijken of er een verband is tussen de waarden van features van training voorbeelden en hun labels. Zoals de naam al suggereert doet men hier, in tegenstelling tot deel 4.1, aan regressie.
Bij *regressie* zijn de labels, de output waarden die men dus wilt voorspellen, numeriek en continu van aard. Men kan bijvoorbeeld de prijs van een huis proberen te voorspellen afhankelijk van de grootte en eventueel nog andere features [14].

We beperken ons hier tot het uitleggen van lineaire regressie met slechts 1 feature en 1 label. Het resultaat zal uiteindelijk een rechte zijn. Deze heeft de volgende vorm:

$$f(x) = ax + b$$

Hierbij stellen $a$ en $b$ constanten voor, is $x$ de feature en is het resultaat van de functie, $f(x)$, het voorspelde label.
De bedoeling is, net zoals bij vele andere *machine learning* algoritmen, dat het voorspelde label zo weinig mogelijk verschilt van het werkelijke label. Om te meten hoe goed het resultaat is van de functie gebruikt men de $R^2$ score [15]. De formule hiervan is de volgende [16]:

$$R^2 = \sum_{i=1}^{n} [y_i - f(x_i)]^2$$

In deze formule is $y_i$ telkens het werkelijke label van voorbeeld $i$ en is $f(x_i)$ het voorspelde label van dat voorbeeld.

Als we de functie $f(x_i)$ uitschrijven kunnen we zien dat we $R^2$ ook kunnen berekenen in functie van $a$ en $b$:

$$R^2(a,b) = \sum_{i=1}^{n} [y_i - (ax_i + b)]^2$$

Het is uiteraard de bedoeling dat men de labels zo correct mogelijk voorspeld en dat de $R^2$ score dus zo laag mogelijk is. Om $a$ en $b$ hiervoor te berekenen gaan we deze eerst afleiden uit bovenstaande formule:

$$\frac{\partial(R^2)}{\partial a} = -2 \sum_{i=1}^{n} [y_i - (a + bx_i)] = 0 \tag{3}$$

$$\frac{\partial(R^2)}{\partial b} = -2 \sum_{i=1}^{n} [y_i - (a + bx_i)]x_i = 0 \tag{4}$$

Dit brengt ons bij volgende vergelijkingen:

$$na + b \sum_{i=1}^{(} n)x_i = \sum_{i=1}^{(} n)y_i \tag{5}$$

$$a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{(} n)x_i^2 = \sum_{i=1}^{n} x_i y_i \tag{6}$$

Na berekeningen vinden we voor $a$ en $b$ uiteindelijk:

$$a = \frac{\overline{y}(\sum_{i=1}^{n} \overline{x}_i^2) - \overline{x} \sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 - n\overline{x}^2} \tag{7}$$

$$b = \frac{(\sum_{i=1}^{n} x_i y_i) - n\overline{x}\overline{y}}{\sum_{i=1}^{n} x_i^2 - n\overline{x}^2} \tag{8}$$

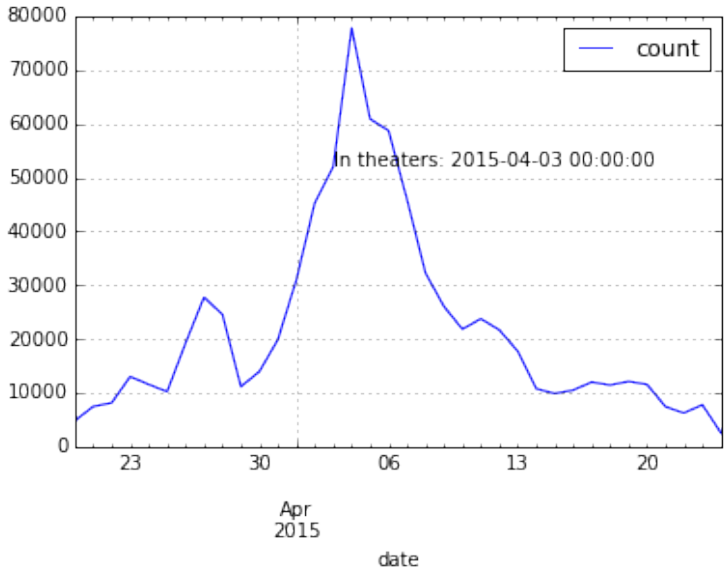
Door de gevonden waarden voor $a$ en $b$ dan in te vullen bekomt men een functie om labels te gaan voorspellen.

# 5 Analyzes van data

Het verzamelen en opslaan van data en *sentiment analysis* toepassen op tekst zijn uiteraard slechts hulpmiddelen om het uiteindlijke doel van deze bachelorproef te volbrengen, namelijk het analyzeren van de data.
We beginnen eerst met de data op zich te bekijken a.d.h.v. grafieken. Hierna gaan we over tot het combineren van data van verschillende bronnen om zo meer te begrijpen over de populariteit van een film en de tweets over een film.

## 5.1 Aantal tweets per dag

Een van de eerste zaken die we kunnen bekijken i.v.m. *tweets* over een bepaalde film is hoeveel *tweets* er per dag over die film op Twitter geplaatst werden. Dit gegeven kunnen we gebruiken om te kijken hoeveel er over een bepaalde film wordt gesproken en hoe populair deze film is op een bepaalde dag. Dit zegt uiteraard niets over het feit of men al dan niet veel kwaads of goeds zegt over deze film.
Alle *tweets* bevatten steeds de datum, tijd en tijdzone waarin de *tweet* geplaatst werd. Het was dus steeds voldoende om hier slechts de datum uit te halen en alle tweets met dezelfde datum in 1 lijst te plaatsen. Zo heeft men dus een aantal lijsten dat gelijk is aan het aantal dagen waarop we minstens 1 *tweet* hebben. Als we dan van elke lijst de lengte nemen krijgen we voor elke dag het aantal *tweets* op die dag.
Vervolgens kunnen we dit plotten op een grafiek met voor elke dag het aantal geplaatste *tweets*.
Verwacht wordt dat er een piek te zien is qua aantal *tweets* tijdens de dagen rond de release dag. Dit is zo omdat men rond die tijd het meest enthousiast is omdat de film nieuw is en omdat veel mensen de film dan ook werkelijk kunnen zien en hun mening kunnen vertellen op Twitter. In figuur 1 kunt u de het aantal *tweets* per dag zien van de film *Fast and Furious 7* in de vorm van een grafiek.



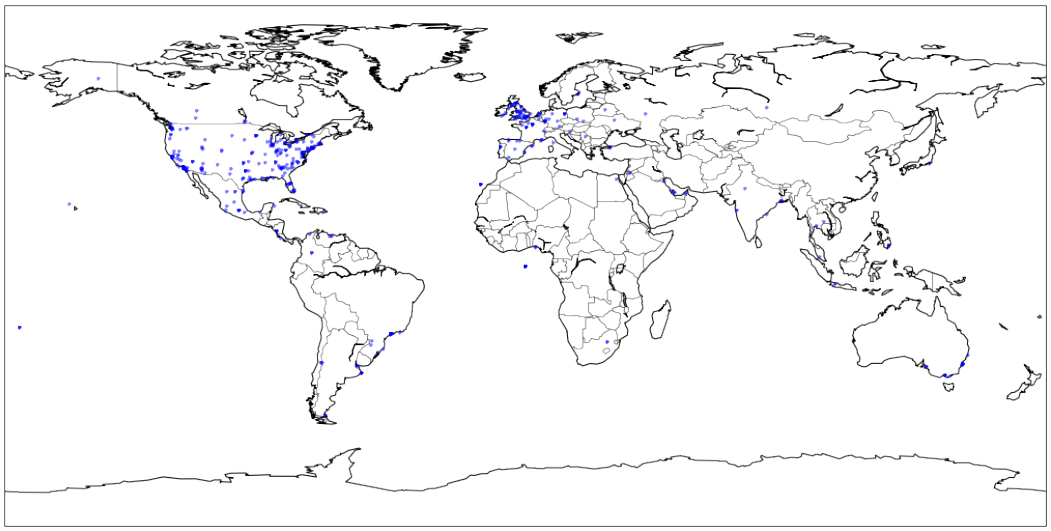

Figuur 1: Aantal tweets per dag van 1 specifieke film, namelijk *Fast and Furious 7*. Op de x-as zijn data te zien en op de y-as voor elke dag het aantal *tweets* op die dag. Op de grafiek zelf staat aangeduid wanneer de film uitgebracht werd in de Verenigde Staten van Amerika.

Zoals verwacht is het aantal *tweets* per dag het hoogste de dagen rond de

release van de film. We zien dat men voor de release steeds meer en meer *tweets* per dag plaatst en dat dit aantal langzaam daalt in de dagen na de release.

## 5.2 Tweets met geolocatie

Een ander attribuut van een *tweet* is de geolocatie. Hiermee bedoelt men de plaats van de gebruiker op het moment dat hij de *tweet* plaatst. Bij een *tweet* wordt dit voorgesteld door 2 getallen, namelijk te longitude en latitude. Deze kan men dan op een kaart plaatsen met voor elk punt op deze kaart dus een aparte *tweet*. Zo'n kaart kan men voor elke film maken. In figuur 2 is zo'n kaart te zien voor de film *Horrible Bosses 2*. Kaarten van andere films zien er gelijkaardig uit.



Figuur 2: Een wereldkaart met voor elk blauw punt een *tweet* met geolocatie. De plaats van het punt op de kaart werd bepaald door de longitude en latitude die werd teruggegeven door de *Search API* van Twitter.

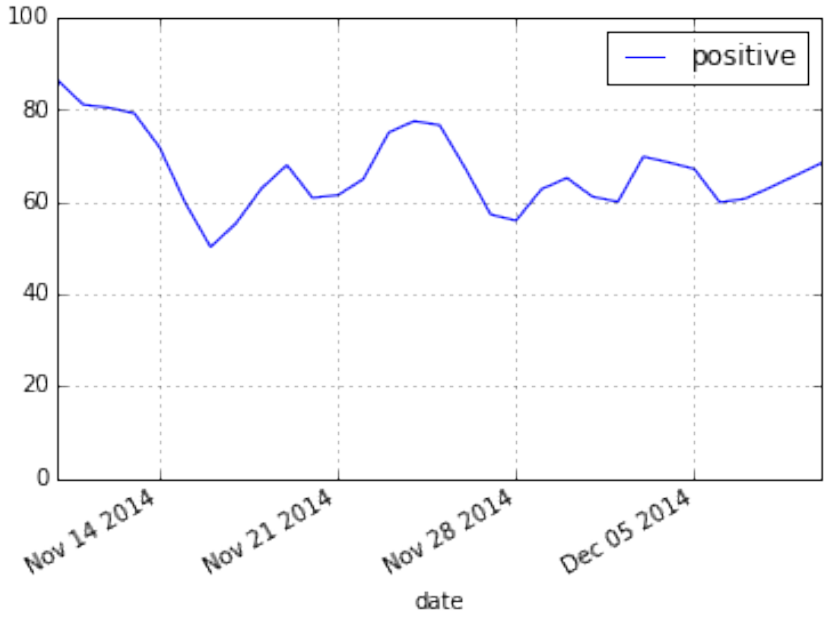
Zoals te zien is op de kaart worden *tweets* over films vooral geplaatst in Noord-Amerika en Europa. Dit is enerzijds te verklaren door het feit dat de meeste (mobiele) gebruikers die Twitter gebruiken zich in die regio's bevinden [22] en anderzijds omdat de films waar ik *tweets* van verzameld heb westerse films zijn die op die plekken dan ook het populairst zijn.

## 5.3 Sentiment analysis per dag

Met slechts 1 soort data kunnen we reeds al redelijk wat concluderen, maar het wordt uiteraard interessanter als we data gaan combineren of manipuleren. Zo kunnen we bijvoorbeeld de reeds uitgelegde *sentiment analysis* toepassen

op de tekst van de *tweets*.

Hier splitsen we de Engelstalige *tweets*, net als bij deel 5.1, op zodat we opnieuw lijsten krijgen met voor elke lijst de *tweets* van een bepaalde dag. In plaats van het aantal *tweets* in die lijsten te tellen, gaan we nu echter op elke *tweet* in elke lijst het resultaat na trainen van de *naive Bayes classifier* toepassen en aan een lijst toevoegen met geanalyseerde *tweets* van die bepaalde dag. Zo bekomt men dus voor elke dag een lijst met voor elke *tweet* het label 'pos' of 'neg' om respectievelijk aan te duiden dat de tekst van de *tweet* als zijnde positief of negatief geclassificeerd is.
Vervolgens gaat men voor elke lijst de frequentie als percentage van positief geclassificeerde *tweets* berekenen t.o.v. het totaal. Uiteindelijk heeft men dus voor elke dag een getal dat zegt wat het percentage van positieve tweets was op die dag.
Uiteindelijk kunnen we deze waarden opnieuw plotten met op de x-as de dagen waarvan we *tweets* hebben geanalyseerd en op de y-as het percentage van positieve *tweets* op die dag. Deze grafiek is te zien in figuur 3.

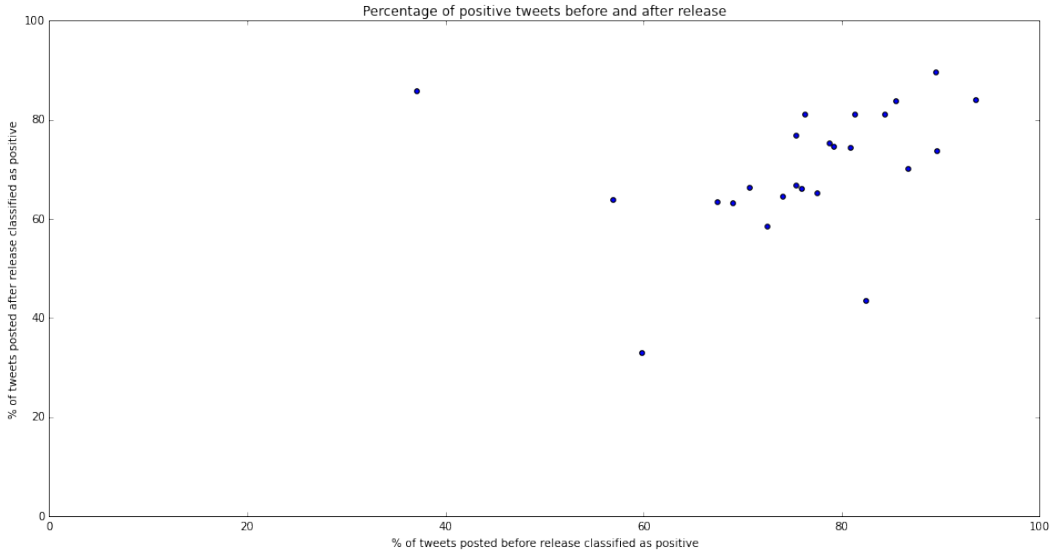



Figuur 3: Het percentage van positieve *tweets* per dag van 1 specifieke film (hier *Horrible Bosses 2*). Op de x-as ziet men de dagen waarvan men *tweets* geclassificeerd heeft en op de y-as ziet men het percentage van positieve tweets op die dag t.o.v. het geheel.

Men ziet dat men de ene dag gemiddeld al wat positiever is over de film, maar voor de rest is uit deze grafiek niet meteen iets opvallend op te merken of te concluderen.

## 5.4 Sentiment analysis voor/na release

Hier gaat men opnieuw de *tweets* van 1 film opsplitsen, maar deze keer maar in 2 lijsten: 1 voor *tweets* voor de release en 1 voor *tweets* na de release. Deze release dag is opnieuw diegene in de Verenigde Staten van Amerika.
Net zoals bij het vorige onderdeel gaat men op de *tweets* van elke lijst *sentiment analysis* toepassen en heeft men uiteindelijk als resultaat opnieuw 2 getallen: het percentage van positief geclassificeerde *tweets* die geplaatst zijn voor de release van de film en het percentage positief geclassificeerde *tweets* die geplaatst zijn na de release.
Hier gaat men in tegenstelling tot het figuur bij deel 5.3 geen lijndiagram maken, maar wel een spreidingsdiagram waarbij elk punt een film voorsteld. De waarden op de x-as stellen dan het percentage van positieve *tweets* geplaatst voor de release voor en op de y-as die na de release. Elke film heeft dus een plaats op de grafiek afhankelijk van de 2 reeds berekende percentages.
Mijn verwachting was dat men bij de meeste films even positief of negatief was na de release als toen de film nog niet uitgekomen was. Het classificeringsalgoritme was allereerst niet perfect en de veel mensen die van het internet en sociale media gebruik maken hebben op voorhand al een trailer kunnen zien van een film en weten dus al min of meer wat ze kunnen verwachten.
De grafiek is te zien in figuur 4.



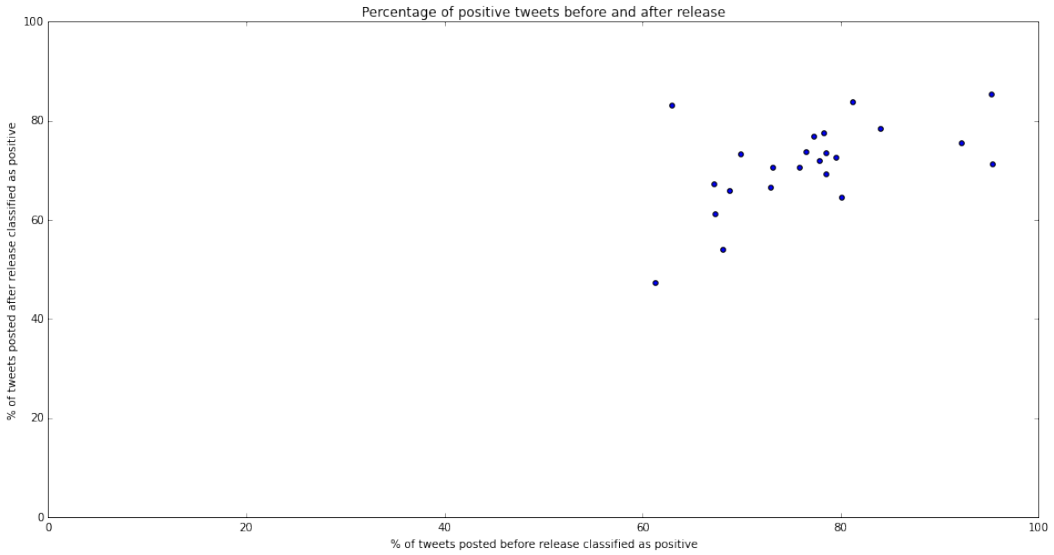

Figuur 4: Grafiek met op voor elke film een punt met als waarde op de x-as het percentage van postief gelabelde *tweets* voor de release en als waarde op de y-as het percentage positief gelabelde *tweets* na de release van de film.

We zien dat men een rechte zou kunnen tekenen die dichtbij vele punten zou liggen en dat over het algemeen het percentage van positieve tweets iets

lager is na de release van een film dan ervoor. Initieel zou men kunnen denken dat mensen voor de release hoge verwachtingen hebben maar dat deze maar gedeeltelijk ingelost worden en men nadien dus iets minder positief is over de film.
We zien echter 3 uitschieters waarbij het percentage van positieve *tweets* voor de release van de film relatief veel verschilt met het percentage van positieve *tweets* na de release. Na het bekijken van een aantal *tweets* van deze uitschieters ben ik tot de conclusie gekomen dat dit het gevolg is van *retweets* [20].
Men spreekt van een *retweet* als men de *tweet* van iemand anders opnieuw plaatst. Om dit te doen is er een knop beschikbaar in de gebruikersinterface van Twitter waardoor het vrij gemakkelijk is om de *tweet* van iemand te *retweeten*. Iedereen met een Twitter account kan een *tweet* van een andere *Twitter* gebruiker met een publiek toegankelijk profiel *retweeten*.
De tekst van een *retweet* bestaat op zijn minst uit de letters 'RT' met daarachter de tekst van de *tweet* die men retweet. De gebruiker kan zelf nog wat tekst bij de tekst van de originele *tweet* plaatsen, maar dit is niet verplicht.
*Retweets* worden ook teruggegeven door de *Search API* van Twitter. Omdat *retweets* ongeveer dezelfde tekst bevatten als de originele *tweet* worden deze ook bijna hetzelfde geclassificeerd door *sentiment analysis*. Zoals reeds besproken in deel 4.1 is de *naive Bayes classifier* die we gebruiken niet perfect en kan hij dus sommige tekst als positief classificeren terwijl deze eigelijk negatief was of omgekeerd.
Door het enthousiasme van sommige Twitter gebruikers en omdat *retweeten* zo gemakkelijk is, zijn er voor sommige films veel *retweets* aanwezig. Zo waren bij 1 film (*Fast and Furious 7*) zelfs 770744 van de 854222 *tweets* (90%) *retweets*. Bij de meeste films waren ongeveer 40% van de *tweets* retweets. Vele van de *tweets* die geretweet werden waren *tweets* die geplaatst werden door filmproducenten of andere personen of bedrijven die gerelateerd zijn aan de film. Dit wijst op het belang van sociale media. Ze zijn een belangrijk communicatiemiddel geworden voor het populair maken van een product, in dit geval een film [21].
Als er dus veel *retweets* zijn en deze dan nog eens dan nog eens fout geclassificeerd worden, heeft dit een groot effect op het uiteindelijke percentage van positief geclassificeerde *tweets*. Dit betekent dus ook dat alle punten van films in de grafiek van figuur 4 fout kunnen zijn en dus herberekend moeten worden. Het verschil tussen het percentage positief geclassificeerde *tweets* voor de release en het percentage na de release is te verklaren door het feit dat sommige *tweets* vooral voor of na de release geretweet worden. Zo wordt bijvoorbeeld de trailer van een film vooral voor de release van die film geretweet.

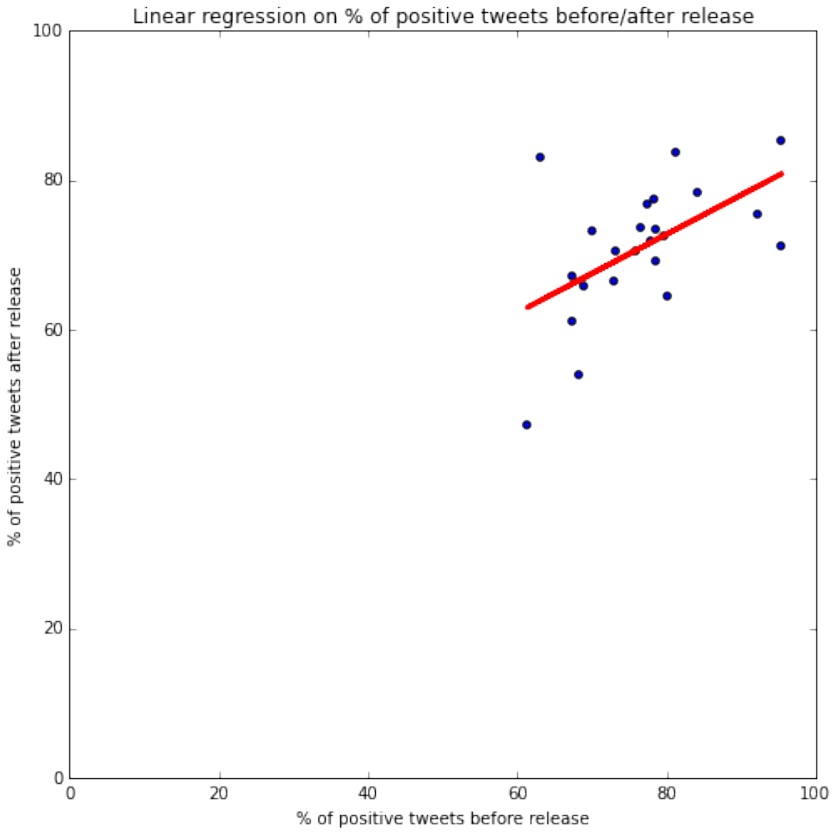
Om het zojuist uitgelegde effect tegen te gaan heb ik in de plaats gekeken naar het aantal verschillende meningen. Dit betekent dat ik van alle *retweets* slechts 1 *retweet* behouden heb, samen met eventueel de originele *tweet* als die ook verzameld werd via de *Search API*. Na het filteren van de *tweets* heb ik

de *tweets* opnieuw gesplitst, het classificeringsalgoritme opnieuw uitgevoerd, werden deze resultaten voor elke film opnieuw gecombineerd met de score van deze film op *IMDb* en werd er van deze data opnieuw een grafiek gemaakt. Een figuur van deze grafiek is te zien in figuur 5.




Figuur 5: Grafiek met voor elke film een punt met als waarde op de x-as het percentage van postief gelabelde *tweets* voor de release en als waarde op de y-as het percentage positief gelabelde *tweets* na de release van de film. Het filteren van *retweets*, zoals reeds beschreven werd, werd toegepast alvorens de grafiek werd gegenereerd.

Na het reeds beschreven filteren van *tweets* blijkt dat het verschil tussen het percentage van positieve *tweets* voor en na de release meestal niet zo groot blijkt te zijn.
We proberen nu om lineaire regressie toe te passen op de verzameling van punten. Het resultaat hiervan is een rechte waarmee men de waarde op de y-as, het percentage van positieve *tweets* na de release dus, tracht te voorspellen a.d.h.v. het percentage van positief geclassificeerde *tweets* voor de release, wat de waarde op de x-as is van een punt.
De grafiek met de rechte gevormd door lineaire regressie is te zien in figuur 6.



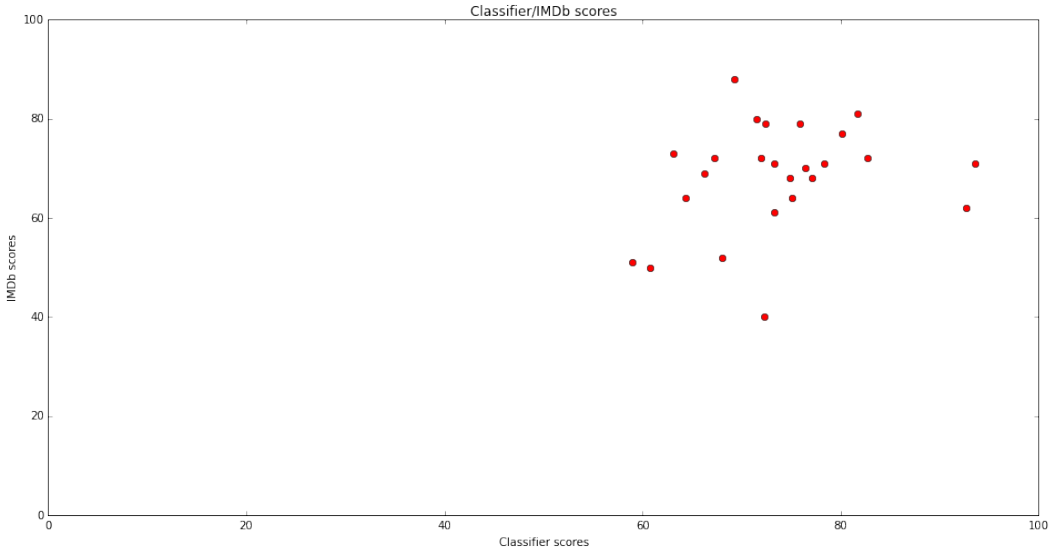

Figuur 6: Grafiek waarbij lineaire regressie is toegepast op punten die films voorstellen. Elk punt heeft als waarde op de x-as nog steeds het percentage van positief geclassificeerde *tweets* die geplaatst zijn voor de release van de film en als waarde op de y-as het percentage van na de release. Het resultaat van de lineaire regressie is op deze grafiek te zien als een rode lijn.

We zien echter dat deze lijn de waardes op de y-as, de percentages van positieve *tweets* na de release, niet heel goed kan voorspellen. Om dit nauwkeuriger te bepalen berekenen we de score zoals uitgelegd in deel 4.2. Dit had als resultaat 0.2915, terwijl het in het optimale geval 1 is. We besluiten dus dat het resultaat van lineaire regressie hier niet goed is in het voorspellen.

## 5.5 Relatie tussen sentiment en IMDb score

Tot nu toe hebben we slechts 1 bron van data gebruikt, namelijk de verzamelde *tweets*. We kunnen informatie over deze *tweets* echter combineren met andere data. Hier combineren we het percentage van positieve *tweets* van een film met zijn gebruikersscore op *IMDb*.
Voor elke film werd allereerst de gebruikersscore op *IMDb* opgezocht via de *API*. Op het moment van opzoeken had elke film reeds voldoende beoordelingen, zodat we er zeker van zijn dat we geen score krijgen die slechts gevormd werd door een kleine hoeveelheid mensen en dus niet betrouwbaar is.

Daarbuiten werden voor elke film ook de *tweets* geclassificeerd. Dit zijn alle Engelstalige *tweets* over die film die op het moment van uitvoeren in de database aanwezig waren, maar waarop ook het filteren werd toegepast, wat reeds beschreven werd in deel 5.4. Na de classificatie van alle *tweets* van een film werd er opnieuw berekend wat de frequentie is van positief geclassificeerde *tweets* t.o.v. het geheel.
Voor elke film bekomt men dus 2 getallen: de gebruikersscore van die film op *IMDb* en het percentage positief geclassificeerde *tweets* voor die film. Deze waarden kunnen we uiteraard op een grafiek tonen met net als in het vorige deel voor elke film een punt, maar hier met als waarde op de x-as het percentage positief geclassificeerde *tweets* voor een film en op de y-as zijn score op *IMDb*. Deze grafiek is te zien in figuur 7.

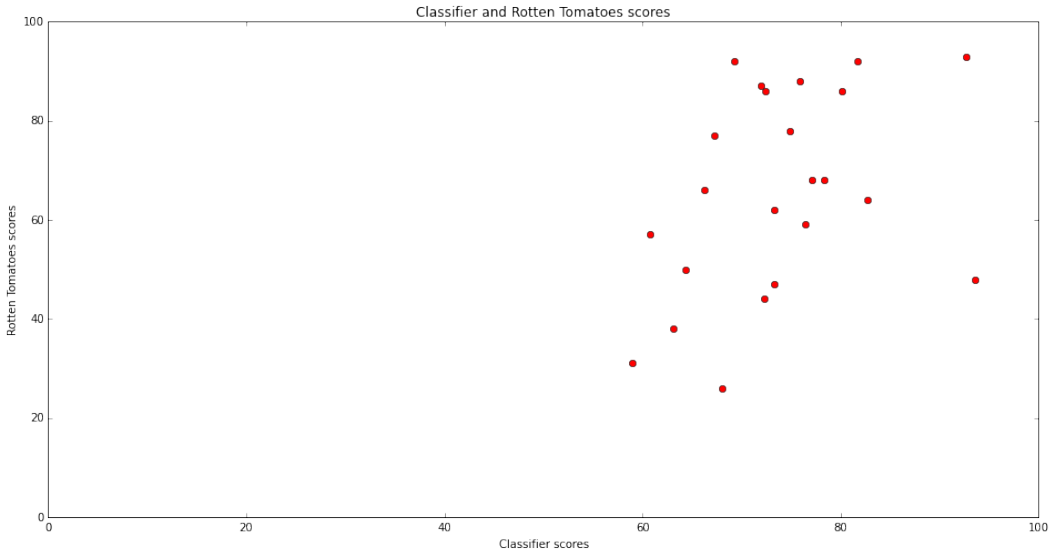



Figuur 7: Een grafiek waarop films geplot zijn als punten met als waarde op de x-as het percentage positief geclassificeerde *tweets* van die film en als waarde op de y-as de gebruikersscore op *IMDb* voor deze film.

Uit deze grafiek valt echter niet veel af te leiden. Men ziet wel dat de *classifier* scores geeft tussen de 58% end de 94%, terwijl de *IMDb* scores reeds beginnen bij 40% en eindigen bij 88%. Na een berekening van gemiddeldes van beide scores blijkt ook dat men bij de *tweets* gemiddeld positiever is dan bij de scores op *IMDb*. Bij de *classifier* was dit gemiddelde namelijk 73,68%, terwijl de gemiddelde gebruikersscore bij *IMDb* slechts 68,20% was.
Zoals intuïtief wel te zien is, is het hier niet nuttig om lineaire regressie te proberen toepassen. Er is namelijk geen verband te zien tussen een bepaalde *classifier* score en de bijhorende *IMDb* score van dezelfde film.

## 5.6 Relatie tussen sentiment en Rotten Tomatoes score

Hier gaan we hetzelfde doen als bij deel 5.5, namelijk voor elke film zijn *tweets* classificeren. Deze keer gebruiken we echter de gebruikersscore van *Rotten Tomatoes* i.p.v. die van *IMDb*. We bekomen dus opnieuw voor elke film 2 getallen: het percentage van positief geclassificeerde tweets en de gebruikersscore van de film op *Rotten Tomatoes*.
We gaan deze waarden opnieuw in een grafiek plaatsen met als waarde op de x-as opnieuw het percentage positief geclassificeerde *tweets* voor een film en op de y-as deze keer de gebruikersscore op *Rotten Tomatoes*. Deze grafiek is te zien in figuur 8.

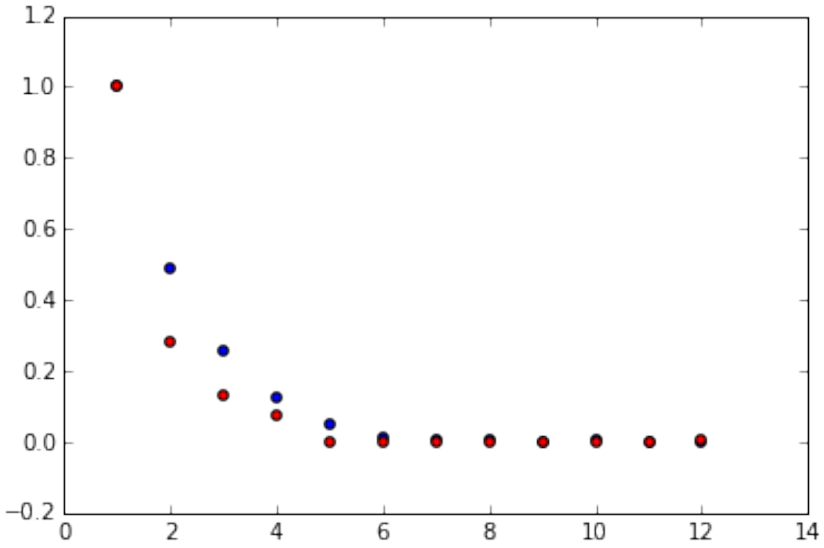



Figuur 8: Een grafiek waarop films geplot zijn als punten met als waarde op de x-as het percentage positief geclassificeerde *tweets* van die film en als waarde op de y-as de gebruikersscore op *Rotten Tomatoes* voor deze film.

Hier ziet men, net als bij figuur 7 een grafiek waar er niet direct een verband te zien is tussen de score van de *classifier* en de gebruikersscore, met deze keer de score van *Rotten Tomatoes*.
De scores van de *classifier* zijn uiteraard dezelfde als die bij deel 5.5, maar bij de scores van *Rotten Tomatoes* is echter wel een verschil op te merken. Terwijl de scores van *IMDb* van de films die ik in achting nam slechts tussen 40% en 88% lagen, lagen de scores van *Rotten Tomatoes* tussen 26% en 93%. Men kan dus stellen dat men bij *Rotten Tomatoes* iets extremer is in het geven van scores.

## 5.7 Aantal tweets per week en box office per week

Naast de gebruikersscore van een film kunnen we ook kijken naar de box office van een film om te kijken hoe populair een film is. Deze bron van data gaan we dan combineren met informatie over onze *tweets*.
Voor elke film gaan we kijken naar het aantal *tweets* die in een bepaalde week over die film geplaatst zijn en naar hoeveel deze film die week heeft opgebracht in de Verenigde Staten van Amerika. We bekomen voor elke week van elke film dus 2 getallen, indien er van die week uiteraard een box office getal is en er voor die week *tweets* in mijn database staan. Dit heeft als resultaat dat men alleen data toont van na de release, omdat er voor de release van de film uiteraard geen opbrengst is in de cinema's.
Voor elke film werd er een grafiek gemaakt waar zowel het aantal *tweets* in een bepaalde week en de box office in getoond werden. Deze waarden werden echter wel geschaald omdat de opbrengst van een film in een bepaalde week doorgaans veel groter is dan het aantal *tweets* en dit anders moeilijk te visualiseren zou zijn. Zowel het aantal *tweets* en het box office getal werden steeds geschaald t.o.v. hun waarde in de eerste week, waardoor het eerste getal voor beide waarden dan ook telkens 1 is.
Ik verwachtte dat de film kort na de release het populairst is en er bijgevolg het meest over getweet wordt en men de grootste opbrengst heeft in de cinema's. Deze populariteit zou wel stilaan dalen, om nadien constant te blijven totdat de film niet meer getoond wordt in de cinema's en dus ook geen opbrengst meer kan hebben.
We laten als voorbeeld de grafiek van de film *Beyond The Lights* zien. De grafieken van deze meeste films zagen er soortgelijk uit. De grafiek is te zien in figuur 9.



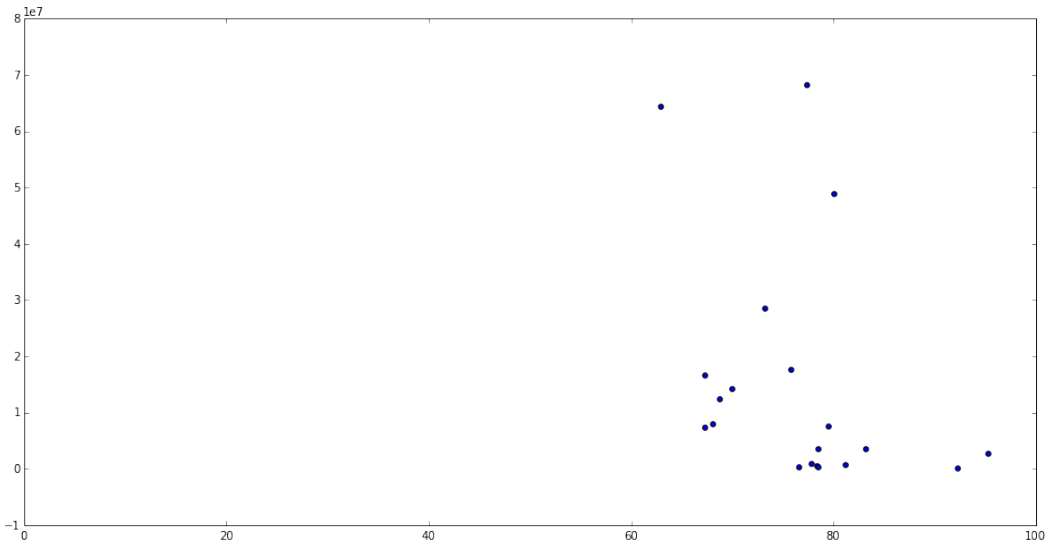

Figuur 9: Grafiek gemaakt a.d.h.v. data van de film *Beyond The Lights* met op de x-as telkens de weken van de film waarvan *tweets* en box office data beschikbaar zijn. De rode punten zijn telkens het aantal *tweets* in die week (geschaald) en de blauwe zijn telkens de opbrengst van de film in die week (ook geschaald).

Zoals verwacht is de film eerst nog populair maar krijgt de film na een aantal weken nauwelijks nog aandacht. Er is ook te zien dat het aantal *tweets* per week sneller daalt dan de opbrengst in de cinema's.

## 5.8 Relatie tussen sentiment kort voor release en box office eerste week

Voor de makers van de films en de cinema's is het meestal belangrijk dat de film succesvol is en een grote opbrengst heeft in de cinema's. Hier bekijken we of we de opbrengst in de eerste week na de release van de film kunnen voorspellen aan de hand van het percentage van positief geclassificeerde *tweets* die geplaatst zijn voor dat de film uitgebracht werd.
Allereerst werden *retweets* gefilterd zoals beschreven in deel 5.4. Hierna werd er naar de datum van elke *tweet* gekeken en werden alleen *tweets* gebruikt en dus geclassificeerd die na de release van de film geplaatst werden. Na het classificeren van elke *tweet* van een film werd, zoals bij vorige delen, opnieuw berekend wat het percentage van positief geclassificeerde *tweets* is t.o.v. het geheel.
Voor elke film werd er ook via *Box Office Mojo* gescrapet hoeveel de film heeft opgebracht in de eerste week na de release in de Verenigde Staten van Amerika. Voor elke film heeft men dan opnieuw 2 getallen. Tenslotte gaat men de punten van deze films visualiseren a.d.h.v. grafiek. Deze grafiek is te

zien in figuur 10.



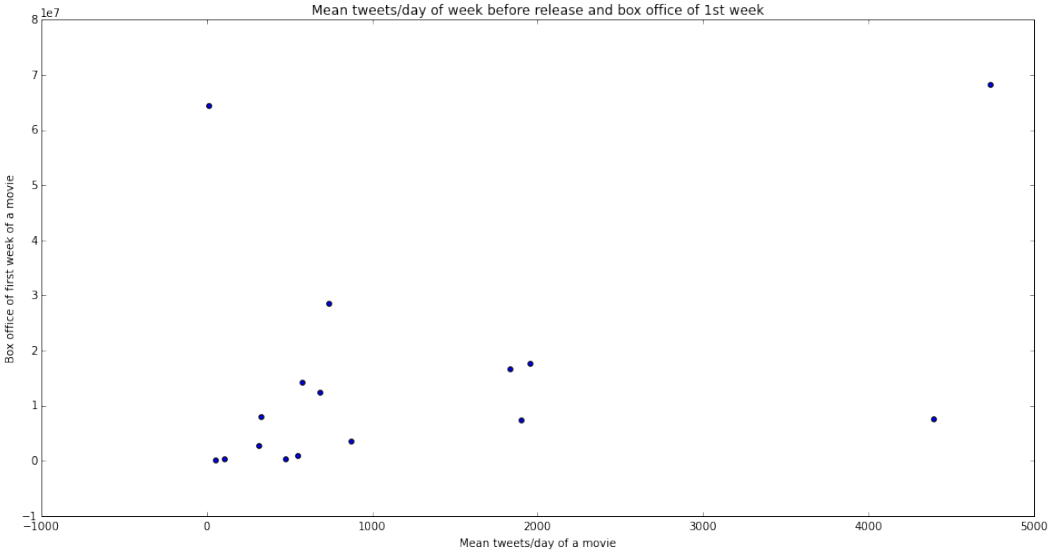

Figuur 10: Een grafiek waarbij elk punt een film voorstelt met als waarde op de x-as het percentage positief geclassificeerde *tweets* die geplaatst zijn voor de release van de film en met als waarde op de y-as de opbrengst van de film in de eerste week na de release van die film.

We zien hier dat een film niet altijd meer gaat opbrengen in de eerste week na de release als men op voorhand positiever is over de film.
Men zou hier lineaire regressie kunnen toepassen op de data, maar dit lijkt vrij zinloos omdat er intuïtief geen verband lijkt te zijn tussen het percentage positief geclassificeerde *tweets* van voor de release en de opbrengst in de cinema's in de eerste week na de release.

## 5.9 Relatie tussen aantal tweets per dag een week voor release en box office van eerste week

Omdat de poging tot het voorspellen van de opbrengst van een film in deel 5.8 niet zo succesvol was proberen we de opbrengst nu te voorspellen met behulp van het gemiddeld aantal geplaatste *tweets* per dag in de week voor de release van een film. Er werd voor elke film een lijst van 7 getallen gemaakt, voor elke dag in de week (voor de release) 1. Elk getal zegt dan hoeveel *tweets* er in die dag over die film geplaatst werden. Als men van deze lijst dan het gemiddelde neemt bekomt men het gemiddelde aantal *tweets* per dag in de week voor de release van de film.
Met de *Search API* van Twitter kan men slechts *tweets* van maximum 7 dagen terug ophalen, maar omdat ik reeds begon met het verzamelen van *tweets* van een film voordat die film gereleased werd had ik steeds *tweets* vanaf minstens

een week voor de release. Bijgevolg was dit gemiddelde wel degelijk steeds dat van 7 dagen en dus getallen.
De box office data is dezelfde als die in deel 5.8.
Voor elke film bekomt men dan opnieuw 2 getallen die men kan plaatsen in een grafiek. Die grafiek is te zien in figuur 11.



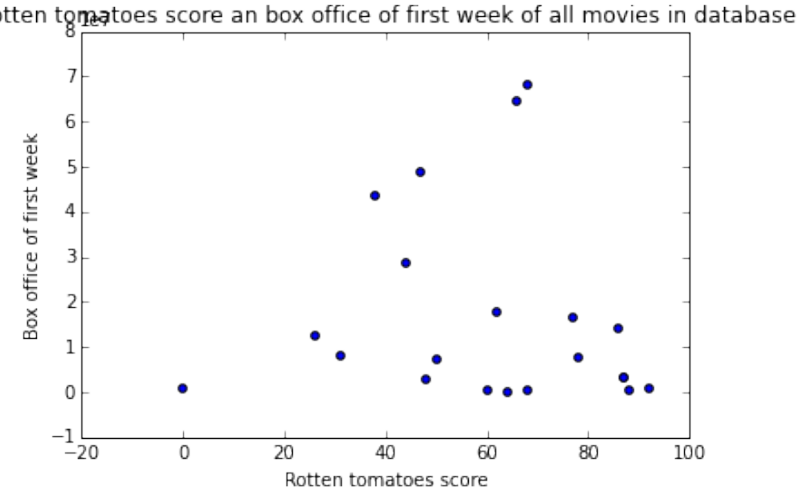

Figuur 11: Een grafiek met opnieuw voor elke film een punt. De waarde op de x-as stelt het gemiddelde aantal *tweets* per dag in de week voor de release voor en de waarde op de y-as is opnieuw de opbrengst van de film in de eerste week na de release van die film.

Ook hier is geen direct verband te merken. Er zijn films waarover niet zo veel getweet wordt in de week voor de release, maar die toch veel opbrengen. Het omgekeerde is ook te zien: sommige films zijn relatief populair op Twitter maar hebben een vrij kleine opbrengst in de cinema's.
We besluiten hieruit dat ook het gemiddeld aantal *tweets* per dag in de week voor de release van een film geen goede maatstaf is voor de opbrengst in de eerste week na de release van die film.

## 5.10 Relatie tussen Rotten Tomatoes score en de box office van de eerste week

Tenslotte gebruiken we eens niet de *tweets*, maar wel *Rotten Tomatoes* en de box office data. We gaan namelijk kijken of er een verband is tussen de gebruikersscore op *Rotten Tomatoes* van een film en de opbrengst van die film in de eerste week na de release.
Met behulp van de *Rotten Tomatoes* API en de *Box Office Mojo* scraper hebben we deze 2 getallen voor elke film verzameld en hebben we tenslotte

a.d.h.v. alle getallen een grafiek gemaakt.
Ik verwachtte dat er toch een licht verband zou zijn en dat films met een hogere score op *Rotten Tomatoes* ook een hogere opbrengst zouden hebben.
De grafiek is te zien in figuur 12.




Figuur 12: Een grafiek waarop films geplot zijn als punten met voor elke film als waarde op de x-as de gebruikersscore op *Rotten Tomatoes* voor deze film en als waarde op de y-as de opbrenst in Amerikaanse dollar van die film in de eerste week na de release.

In tegenstelling tot wat verwacht werd is er helemaal geen verband te zien en heeft het ook niet veel zin om lineaire regressie toe te passen op deze data. We zien namelijk dat films met een hogere gebruikersscore op *Rotten Tomatoes* niet per sé meer opbrengen in de eerste week na de release. Dit zou kunnen verklaard worden door het feit dat er voor sommige films meer reclame, niet alleen op het internet, gemaakt werd of dat er voor sommige films hogere verwachtingen waren en dat veel mensen die film ook wilden gaan zien. Het is bij sommige films dan ook mogelijk dat het aantal kijkers niet overeenkomt met de score die mensen gegeven hebben aan de film.

# 6 Conclusie

Bij de voorbereiding werd vooral *K-Means clustering* uitgelegd en ook hoe dit nuttig zou kunnen zijn in de bachelorproef zelf. Door de aard van mijn experimenten bleek dit algoritme echter niet nodig te zijn.
Er bleken wel andere zaken nodig te zijn om analyzes te gaan doen. Zo moest men allereerst *tweets* verzamelen van films d.m.v. te zoeken naar *tweets* met een bepaalde *hashtag* gebruik makende van de Twitter *Search API*. Omdat er geen structuur moest worden opgelegd aan de op te slagen *tweets* werd er gekozen voor een NoSQL database, en meerbepaald MongoDB, om deze op te

slaan.
Voor een deel van de metadata van films te vinden werden *IMDb* en *Rotten Tomatoes* gebruikt. Vooral de releasedatum van de film in de Verenigde Staten van Amerika en de gebruikersscore werden hiervan gebruikt. Info van beide sites was toegankelijk via een *API*. Deze info werd niet opgeslagen omdat zaken zoals de gebruikersscore nog kunnen veranderen.
Andere metadata die nodig was over films was de box office, oftwel de opbrengst van een film. Deze kan ook een maatstaf zijn voor de populariteit van een film. Voor de site waar deze info van werd gehaald, *Box Office Mojo*, was er echter geen *API* beschikbaar. Men moest daarom overgaan tot *scrapen*, waarbij men de benodigde info haalt van een *HTML*-pagina van de site zelf. Om dezelfde reden als hiervoor wordt ook deze data niet opgeslagen.

Voor het analyzeren van data werden 2 hulpmiddelen gebruikt: *sentiment analysis* en lineaire regressie.
Bij *sentiment analysis* wil men bepalen of de tekst van een *tweet* positief of negatief is. Men wil aan deze tekst dus een klasse toekennen oftwel gaan classificeren. Dit doet men via een *naive Bayes classifier*. Deze bepaalt welk label de grootste kans heeft om voor te komen, gegeven bepaalde waarden van de attributen van het te classificeren voorbeeld. De *classifier* werd eerst getraind op reeds gelabelde film reviews, om bij de experimenten dan te kunnen toepassen.
Bij lineaire regressie wil men waarden van continue aard gaan voorspellen. Men construeert een rechte waarbij men a.d.h.v. de waarde op de x-as (de input waarde) de waarde op de y-as, de outputwaarde of het label, kan aflezen. Deze rechte werd gevormd door zijn parameters aan te passen zodat ze een zo klein mogelijke fout maakte op de reeds gelabelde data.

Tenslotte kon men aan de analyzes zelf beginnen.
Allereerst is men met een aantal simpele analyzes begonnen. Bij het tonen van het aantal *tweets* per dag van een film was te zien dat dit aantal piekte op de dagen rond de release dag van de film. Voor deze dag steeg dit aantal steeds en na de release daalde het aantal geleidelijk.
Bij het plotten van de *tweets* met geolocatie van een film was dan weer te zien dat de meeste van deze *tweets* geplaatst werden in Noord-Amerika en Europa. Dit werd verklaard door het feit dat er vooral voor films is gekozen die populair zijn in de westerse wereld en omdat de meeste Twitter gebruikers zich ook in die regio bevinden.
Hierna werd *sentiment analysis* gebruikt om te kijken wat het percentage van positief geclassificeerde *tweets* is per dag. Dit aantal fluctueerde echter veel en hier was geen tendens uit op te merken.
Vervolgens werd voor elke film het percentage van positief geclassificeerde *tweets* voor de release van de film berekend en het percentage van na de release. Hiermee werd een grafiek gemaakt met voor elke film een punt. Vele

punten lagen dicht bij elkaar, maar er waren toch een aantal uitschieters. Dit kwam doordat een aantal *tweets* over een film, zoals bijvoorbeeld een *tweet* over een trailer van een film, veel geretweet werden. *Retweets* hebben dezelfde tekst als gewone *tweets* en werden door de *naive Bayes classifier* dan ook hetzelfde geclassificeerd. Als de originele *tweet* dan nog eens verkeerd geclassificeerd werd had dit een grote invloed op het uiteindelijke percentage van positieve *tweets*. Daarom werden *retweets* deels weggefilterd. Dit had uiteraard ook effect op films die eerst niet als uitschieter gezien werden. Achteraf waren er echter minder uitschieters. Op het resultaat na filteren werd lineaire regressie toegepast, maar het resultaat hiervan kon echter niet zo'n goede voorspellingen maken.
Hierna werden voor alle films de gebruikersscores op *IMDb* en *Rotten Tomatoes* maakte men voor elke site een grafiek met voor elk punt de score en het percentage positief geclassificeerde *tweets*. Bij de grafiek voor *IMDb* zag men dat de scores van de *classifier* iets hoger lagen dan die van *IMDb*. Er werd ook opgemerkt dat de *scores* van *Rotten Tomatoes* iets extemer waren dan die bij *IMDb*. Bij beide grafieken was het echter niet zinvol om lineaire regressie te proberen op de punten.
Er werd ook gekeken hoe het aantal *tweets* per week en de opbrengst van die film zich gedraagden. Men zag hier dat beide aantallen na de release van de film vrij snel daalden en nadien vrijwel constant bleven.
Iets wat ook onderzocht werd was de relatie tussen het percentage positief geclassificeerde *tweets* die geplaatst werden voor de release van een film en de opbrengst van die film in de eerste week na de release. Men zag echter dat hier geen verband bij op te merken was en dat een film dus niet meer gaat opbrengen als men er op voorhand positiever over is op Twitter.
Hierna werd er een soortgelijke relatie onderzocht, namelijk die tussen het gemiddeld aantal *tweets* per dag een week voor de release van een film en de opbrengst van die film in de eerste week na de release. Ook hier was er geen verband te zien. Er zijn namelijk films waarover veel getweet werd die weinig opbrachten en omgekeerd.
Tenslotte werd een relatie bekeken waarbij de *tweets* in de database niet gebruikt werden. Er werd namelijk gekeken of er een verband was tussen de score van een film op *Rotten Tomatoes* en de opbrengst van die film in de eerste week na de release. In tegenstelling tot wat verwacht werd was er ook hier geen verband te merken. Dit kan verklaard worden door het feit dat er voor sommige films hoge verwachtingen waren door bijvoorbeeld reclame of een goede trailer, maar dat deze verwachtingen niet ingelost werden. Er zijn echter ook films met een hoge score op *Rotten Tomatoes* maar met een lage opbrengst in de eerste week na de release.

Men kan dus besluiten dat *Twitter* meestal geen goede voorspeller is voor de populariteit van een film, waarbij men zowel de opbrengst van een film bedoelt als zijn gebruikersscore op *IMDb* en *Rotten Tomatoes*. Zelfs tussen

deze 2 manieren om populariteit te meten is geen verband op te merken. Daarbuiten merken we echter wel op dat *retweets* een belangrijke manier zijn voor het verspreiden van een boodschap en sociale media een belangrijk communicatieplatform is voor filmproducenten.

# 7 Referenties


[1] Twitter, Inc. "The Twitter API Overview"
https://dev.twitter.com/overview/api/tweets

[2] Twitter, Inc. "The Search API"
https://dev.twitter.com/rest/public/search

[3] MongoDB, Inc. "The MongoDB 3.0 Manual"
http://docs.mongodb.org/manual/

[4] MongoDB, Inc. "NOSQL DATABASES EXPLAINED"
http://www.mongodb.com/nosql-explained

[5] Twitter, Inc. "Tweets"
https://dev.twitter.com/overview/api/tweets

[6] MongoDB, Inc. "ObjectId"
http://docs.mongodb.org/manual/reference/object-id/

[7] Flixster, Inc. "Rotten Tomatoes"
http://www.rottentomatoes.com/

[8] IMDb.com, Inc. "IMDb"
http://www.imdb.com/

[9] Study.com. "Movie Critic: Job Description, Duties and Requirements"
http://study.com/articles/Movie_Critic_Job_Description_Duties_and_Requirements.html

[10] Flixster, Inc. "How do you determine whether a review with no stars is Rotten or Fresh?"
http://flixster.desk.com/customer/portal/articles/62684-how-do-you-determine-whether-a-review-with-no-stars-is-rotten-or-fresh-

[11] apathetic. "IMDb"
https://www.mashape.com/apathetic/imdb

[12] Flixster, Inc. "API Overview"
http://developer.rottentomatoes.com/docs

[13] IMDb.com, Inc. "Box Office Mojo"
http://www.boxofficemojo.com/

[14] NG, A. "Machine Learning"
Opgehaald van `https://class.coursera.org/ml-005`

[15] Weisberg, Sanford. "Applied linear regression."
Vol. 528. John Wiley & Sons, 2005.

[16] Wolfram Research, Inc. "Least Squeares Fitting"
http://mathworld.wolfram.com/LeastSquaresFitting.html

[17] Go, Alec, Lei Huang, and Richa Bhayani. "Twitter sentiment analysis."
Entropy 17 (2009).

[18] Thomas M. Mitchell. 1997. "Machine Learning"
McGraw-Hill, Inc., New York, NY, USA.

[19] NLTK Project. "nltk.corpus package"
http://www.nltk.org/api/nltk.corpus.html

[20] Twitter, Inc. "Veelgestelde vragen over Retweets (RT)"
https://support.twitter.com/articles/20169348-veelgestelde-vragen-over-retweets-rt

[21] De Vries, L., Gensler, S., & Leeflang, P. S. H. (2012). "Popularity of Brand Posts on Brand Fan Pages: An Investigation of the Effects of Social Media Marketing."
Journal of Interactive Marketing, 26(2), 8391.

[22] LEETARU, Kalev et al. "Mapping the global Twitter heartbeat: The geography of Twitter"
http://journals.uic.edu/ojs/index.php/fm/article/view/4366/3654