Microblog Sentiment Analysis for Rating TV Shows

Thesis submitted in partial fulfilment of the requirement for the degree of

Bachelor of Science In Computer Science

Under the Supervision of

Mr. Moin Mostakim

By

Ashik Abdullah (15141008)

School of Engineering & Computer Science

Department of Computer Science & Engineering

BRAC University

Declaration

This is to certify that the research work titled "Microblog Sentiment Analysis for Rating TV shows" is submitted by Ashik Abdullah to the Department of Computer Science & Engineering, BRAC University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering. I hereby declare that this thesis is based on results obtained from my own work. The materials of work found by other researchers and sources are properly acknowledged and mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted to any other University or Institute for the award of any degree or diploma. I carried our research under the supervision of Mr. Moin Mostakim.

Signature of Supervisor:

____

Moin Mostakim
Supervisor
Department of CSE,
BRAC University

Signature of author:

____

Ashik Abdullah
15141008

BRAC University

FINAL READING APPROVAL

Thesis Title: Microblog Sentiment Analysis for Rating TV Shows

Date of submission: 18th April 2017

This final report on our research is read and approved by the supervisor Mr. Moin Mostakim. Its format, citation and bibliographic style are consistent and acceptable. Its illustrative materials including figures, tables and charts are in place. The final manuscript is satisfactory and is ready for submission to the Department of Computer Science & Engineering, School of Engineering & Computer Science, BRAC University.

Signature of Supervisor:

____

Moin Mostakim

Supervisor

Department of CSE,

BRAC University

Acknowledgements

I would like to start by thanking my thesis supervisor Mr. Moin Mostakim for allowing me to work on this thesis under his supervision and for his inspiration, ideas and suggestions to improve this work. He has offered me help to understand and supported at many difficult stages of my work, starting from data collection till final approval. I am also grateful to the members of 'Twitter' for the data that I have collected from their valuable reviews.

Abstract

Sentiment analysis is being used on many fronts to extract public sentiment. It can collect data automatically from microblogging sites, such as Twitter. This user generated data can be used for various application. For example we can make product review, predict future events such as election results etc. For a TV show to gain market and to quantify its success, public opinion can be extracted to find the popularity of a particular TV show. People nowadays are writing on microblogging sites about various TV shows they are watching.

This research focuses on how data from twitter stream can provide sentiment data to rate various TV shows. The goal is to automatically extract the sentiments or opinions conveyed by users from twitter posts and then classify the post in a scale of 1 to 5, and compare them with IMDB user ratings. I used semi supervised approach to calculate the value. For classification I used Support Vector Machine (SVM). For the purpose of the calculation, I considered two types of model: multiple regression and MARS (Multivariate Adaptive Regression Splines, implemented in the earth R package), and assessed their performance using 10-fold cross-validation. For my work I choose twitter as the microblogging site as this is one of the most popular microblogging platform in the world.

Table of Content

List of Figures

List of Table

List of Abbreviation

IMDB: Internet Movie Database

MARS: Multivariate Adaptive Regression Spline

SQL: Structured Query Language

CSV: Comma Separated Value

GCV: Generalised Cross Validation

Chapter 1

Introduction

1.1 Introduction

In recent years, microblogging sites have become a very popular source for publishing huge amount of user-generated information. One of the unique characteristics of these microblogging sites is that the messages that are posted by the users are short in length and users publish their views and opinions on different topics such as politics, religion, economics, business, and entertainment. These large amount of user-generated information on the microblogging sites are utilized for many applications. Product review mining is one such application where potential consumers go through the opinions expressed by previous consumers on different sites before acquiring a particular product or service, while companies analyze the feedbacks on different products or services posted by consumers on these sites to gain knowledge about which products or services to sell more and which should be improved. These microblogging sites are also used as a source of data for making future predictions of events, such as predicting election results. Here, we are not talking about going through just one or two user messages on a particular product or service and making a decision on that. Instead, millions of messages that are posted daily on the microblogging sites need to be checked, all the relevant posts for that product or service need to be extracted, different types of user opinions need to be analyzed, and finally the user opinions and feedbacks need to be summarized into useful information. This can be a time consuming and tedious for human being . This is where sentiment analysis comes in use.

Sentiment analysis or opinion mining is the automatic extraction of opinions, emotions, and sentiments from texts. Sentiments, opinions, and emotions are subjective impressions and not facts, which are objective or neutral. Through sentiment analysis, a given text can be classified into one of the three categories - positive, negative, or neutral. Sentiment analysis of texts can be performed at different levels like - document, sentence, phrase, word, or entity level. Since our domain is restricted to microblogging sites, more specifically Twitter, as we only deal with Twitter corpus, we perform sentiment analysis at tweet level. There are various websites where we can find popularity of a particular TV show. People visits those sites and give a numeric rating to a particular episode or a particular show. But microblogging sites provides us much more data comparing to the websites. So calculating the popularity of a TV show from microblog data would make the public opinion more clear.

1.2 Motivation

Nowadays TV shows are becoming very popular across the world. For a particular person to decide which TV show he/she should watch, rating systems are a good parameter. There are certain website such as IMDB which provides rating for each episode of a TV show. These websites rating is based on the public rating. People visiting website can give a numeric rating from 0 to 10 for a particular episode. But the scenario is many people do not go to the website in order to rate each and every episode. We all know that everyone is posting their opinion in social media. The data found regarding an episode of a TV show in the social networking sites are much higher than in the websites. Moreover the data found from the website is numeric and subjective. For example, some people will rate 8 as an excellent episode and others will rate it as 9. In that case sentiment analysis of the words used to describe an episode will limit this variance.

1.3 Limitation of the Twitter Data

The data collected from twitter have some limitations. I have used twitter rest API(3) in order to collect the data from twitter. First of all, I have to select a particular episode of a TV show, then the API will give me the data on that particular topic. The problem with this process is I found a lot of data with spelling mistake. The bigger problem than the spelling mistake is the acronyms. Various kind of acronyms are used by people which are not existed in the dictionary. So I have to manually assign values to those words. Collecting the data manually by each episodes and assigning value of the acronyms were time consuming. If I could automatically differentiate the data then the work would have been easier. Another part was the accuracy, while comparing the result with the existing result I found that there was hardly any pattern. Sometime the sentiment result was following the IMDB result and sometime it was giving the exact opposite result. In this research I used scatter plot in graph to show the comparative result.

1.4 Research goal

TV shows have gained more popularity among people than any other thing in the last decade or so. People from all ages and countries are watching multiple TV shows. Now, for a person who have not watched any TV shows what will be the criteria to watch a particular TV show. Obviously he/she could find the rating and start watching it. But ratings does not always show the clear picture. They can be misleading. Moreover, how do you compare a TV show with another one. This is where the sentiment analysis can come handy.

My primary goal from this research is to find rating of a particular TV show from analysing the Twitter data and compare them with IMDB rating. This thesis will show comparison between various TV shows as well as the reaction of people about each episode. This can also be used to know about what people are thinking about any upcoming series or a season of an upcoming TV series.

Chapter 2

Literature Review

2.1 Literature review

Sentiment analysis is a growing area of Natural Language Processing with research ranging from document level classification. (Pang and Lee 2008) to learning the polarity of words and phrases (e.g., (Hatzivassiloglou and McKeown1997; Esuli and Sebastiani 2006)). Given the character limitations on tweets, classifying the sentiment of Twitter messages is most similar to sentence-level sentiment analysis(e.g., (Yu and Hatzivassiloglou 2003; Kim and Hovy2004)); Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Bermingham and Smeaton, 2010) and Pak and Paroubek (2010). Go et al. (2009) use distant learning to acquire sentiment data. They use tweets ending in positive emoticons like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM), and they report SVM outperforms other classifiers.

2.2 Methodological biases

There are high chance the data collected are biased and to be noise. Data Bias: The collected data from Twitter can be biased. As I could not collect data from all over the world.

People from different place have different views. Also some TV shows which are old and people back then did not have twitter to express their feeling; this TV shows does not have much data. So the data is not completely neutral. Signal Bias: Since we are already aware of what particular events in the past had a major impact on threating data, the algorithm could have been tweaked to reflect that. It’s relatively easy to make an algorithm pick up particular shares at the right time to make a huge change in analysis, which might perceive the algorithm being successful. Data Noise: Some TV shows are popular with some age groups and some are available to a particular region. Some topic of TV series are solely based on a particular region. So the people from other region will not be able to relate with that topic and might have a negative review. That might create data noise. It can be minimised by categorizing data by region, age group and some other variables.

Chapter 3

Sentiment Analysis

3.1 Sentiment Analysis

Sentiment analysis has several approaches and often a model is based on combination of many layer of techniques to reach the conclusion. I followed keyword analysis approach. I used the TV show name in order to find the pattern of the data. Here the data is sorted on categories then are given as input to the algorithm code to find out the graphs where the entire analysis is given. text processing approach a text is broken down to words, or a string of words and it does not focus on the context in the sense that this model incorporates a large dictionary of words that carry different level of sentiment. The words in the dictionary are then matched with the words and value of sentiment in the words are found. All the values are added up to reach the final sentiment valuation.Grouping of data is very important when it comes to sentiment analysis. Since I have done analysis on a particular TV show, therefore data sorting was a very initial step of my thesis that I had to do. The different factors of analysis like comparison between the ratings and the comparison between tv shows are done separately.

Chapter 4

Data Collection and Processing

4.1 Data

Our dataset is a collection of tweets downloaded by querying Twitter REST API v1.1 over a span from May to October 2016 . As Twitter API supports topic filtering and allows specifying the topic of the retrieved posts, the optional topic parameter in the Twitter Search URL was set to 'showName' to extract all tweets related to a Particular TV show. Eventually, I collected a total of 80,000 tweets by polling Twitter API. Then the data were divided into different text files according to their name. Every TV show I worked on have a different text file for the data.

IMDB data were downloaded using Python, specifically through a package called "IMDBPy", which allows users to directly access the IMDB database. The Python script retrieves the show title, episode title, rating, number of critics who rated that episode, and more and writes this information to a CSV file. The main limitation with this script is that users must directly look up a show's ID to be used in the script beforehand. Since many movies and shows share the same title, it's difficult for the program to select the exact show a user wants without the identification number.

Words contained in the tweets were scored based on values provided by the AFINN-111 list, which can be found through a simple Google search. The list was published in the Technical University of Denmark by Finn Årup Nielsen. Each word in the list is assigned an integer ranging between -5 to +5 based on its valence.

Here is a table showing the breakdown of the collected data:

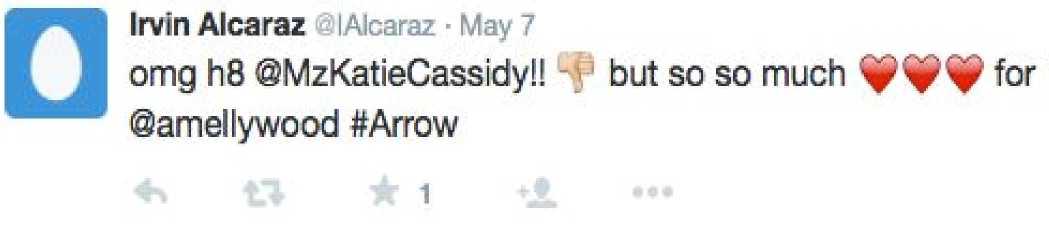
Name of TV show	Number of tweets	Number of season	Number of Episodes
Big Bang Theory	7,044	07	140
Breaking Bad	6,548	05	54
Family Guy	3,498	12	240
Friends	8,905	10	160
Glee	2,058	05	100
Game of Thrones	12,039	04	50
Greys Anatomy	4,367	10	200
How I Met Your Mother	9,518	9	180
Mad Men	2,605	6	60
Sex in the City	3,992	6	105
Simpsons	11,071	20	400
South Park	8,906	17	170
West Wing	1,295	7	140

Table 4.1 Collected data of TV shows.

The collected data were not ready to use. They needed processing to perform the sentiment analysis. The collected data need to go through several processes in order to build the sentiment data. The following section will be discussed about the data cleaning process.

4.2 Data Cleaning and Processing

Tweets are unstructured text, which make them difficult to score accurately. Although each tweet is only 140 characters, they're filled with links, acronyms, emoticons, misspelled words, slang words, and much, much more. The final cleaning process involved three parts. Initial cleaning by deleting certain words such as "&amp" and "http://". Replacing acronyms with their actual words, for example, LOL would be replaced by laugh out loud. Replacing the emoticon encoded values with their actual meaning so that a smiley face was represented by the word "smile". This step was critical for the sentiment analysis scoring process because the scoring file links only to words. The cleaning process is shown with the following example of what a tweet undergoes at each step. Figure 1 illustrates the tweet that will be used in the example. Although this tweet was made for the purpose of this example, it is representative of the challenges involved in scoring unstructured text messages.



Irvin Alcaraz @IAlcaraz · May 7

omg h8 @MzKatieCassidy!! 👎 but so so much ❤️❤️❤️ for @amellywood #Arrow

Figure 4.1. Example of how a tweet looks on twitter.com

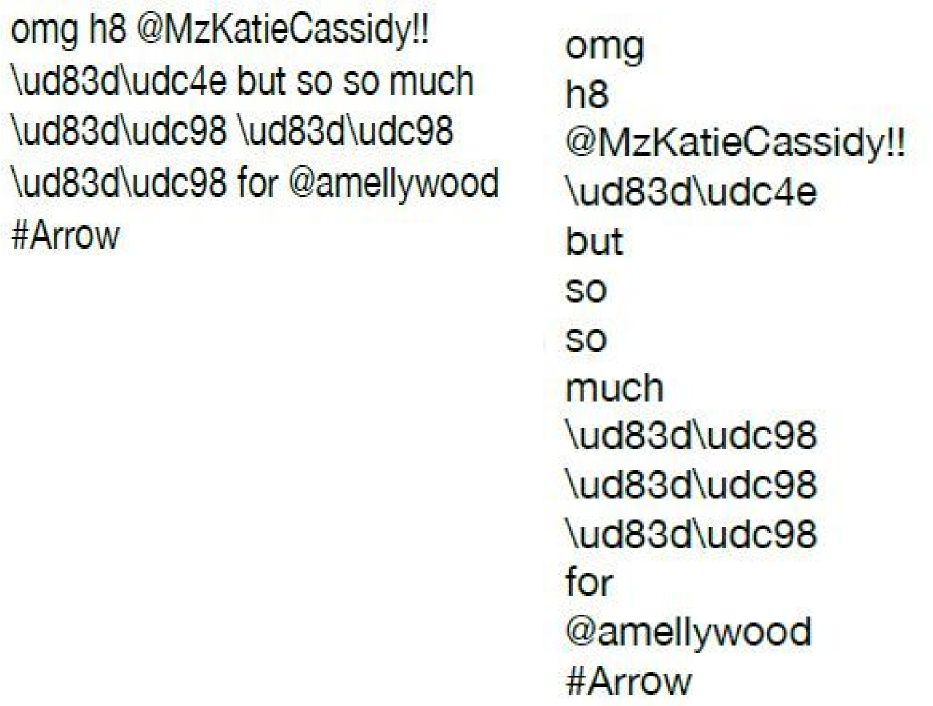
The example tweet shown in Figure 1 is how the data appear on the Twitter website. However, Figure 2 reveals how tweets appear once they're read into

```
omg h8 @MzKatieCassidy!! \ud83d\udc4e but so so much \ud83d\udc98
\ud83d\udc98 \ud83d\udc98 for @amellywood #Arrow
```

Figure 4.2. How twitter data is stored.

It may be easy to classify this tweet just by reading it, however, having the computer score it is challenging. In order for the program to score tweets as accurately as possible, code was written to remove certain words that do not have a sentiment value and to replace acronyms and emoticons with their meaningful translations. The cleaning process begins with some data manipulation before trying to delete or replace any words.

For the data cleaning process, first I have to unroll the data. Each sentence are broke into words and the emoticons and acronyms are transferred into their values.

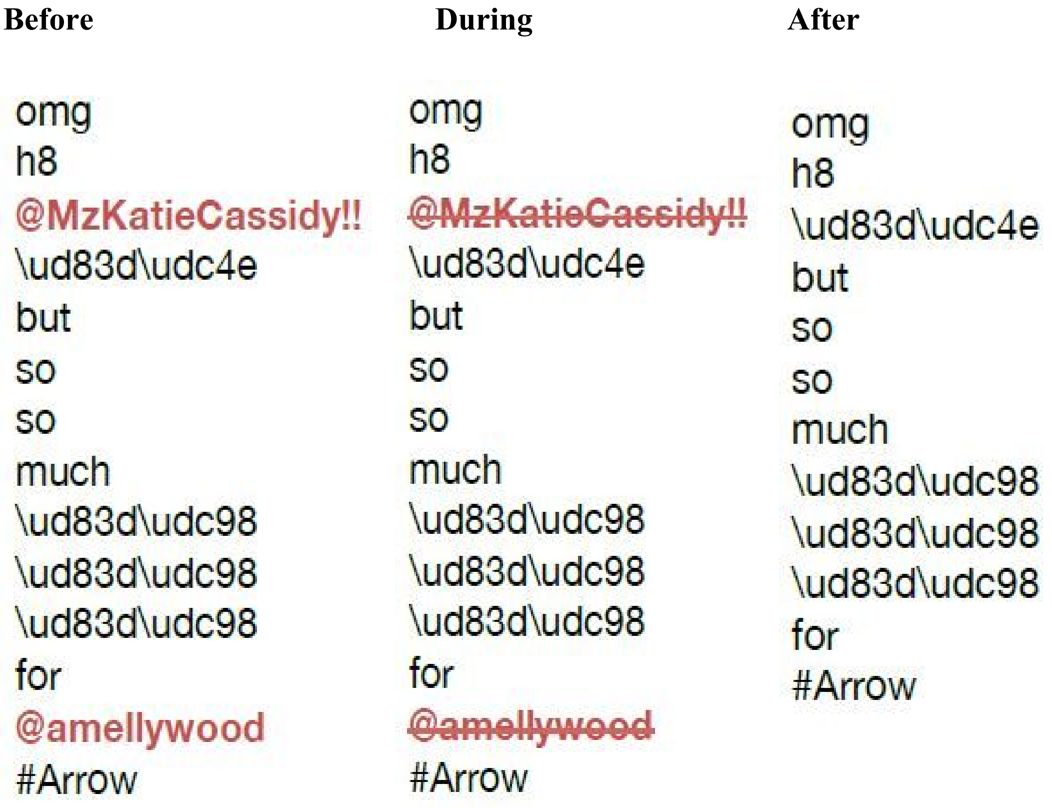


```
omg h8 @MzKatieCassidy!!
\ud83d\udc4e but so so much
\ud83d\udc98 \ud83d\udc98
\ud83d\udc98 for @amellywood
#Arrow
```

```
omg
h8
@MzKatieCassidy!!
\ud83d\udc4e
but
so
so
much
\ud83d\udc98
\ud83d\udc98
\ud83d\udc98
for
@amellywood
#Arrow
```

Figure 4.3 how tweets look after unrolling

Tweets were separated into individual words in order to make string matching easier. Instead of scanning through the entire tweet for multiple phrases to delete, the program matches the phrases directly to each word. Handling the words this way streamlines the next step. It should be noted that the actual tweet data sets consisted of many records and they had to be split into smaller, more manageable data sets to be unrolled. Data sets were processed in partitions to handle the larger datasets with over 50,000 tweets that took a long time to unroll. Partitioning the data optimized the unrolling procedure, which streamlined the initial cleaning outlined in the next step. Next step is deleting the unnecessary part. Below is a figure to show the process.



Before

omg
h8
@MzKatieCassidy!!
\ud83d\udc4e
but
so
so
much
\ud83d\udc98
\ud83d\udc98
\ud83d\udc98
for
@amellywood
#Arrow

During

omg
h8
~~@MzKatieCassidy!!~~
\ud83d\udc4e
but
so
so
much
\ud83d\udc98
\ud83d\udc98
\ud83d\udc98
for
~~@amellywood~~
#Arrow

After

omg
h8
\ud83d\udc4e
but
so
so
much
\ud83d\udc98
\ud83d\udc98
\ud83d\udc98
for
#Arrow

Figure 4.4 Initial clearing with regular expression.

Initial cleaning was performed through regular expressions within SAS. Regular expressions are basically pattern matching with strings. Similar to the example tweet, many posts often contain URL links and usernames, which do not contain any sentiment value. Perl code allows easy matching of words that start with “@”, contain “http”, and any other phrases that are not meaningful to score. Using regular expression also accommodates the variation in the usernames and web addresses by allowing SAS to look for key strings but ignore various patterns in the string.

The next part is to select unique words from a tweet and save them into a csv file.

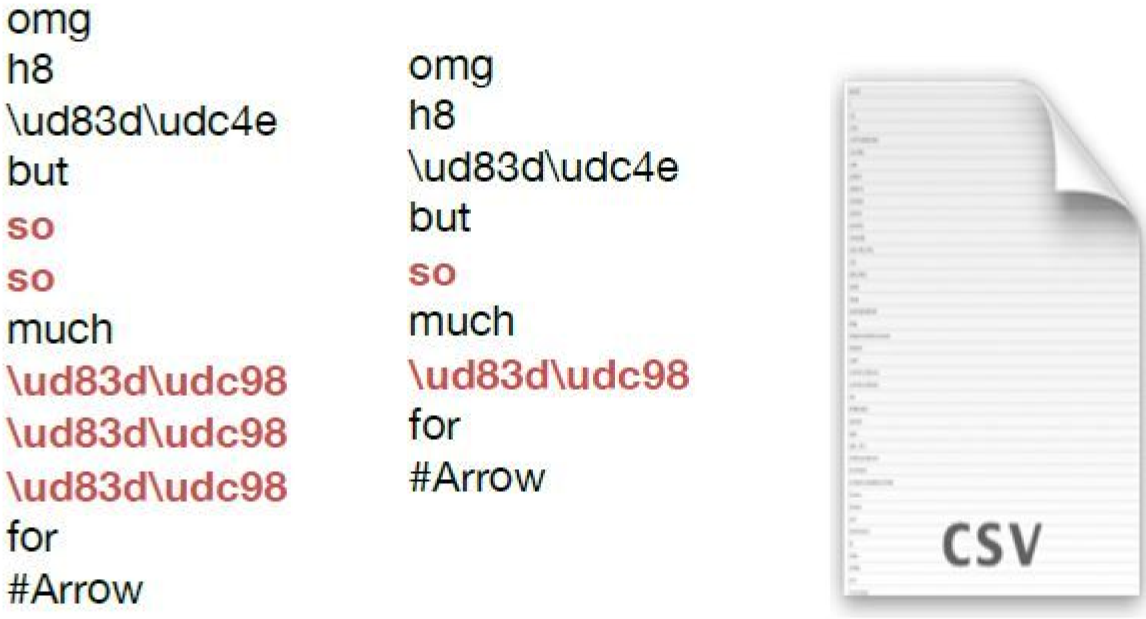
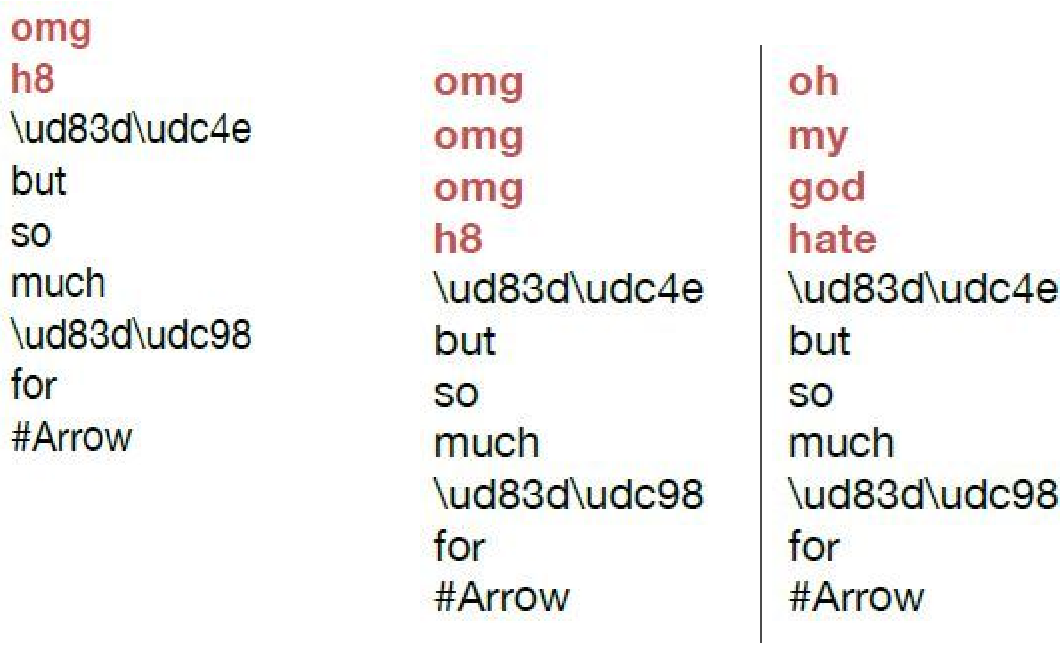


Figure 4.5 Writing unique words to csv file

Once each tweet was split into individual words, the number of observations increased considerably to the point where it was difficult to process due to lengthy processing time. Certain shows had approximately 100,000 tweets for just one episode and when unraveled, this resulted in millions of words. These episodes led to unreasonable processing times and sometimes crashed the program due to insufficient memory. To bypass the memory errors, an algorithm was developed to output distinct words to a file with a simple SQL query. Referring back to the example, Figure 5 illustrates that the text “so” and “\ud83d\udc98” would only be output once despite appearing multiple times in the original tweet. This resulted in a smaller overall word data set to be used for further cleaning and scoring.

Next is replacing acronyms with their meaning. A regular tweet contains few acronym. To perform the sentiment analysis we need to replace the acronym with the regular word.



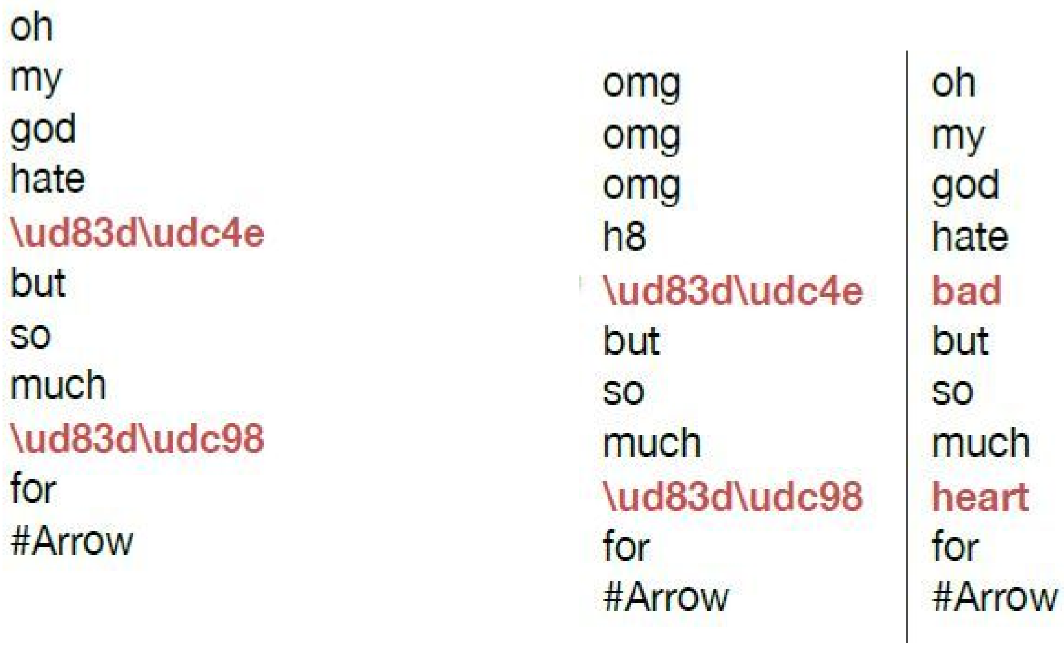
```
omg
h8
\ud83d\udc4e
but
so
much
\ud83d\udc98
for
#Arrow
```

```
omg
omg
omg
h8
\ud83d\udc4e
but
so
much
\ud83d\udc98
for
#Arrow
```

```
oh
my
god
hate
\ud83d\udc4e
but
so
much
\ud83d\udc98
for
#Arrow
```

Figure 4.6 : replacing acronyms with meaningful words

The CSV file containing unique words from all tweets was then read into Python to translate the acronyms into English words. Combining regular expressions and dictionary look up tables, Python replaced acronyms with their actual meanings. As shown above, “omg” would be replaced with three separate words “oh”, “my”, and “god”. Once all acronyms were properly translated, the next step was run to handle the emoticons.



oh
my
god
hate
\ud83d\udc4e
but
so
much
\ud83d\udc98
for
#Arrow

omg	oh
omg	my
omg	god
h8	hate
\ud83d\udc4e	bad
but	but
so	so
much	much
\ud83d\udc98	heart
for	for
#Arrow	#Arrow

Figure 4.7 Replacing emoticons with meaningful words.

Similar to the previous step, Python translated what each encoded emoticon actually meant in plain words. In this example, the “thumbs down” emoticon is translated to “bad” and the “red heart” emoticon is replaced with the word “heart”. Compared to the unstructured text from the original example tweet, the data is finally in a meaningful form that the computer can make sense of for sentiment scoring.

Chapter 5

Algorithms

5.1 Algorithms

Machine Learning approach for this thesis was chosen to be the most suitable. There are many machine learning approaches which helps us find the pattern and predict the possible outcome in future. In this thesis, we have applied multiple regression and has applied Multivariate Adaptive Regression Splines (MARS), implemented in the earth R package for the purpose of analysis. Although there are many other algorithms like decision tree ID3 and other prediction algorithms but applying linear regression helped to find pattern out of the numerical value data set. Therefore, I have used the following algorithms:

1. Multiple regression
2. MARS
3. Twitter Rest API

5.2 Multiple Regression

The following list represents all the models considered:

- Rating = ShowTitle + VoteCount + TotalScore
- Rating = ShowTtitle + VoteCount + MeanScore + SDScore
- Rating = ShowTitle + VoteCount + MeanScore

Backwards elimination stepwise regression was used to select a final model with show title, mean score, and vote count. Table 5.1 reveals that all predictors are significant in this model based on the p-values. In this model 64.16% of the variability in the IMDB ratings of shows could be explained by the model with title of the show, number of critic ratings, and mean score as the predictors.

Source	F-Value	P-Value	R-Squared
ShowTitle	4.27	0.0428	0.6461
VoteCount	13.81	<0.0001	
MeanSquare	7.06	0.0100	

Table 5.1. Results for Model With ShowTitle, VoteCount & MeanScore

5.3 MARS

MARS is a nonparametric regression that fits curved lines based on calculated splines. This model is more flexible and combines model selection with basis functions. This analysis uses the generalized cross validation (GCV) as an approximation to assess model performance. All the models fit for the multiple regression analysis were also fit using MARS, which ended up with the same final model as the multiple regression including show title, vote count, and mean score. Variable importance was calculated based on the square root of the GCV from a submodel minus the square root of the GCV from the selected model scaled to 100. The submodel is formed by removing all basis functions that have a certain variable removed. Based on variable importance, the number of critics has the largest importance, while mean sentiment score has the lowest. In other words, the contribution of the number of critics is the largest after accounting for the other variables in the model. Lastly, the R2 squared of 0.6992 is similar to the one found through multiple regression which reveals that a majority of the variability in the ratings can be explained by this model.

Functional Component	Variable Importance	R-squared
VoteCount	100.00	0.6992
ShowTitle	35.33	
MeanScore	1.82	

Table 5.2. Variable Importance for MARS Model with MeanScore, VoteCount & ShowTitle

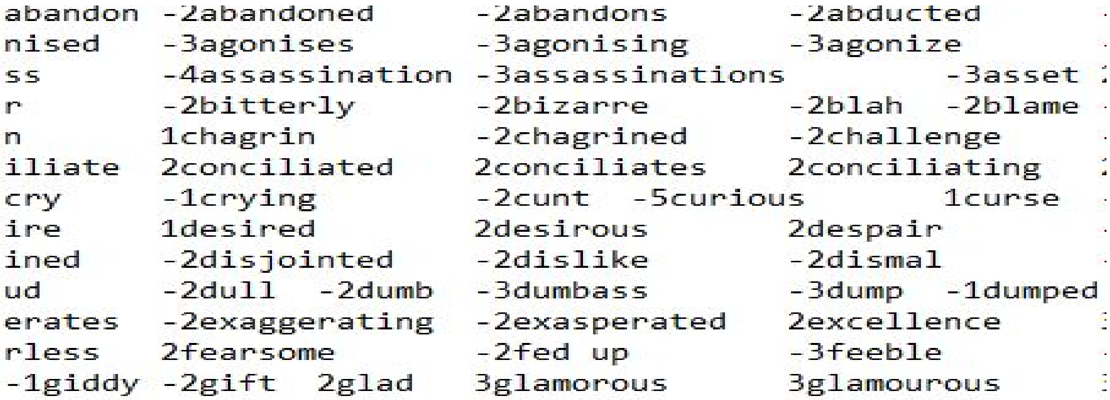
Chapter 6

Rating Calculation

6.1 Score of Words

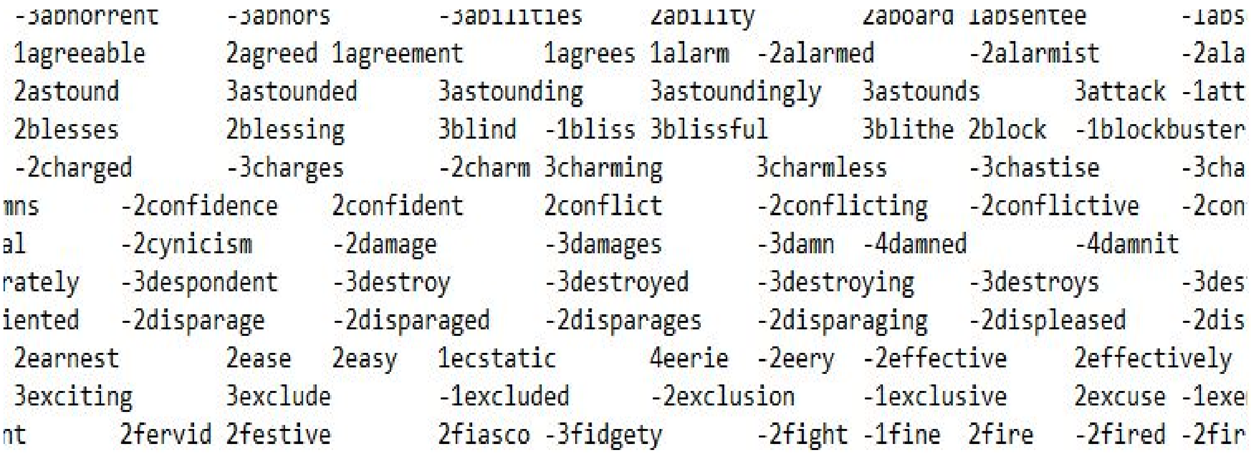
Main part of the scoring process is to convert the words into a numeric value. For this, i have used some inbuilt libraries. The libraries gave us numeric value to the word. Words contained in the tweets were scored based on values provided by the AFINN-111 list, which can be found through a simple Google search. The list was published in the Technical University of Denmark by Finn Årup Nielsen. Each word in the list is assigned an integer ranging between -5 to +5 based on its valence. Below is the screenshot showing the values got from AFINN-111

AFINN-111 provided us 2477 words assigned in integer value from -5 to +5



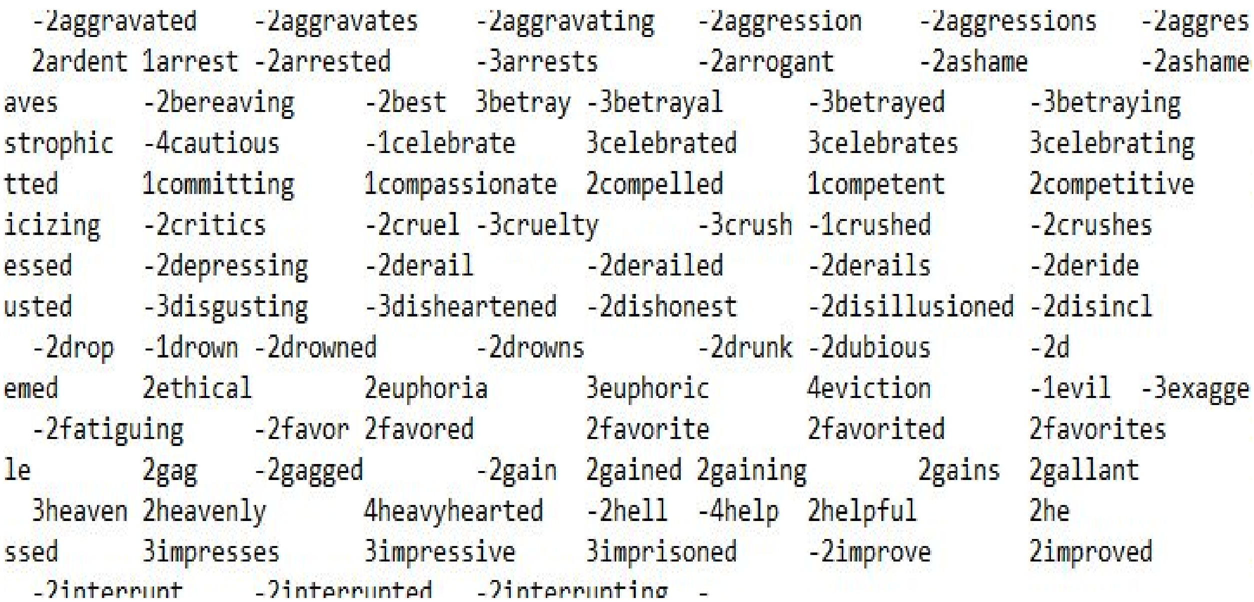
```
abandon  -2abandoned      -2abandons       -2abducted
nised    -3agonises       -3agonising      -3agonize
ss       -4assassination  -3assassinations        -3asset
r        -2bitterly       -2bizarre        -2blah   -2blame
n        1chagrin         -2chagrined      -2challenge
iliate   2conciliated     2conciliates     2conciliating
cry      -1crying         -2cunt   -5curious        1curse
ire      1desired         2desirous        2despair
ined     -2disjointed     -2dislike        -2dismal
ud       -2dull   -2dumb  -3dumbass        -3dump   -1dumped
erates   -2exaggerating   -2exasperated    2excellence
rless    2fearsome        -2fed up         -3feeble
-1giddy  -2gift   2glad   3glamorous       3glamourous
```

Figure 6.1 AFINN-111 database



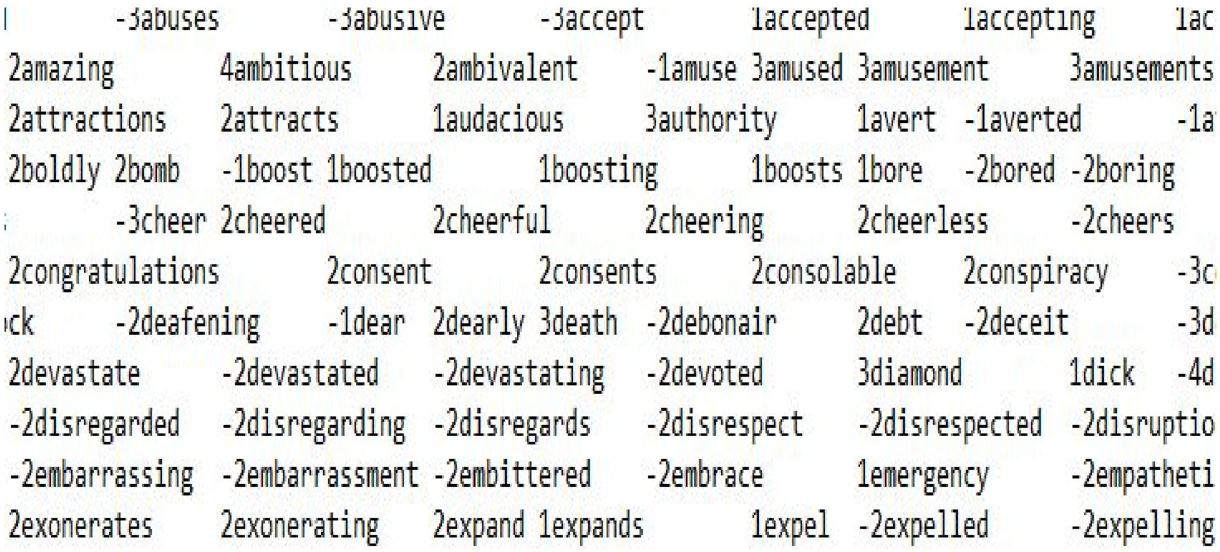
```
 -3abhorrent     -3abhors        -3abilities     2ability        2aboard 1absentee       -1abs
 1agreeable      2agreed 1agreement      1agrees 1alarm  -2alarmed       -2alarmist      -2ala
 2astound        3astounded      3astounding     3astoundingly   3astounds       3attack -1att
 2blesses        2blessing       3blind  -1bliss 3blissful       3blithe 2block  -1blockbuster
 -2charged       -3charges       -2charm 3charming       3charmless      -3chastise      -3cha
mns     -2confidence    2confident      2conflict       -2conflicting   -2conflictive   -2con
al      -2cynicism      -2damage        -3damages       -3damn  -4damned        -4damnit
rately  -3despondent    -3destroy       -3destroyed     -3destroying    -3destroys      -3des
iented  -2disparage     -2disparaged    -2disparages    -2disparaging   -2displeased    -2dis
 2earnest        2ease   2easy   1ecstatic       4eerie  -2eery  -2effective     2effectively
 3exciting       3exclude        -1excluded      -2exclusion     -1exclusive     2excuse -1exe
nt      2fervid 2festive        2fiasco -3fidgety       -2fight -1fine  2fire   -2fired -2fir
```

Figure 6.2 AFINN-111 Database



```
  -2aggravated    -2aggravates    -2aggravating   -2aggression    -2aggressions   -2aggres
  2ardent 1arrest -2arrested      -3arrests       -2arrogant      -2ashame        -2ashame
aves    -2bereaving     -2best  3betray -3betrayal      -3betrayed      -3betraying
strophic        -4cautious      -1celebrate     3celebrated     3celebrates     3celebrating
tted    1committing     1compassionate  2compelled      1competent      2competitive
icizing -2critics       -2cruel -3cruelty       -3crush -1crushed       -2crushes
essed   -2depressing    -2derail        -2derailed      -2derails       -2deride
usted   -3disgusting    -3disheartened  -2dishonest     -2disillusioned -2disincl
  -2drop  -1drown -2drowned       -2drowns        -2drunk -2dubious       -2d
emed    2ethical        2euphoria       3euphoric       4eviction       -1evil  -3exagge
  -2fatiguing     -2favor 2favored        2favorite       2favorited      2favorites
le      2gag    -2gagged        -2gain  2gained 2gaining        2gains  2gallant
  3heaven 2heavenly       4heavyhearted   -2hell  -4help  2helpful        2he
ssed    3impresses      3impressive     3imprisoned     -2improve       2improved
  -2interrupt     -2interrupted   -2interrupting  -
```

Figure 6.3 AFINN-111 Database



```
|       -3abuses       -3abusive      -3accept       1accepted      1accepting     1ac
2amazing       4ambitious     2ambivalent    -1amuse 3amused 3amusement     3amusements
2attractions   2attracts      1audacious     3authority     1avert  -1averted      -1a
2boldly 2bomb  -1boost 1boosted      1boosting      1boosts 1bore  -2bored -2boring
:       -3cheer 2cheered      2cheerful      2cheering      2cheerless     -2cheers
2congratulations      2consent       2consents      2consolable    2conspiracy    -3c
ck      -2deafening    -1dear  2dearly 3death  -2debonair     2debt  -2deceit       -3d
2devastate     -2devastated   -2devastating  -2devoted      3diamond       1dick  -4d
-2disregarded  -2disregarding -2disregards   -2disrespect   -2disrespected -2disruptio
-2embarrassing -2embarrassment -2embittered  -2embrace      1emergency     -2empatheti
2exonerates    2exonerating   2expand 1expands      1expel  -2expelled     -2expelling
```

Figure 6.4 AFINN-111 Database

6.2 Score of Hashtags

A tweet does not contain only words. It also carries hashtags and emoticons. Emoticons are converted into words but the hashtags also carries different value. To calculate the value of hashtags I have used another library. For Scoring purpose, I use the the hashtagged data set (HASH), which we compile from the Edinburgh Twitter corpus 1,

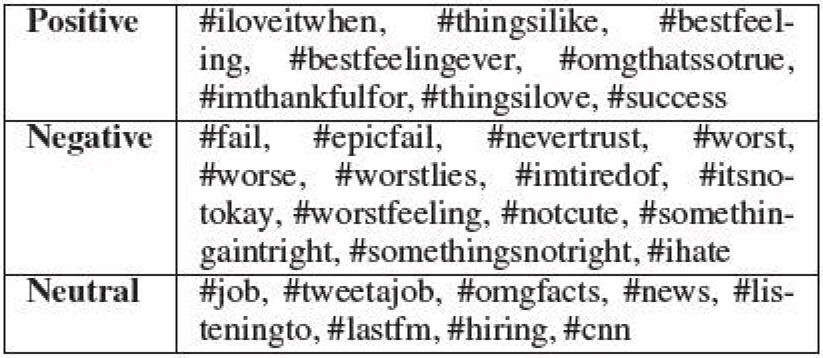
	Positive	Negative	Neutral	Total
HASH	31,861 (14%)	64,850 (29%)	125,859	222,570
EMOT	230,811 (61%)	150,570 (39%)	-	381,381
ISIEVE	1,520(38%)	200 (5%)	2,295 (57%)	4,015

Table 6.1 Corpus statistics

Hashtag	Frequency	Synonyms
#followfriday	226,530	#ff
#nowplaying	209,970	
#job	136,734	#tweetajob
#fb	106,814	#facebook
#musicmonday	78,585	#mm
#tinychat	56,376	
#tcot	42,110	
#quote	33,554	
#letsbehones	32,732	#tobehonest
t#omgfacts	30,042	
#fail	23,007	#epicfail
#factsaboutme	19,167	
#news	17,190	
#random	17,180	
#shoutout	16,446	

Table 6.2: Most frequent hashtags in the Edinburgh corpus

This hashtagged dataset has to be converted into positive and negative and neutral. Below is a tabular presentation of the positive, negative and neutral hashtags.

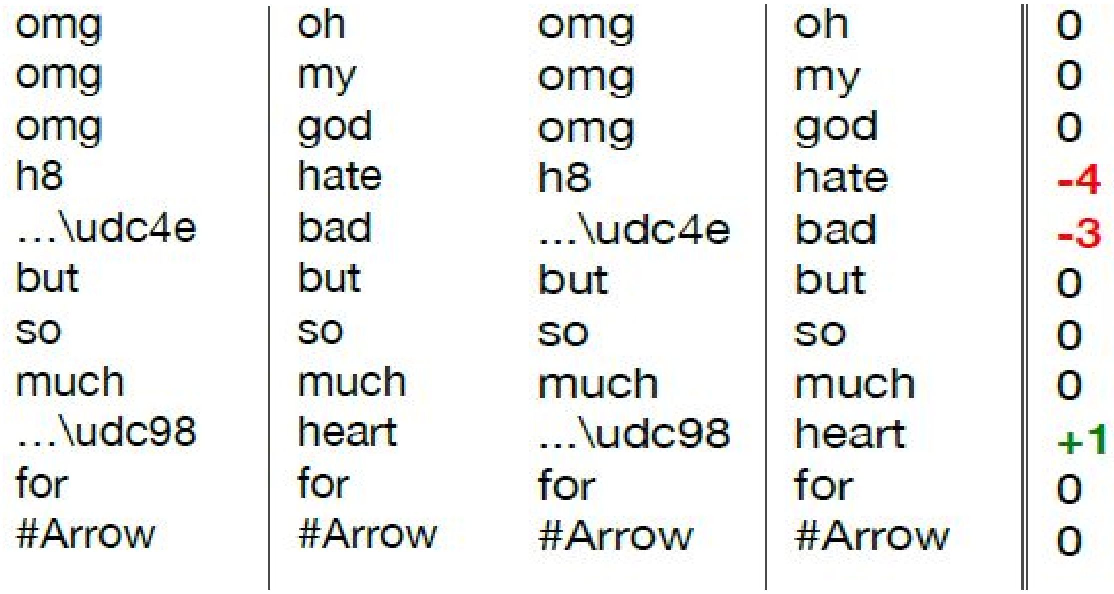


Positive	#iloveitwhen, #thingsilike, #bestfeeling, #bestfeelingever, #omgthatssotrue, #imthankfulfor, #thingsilove, #success
Negative	#fail, #epicfail, #nevertrust, #worst, #worse, #worstlies, #imtiredof, #itsnotokay, #worstfeeling, #notcute, #somethingaintright, #somethingsnotright, #ihate
Neutral	#job, #tweetajob, #omgfacts, #news, #listeningto, #lastfm, #hiring, #cnn

Figure 6.5 Positive, negative and neutral tweet.

6.3 Score Calculation :

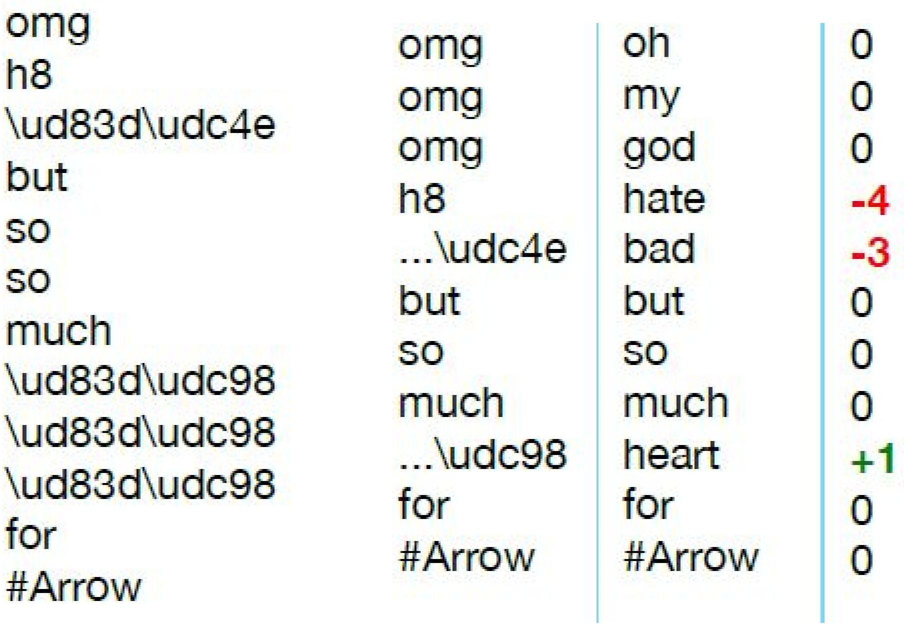
The cleaned word stored in the csv file are given score individually.



omg	oh	omg	oh	0
omg	my	omg	my	0
omg	god	omg	god	0
h8	hate	h8	hate	-4
...\udc4e	bad	...\udc4e	bad	-3
but	but	but	but	0
so	so	so	so	0
much	much	much	much	0
...\udc98	heart	...\udc98	heart	+1
for	for	for	for	0
#Arrow	#Arrow	#Arrow	#Arrow	0

Figure 6.6 Cleaned words are given value

An additional Python function was developed to score each cleaned word based on the AFINN dictionary. Once the cleaned words were assigned an integer from -5 to 5, the original word, cleaned word, and sentiment score were written to a CSV file. Words that did not match any sentiment words were assigned a value of 0. The scored data was then merged back with the original data consisting of all words unrolled as shown previously. After that I joined the uncleaned data with the clean data with score.

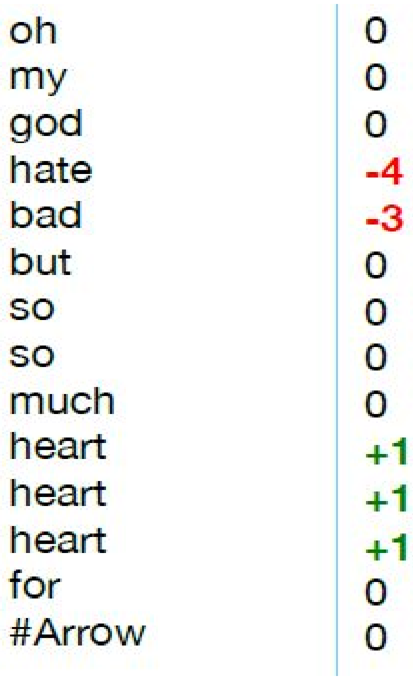


omg
h8
\ud83d\udc4e
but
so
so
much
\ud83d\udc98
\ud83d\udc98
\ud83d\udc98
for
#Arrow

omg	oh	0
omg	my	0
omg	god	0
h8	hate	-4
...\udc4e	bad	-3
but	but	0
so	so	0
much	much	0
...\udc98	heart	+1
for	for	0
#Arrow	#Arrow	0

Figure 6.7 Data before and after cleaning with score.

The next figure will show how the data looks after the whole process.

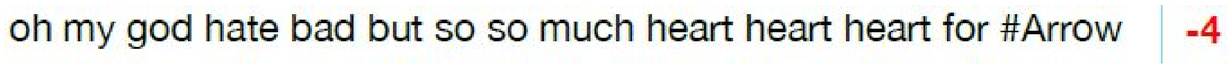


oh	0
my	0
god	0
hate	-4
bad	-3
but	0
so	0
so	0
much	0
heart	+1
heart	+1
heart	+1
for	0
#Arrow	0

Figure 6.8. Final presentation of the data.

SQL queries were used to combine the original unrolled data with the condensed scored data joined by original word and uncleaned word. Joining the data sets on the uncleaned word not only allows us to score the original data, but also replaces the messy words with the newly cleaned ones. Figure 9 demonstrates this joining process by starting with the original, uncleaned data, adding in the newly scored data, and ending with a table of just the cleaned words and their corresponding scores. To further clarify in the preceding example– while "omg" appears only once in the original data, it matches the three "omg" observations in the uncleaned column in the scored data set. Therefore, the final table will have the corresponding cleaned words "oh", "my", and "god". Similarly, although "so" appears only once in the scored data set, it will show up twice in the final table because it occurred twice in the original data. Now that the original data is scored, all that is left in the cleaning process is to roll the words back into tweets and total their scores.

After that I rerolled the individual words into one tweet and calculate the whole value of that particular tweet. And after scoring the tweets individually I took the average of all tweet regarding a particular episode to finally rate that particular episode.



oh my god hate bad but so so much heart heart heart for #Arrow	-4

Figure 6.9 Final score for a tweet.

Chapter 7

Result and Findings

7.1 Visualization

This chapter will focus on the graphical representation of the findings. After the calculation the results were plotted into a graph. For this paper I used scatter plotting. Each episode was plotted in a graph. Following figures will show the graphs.

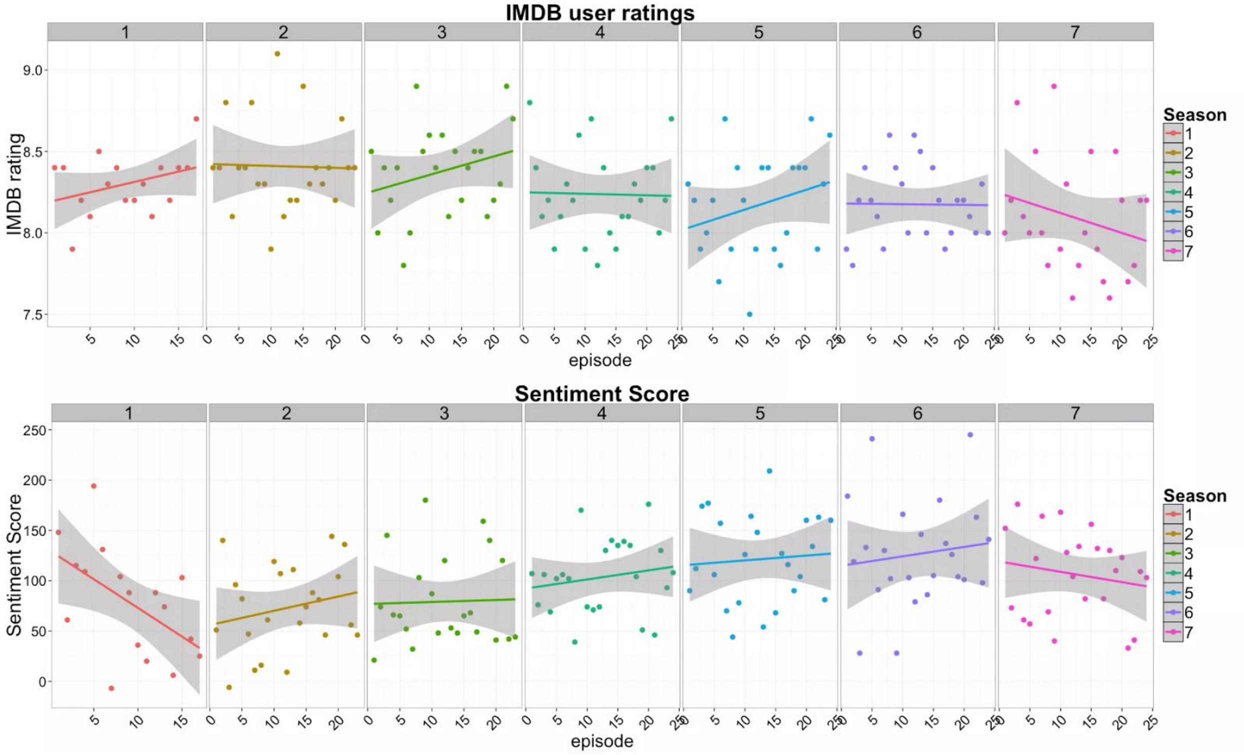


Figure 7.1 Big Bang Theory

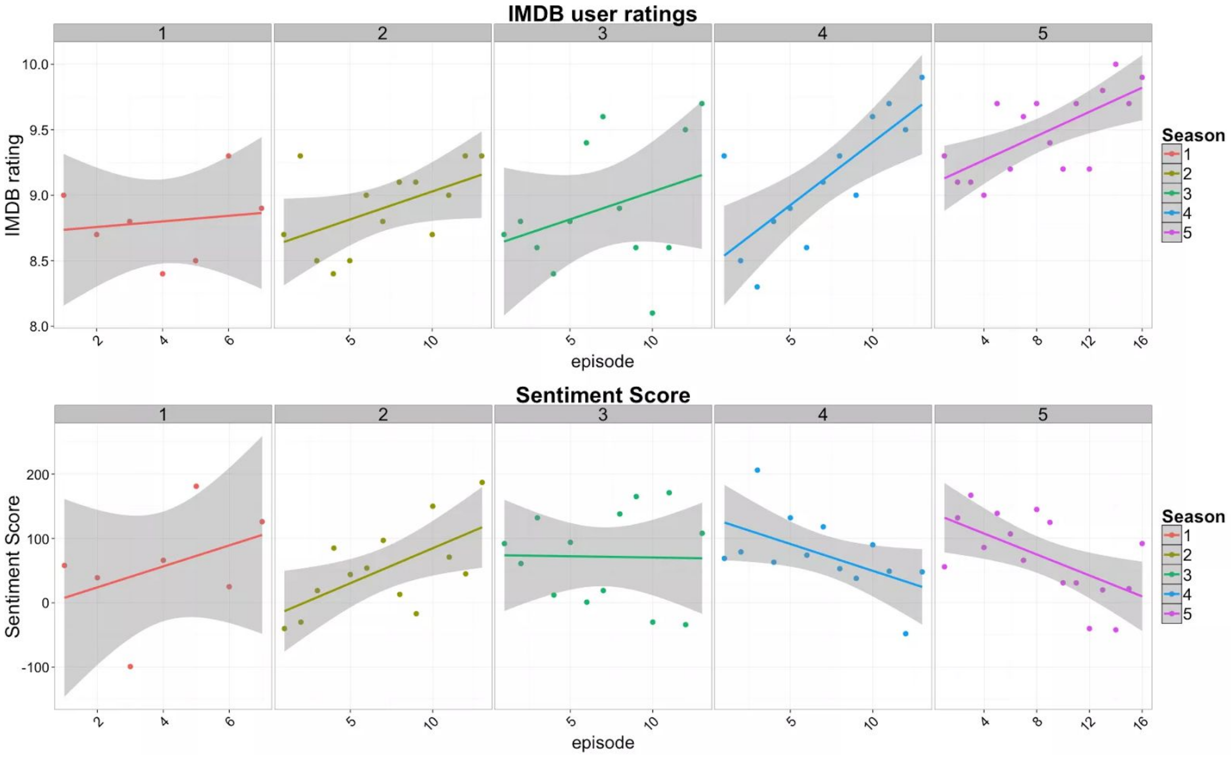


Figure 7.2 Breaking Bad

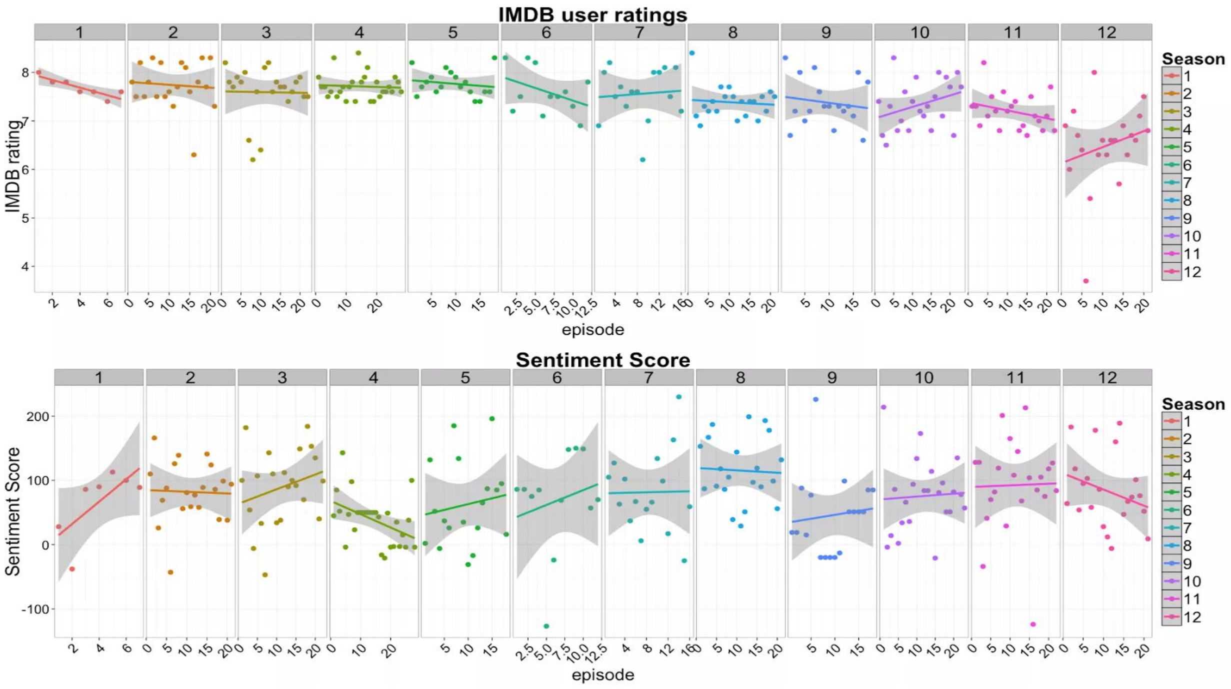


Figure 7.3 Family Guy

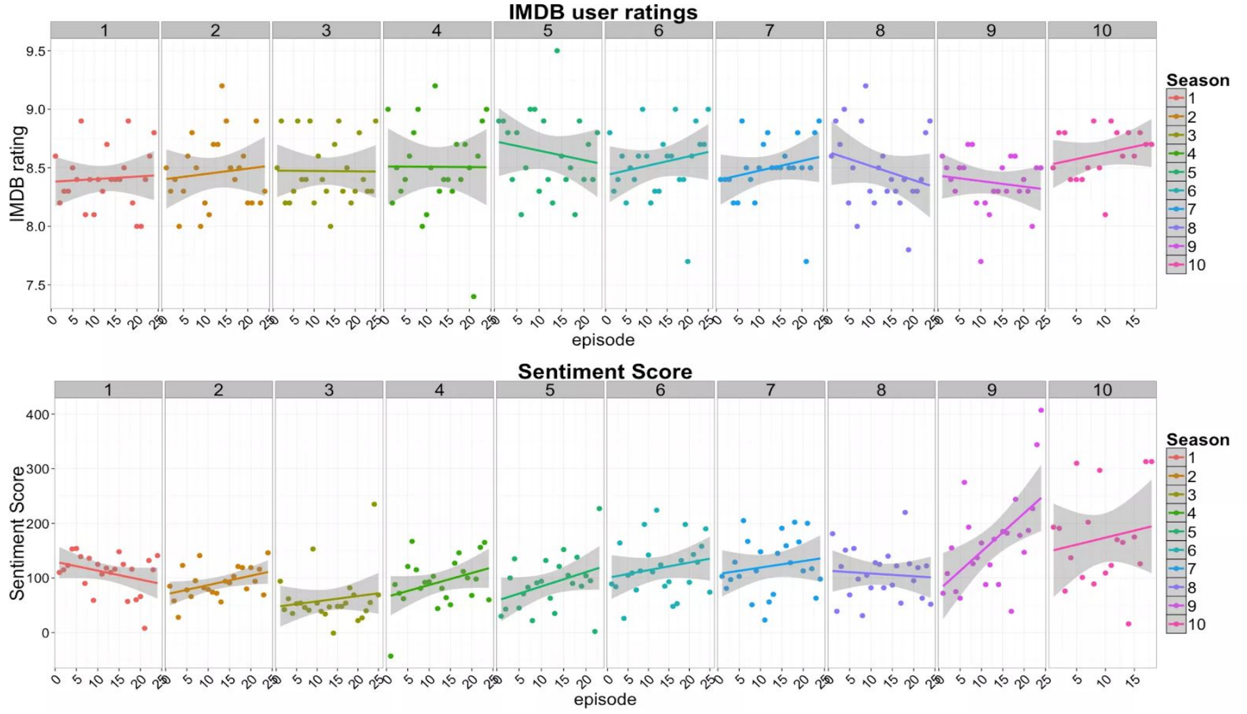


Figure 7.4 Friends

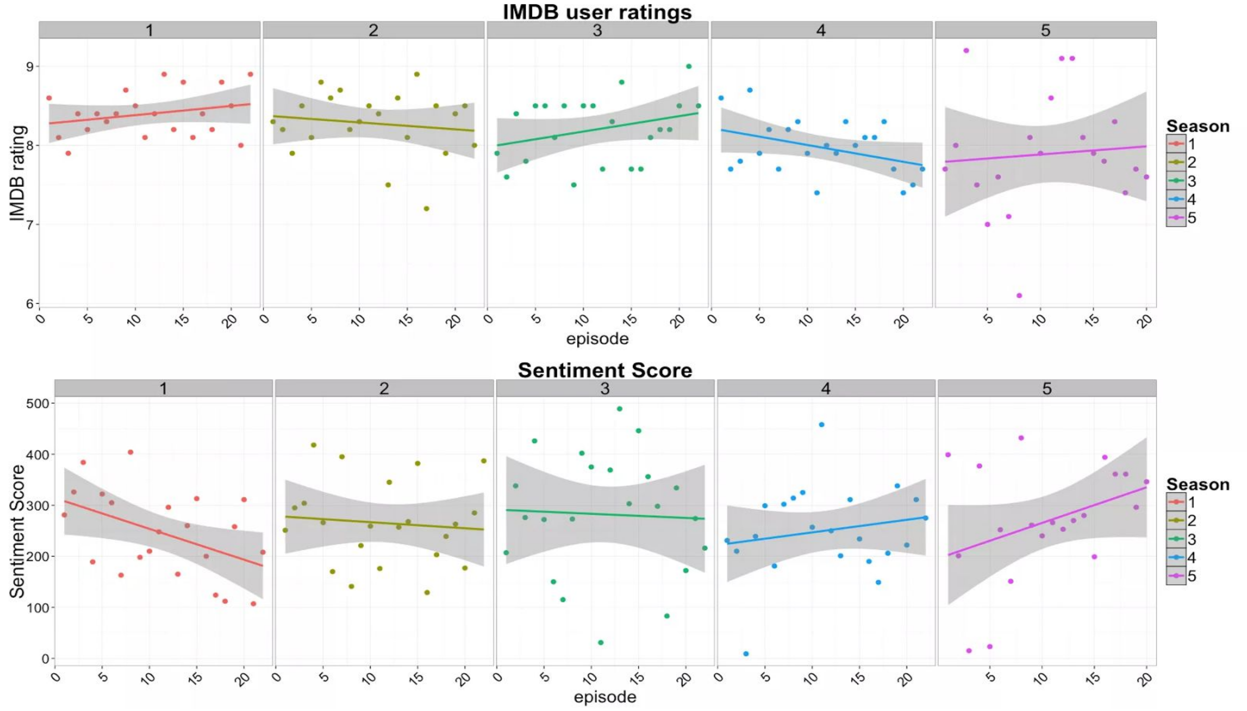


Figure 7.5 Glee

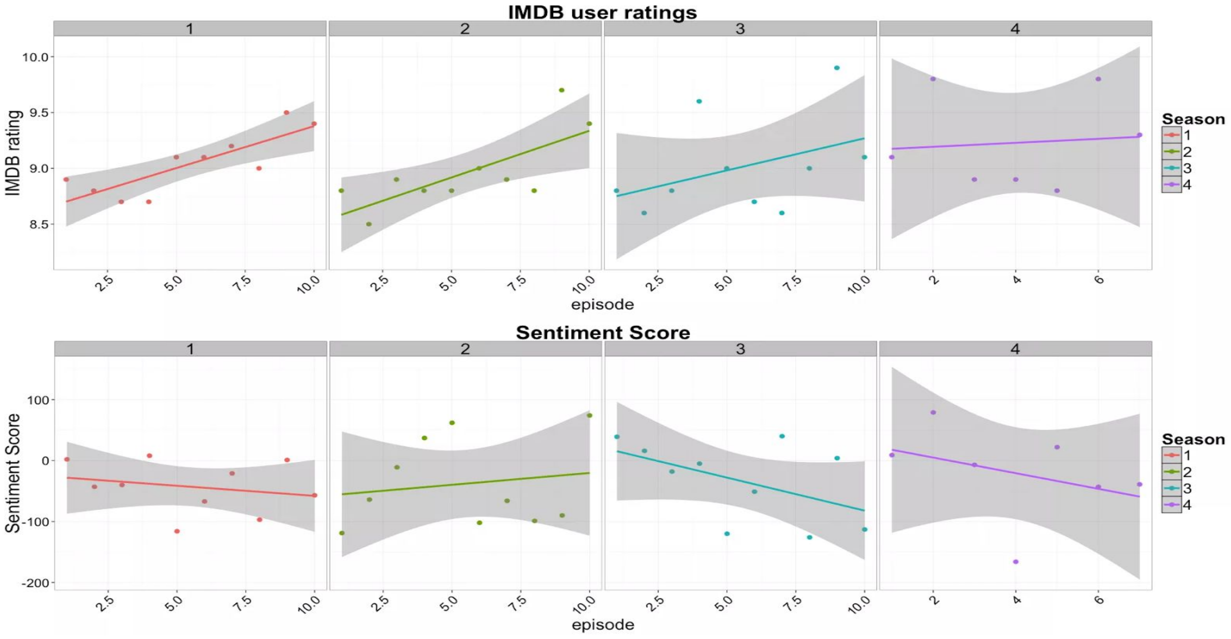


Figure 7.6 Game of Thrones

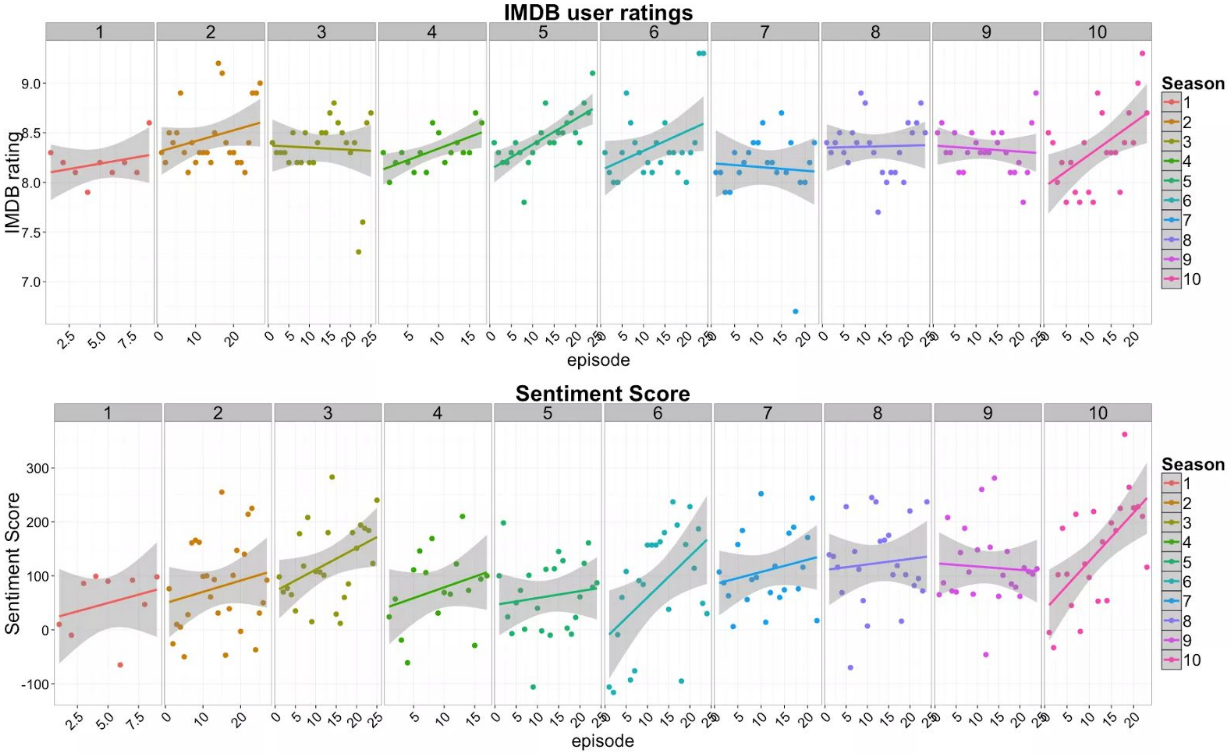


Figure 7.7 Greys Anatomy

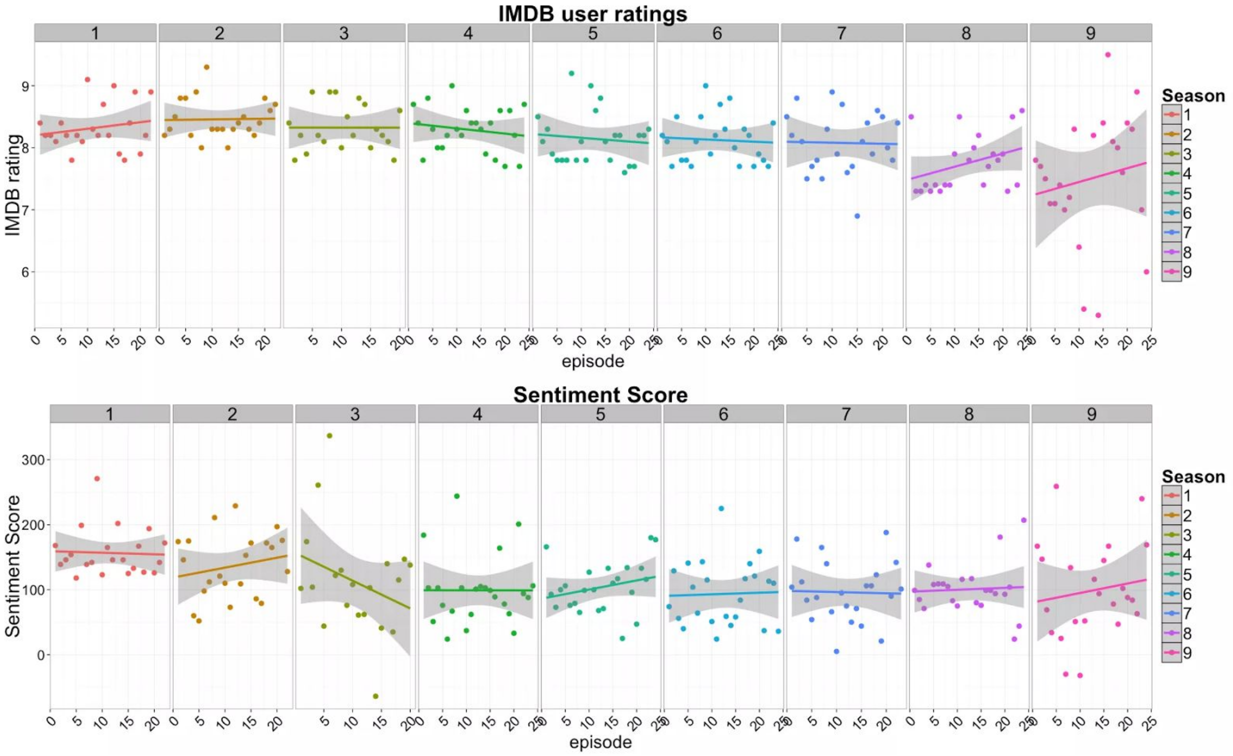


Figure 7.8 How I Met Your Mother

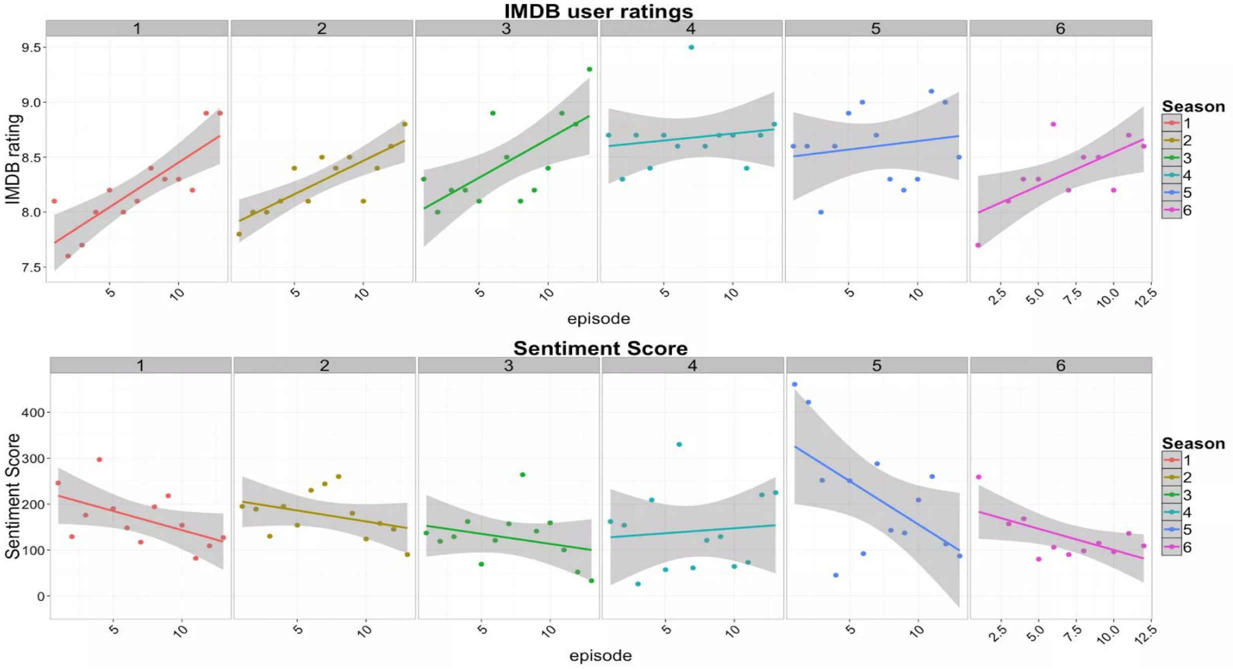


Figure 7.9 Mad Men

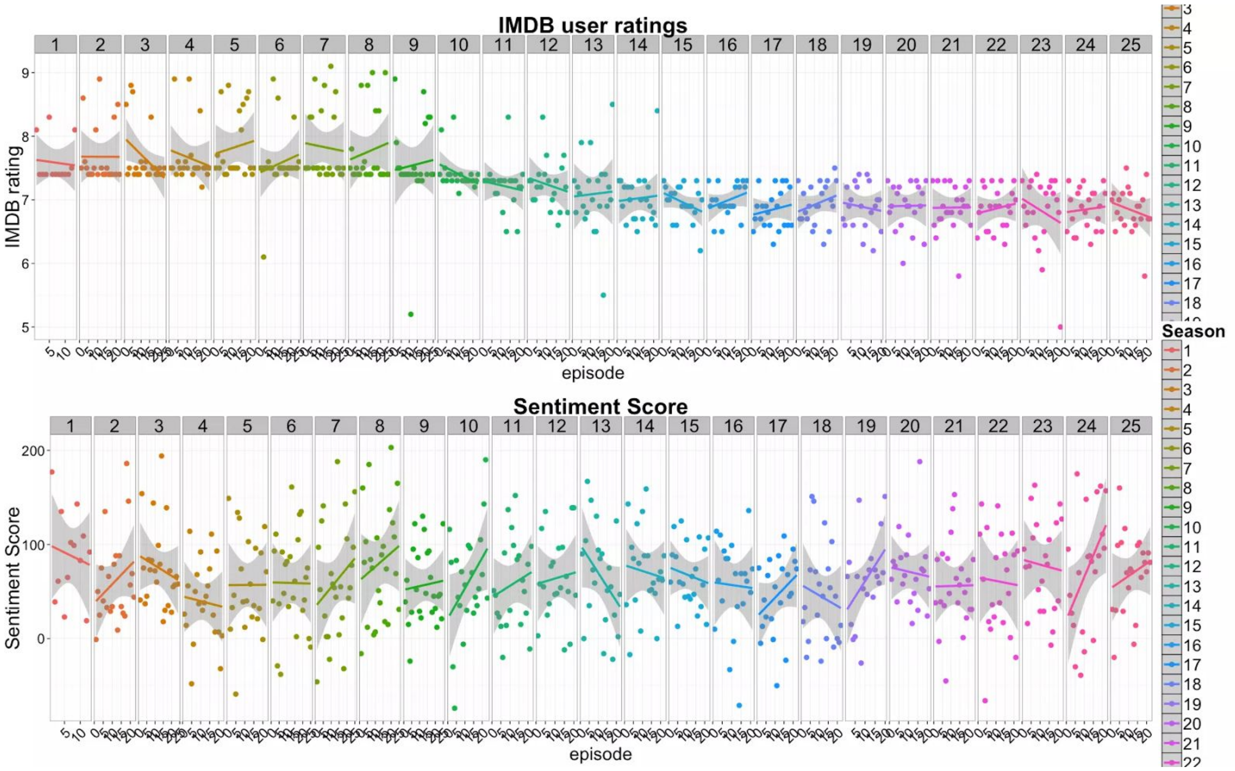


Figure 7.10 Simpsons

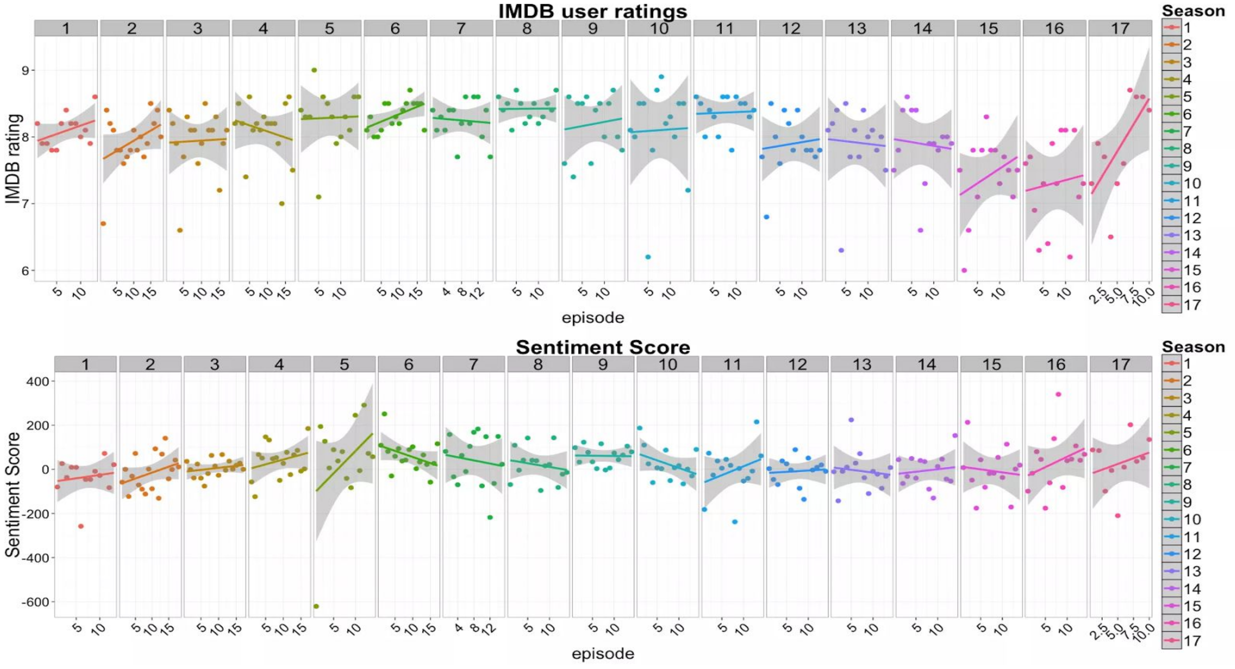


Figure 7.11 South Park

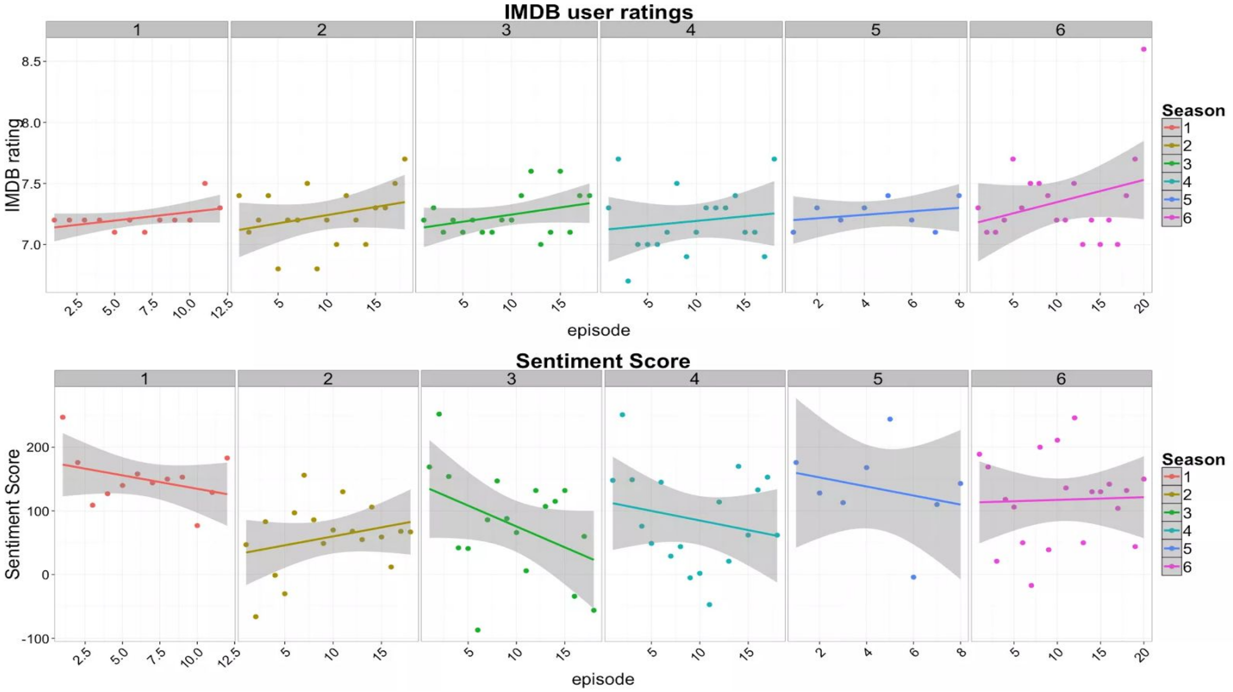


Figure 7.12 Sex in the City

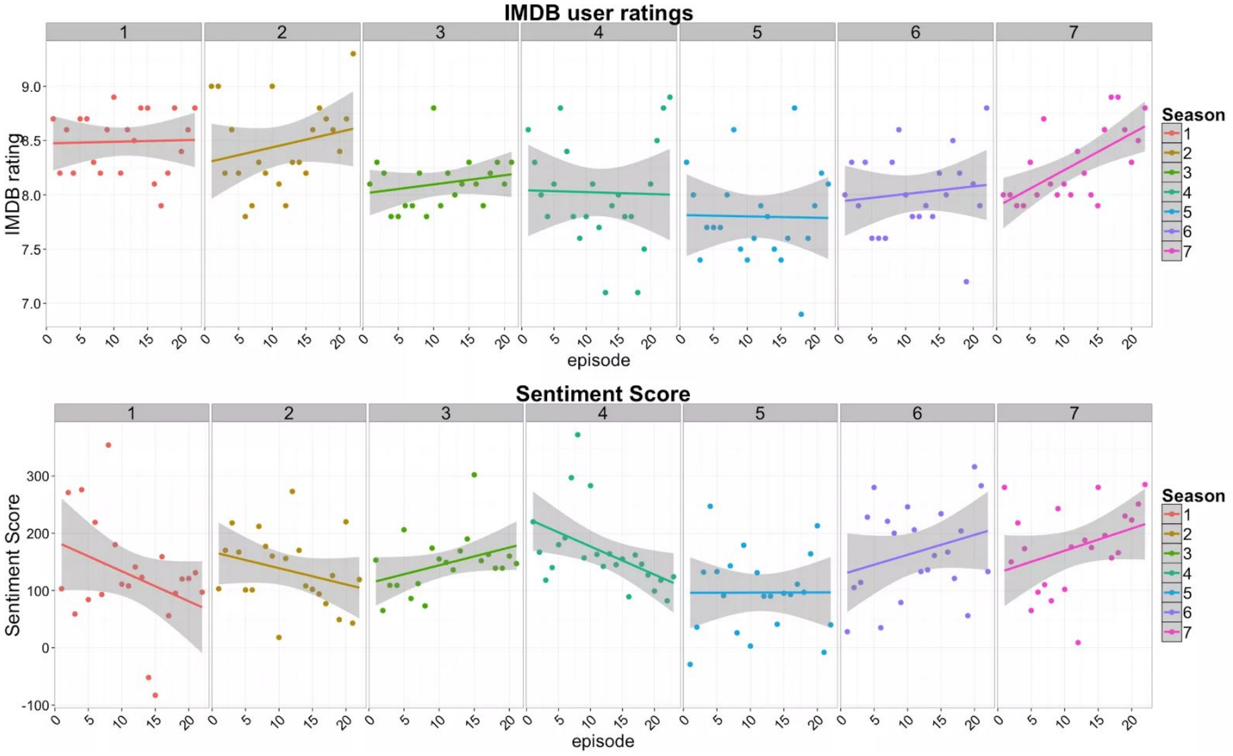


Figure 7.13 West Wing

7.2 Result & Findings :

Shows like Big Bang theory Friends and Family Guy did not experience drastic variation between the mean score and rating by episode compared to other shows. Glee, however, appears to have the largest discrepancy between mean score and rating. The ratings plummeted around episode 10, but the mean sentiment score consistently stayed around 1 to 2. The mean score appears to even increase later in the season, while ratings continued to get worse. This large discrepancy between mean score and rating for Glee is difficult to explain, however, there appears to be a logical reason for the disparity for Breaking bad and Game of Thrones. Both The Breaking Bad and Game of Thrones consistently have negative mean sentiment scores, but the ratings appear to be on the rise. Both shows had a substantial amount of tweets containing swear words compared to the other shows. For example, one Breaking Bad tweet received a score of -98 because it only contained one swear word (to the reader's imagination) repeated over and over. Unfortunately, the AFINN-111 dictionary assigns swear words the most extreme sentiment values of -4 or -5, which may not the correct context of the word given opposite connotations used by the younger generation who are also more likely to tweet. lastly, the scatterplot shown displays vote count for each episode by the show to provide some perspective about the magnitude of the number of critics. Big Bang Theory and Friends have the lowest number of critics per episode, while Breaking bad and Game of Thrones have the largest. For almost all of the shows, there appears to be one episode early in the season that has a considerably higher number of critics than the rest of the episode. It is surprising that this does not occur for the season finales of the shows.

Chapter 8

Conclusion

8.1 Conclusion and Future work:

Crowdsourcing Twitter data allows us to capture real time reactions from the online community, especially the instant feedback for television shows. These tweets represent the unfiltered, candid thoughts of users that might be more honest than an official review because they're capturing user's' natural reactions. Sentiment analysis on tweets for five different shows with two different types of statistical model was performed. Out of all possible predictor variables included in the model, mean score, show title, and vote count were the only significant predictors for rating after accounting for all other variables in the model. The sentiment analysis and models discussed in this paper only scratch the surface of what can be done with this data. Some future steps that warrant exploration include:

- investigating more cleaning methods such as stemming
- comparing multiple regression and MARS models with cross validation
- comparing tweets before, during, and after the airing of an episode
- using SAS Text Miner to form text topics
- examining the geolocation of tweets by show
- downloading more data for more shows

Appendix

Other cleaning method Considered

In addition to the cleaning process described above, various other options were explored. Ultimately, these cleaning methods were not included in the final process because of their inaccuracy. While some of these other methods were more efficient, the main cleaning process used for this project was the most accurate out of these three.

Checking for embedded sentiment word

A common problem speculated to occur was missed sentiment words due to hashtag phrases. Since many hashtag phrases do not contain spaces, it would be difficult to accurately score these phrases if they contained any sentiment words. A SAS macro to separate any sentiment words embedded in blocks of text was written so that the scoring program would be able to capture these hidden words. For example, if the word with the embedded sentiment word was: “#lovethisshow”, the macro would separate this phrase into “#love” and “thisshow”. Now the sentiment word “#love” could be properly accounted for in the scoring process. Just as a side not, the scoring code would still be able to assign the phrase “#love” a sentiment value despite the preceding “#”. This method was not implemented in the final process because the occurrence of sentiment words embedded in hashtags was surprisingly low. The problem of embedded sentiment words only occurred in less than .01% of all words in the tweets for the various episodes that were tested. In addition, when this macro was included in the code that would unroll each tweet it caused the program to run unnecessarily longer than it needed to be. Therefore, for the sake of efficiency and after discovering that the problem was not as common as previously believed, this method was not incorporated into the final process

Fuzzy String Matching

Another cleaning method that was investigated was fuzzy string matching through a Python package called “FuzzyWuzzy”. This package includes functions that will match strings based similarities based on distances, token sets, and sorts. The partial string similarity attempts to account for inconsistent length strings through what the developers call “best partial”. The site SeatGeek has an in-depth explanation, but the following example will be used for the the context of this study. The following function will compare the two strings provided and assign a value of how similar they are and can be used for identifying substrings of the given string. For example consider the code: fuzz.partial_ratio(“amazing”, “ahhmazing”) = 85 This function does a great job of capturing slang words that look similar to sentiment words but do not match exactly. The use of fuzzy string matching was not included in the final processes because it ended up over scoring words. Many of the scores greater than 100 did not match the sentiment word at all. For example, matching the word “brilliant” and “ill” resulted in a value of 100, but these words obviously have opposite sentiment score. Since so many words resulted in erroneous scores, this method was excluded because in order to maximize scoring accuracy.

Linearity:

Since the points do not appear to form any patterns in the Residual by Predicted Value plot, the linearity condition does not appear to be violated.

Normality:

Since the distribution of the residuals appears to be approximately normal in the Percent by Residual plot, the condition does not appear to be violated.

Equal Variance:

The equal variance does not appear to be violated since there is no fanning shape or pattern in the Residuals by Predicted Value plot.

Glossary

Twitter: Twitter is an online news and social networking service where users post and interact with messages, "tweets," restricted to 140 characters.

IMDB: The Internet Movie Database (abbreviated IMDb) is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews, operated by IMDb.com, Inc., a subsidiary of Amazon.

MARS: In statistics, multivariate adaptive regression splines (MARS) is a form of regression analysis introduced by Jerome H. Friedman in 1991. It is a non-parametric regression technique and can be seen as an extension of linear models that automatically models nonlinearities and interactions between variables.

Twitter rest API: The REST APIs provide programmatic access to read and write Twitter data. Create a new Tweet, read user profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are in JSON format.

AFINN-111: AFINN is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen in 2009-2011. The file is tab-separated. There are two versions: AFINN-111: Newest version with 2477 words and phrases.

References

[1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011, June). Sentiment analysis of twitter data. In Proceedings of the workshop on languages in social media (pp. 30-38). Association for Computational Linguistics.

[2] Sahayak, V., Shete, V., & Pathan, A. (2015). Sentiment Analysis on Twitter Data.

Kouloumpis, E., Wilson, T., & Moore, J. D. (2011). Twitter sentiment analysis: The good the bad and the omg!. Icwsm, 11, 538-541.

[3] https://dev.twitter.com/rest/public

[4] Yu, H., and Hatzivassiloglou, V. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Proc. Of EMNLP. 541

[5] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.Technical report, Stanford

[6] E Kouloumpis, T Wilson, J Moore. Twitter sentiment analysis: The Good the Bad and the OMG!. ICWSM, 2011.

[7] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Alan Ritter, Theresa Wilson. 2013. Sentiment Analysis in Twitter.

[8] http://nltk.org/api/nltk.metrics.html

[9] V. Hatzivassiloglou and K. McKeown, Predicting the semantic orientation of adjectives. In Proceedings of the Joint ACL/EACL Conference,2004, pp. 174–181

[10] A Kennedy, D Inkpen. 2006. Sentiment classification of movie reviews using contextual valence shifters. In Computational Intelligence, Wiley Online Library

[11] FA Nielsen. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs.

[12] Enke, D., & Thawornwong, S. (2005), The use of data mining and neural networks for forecasting stock market returns, Expert Systems with Applications, 29(4), 927-940.

[13] Thelwall, M., Buckley, K. and Paltoglou, G. (2011), Sentiment in Twitter events. J. Am. Soc. Inf. Sci., 62: 406–418. doi:10.1002/asi.21462

[14] Majhi, R., Panda, G., Sahoo, G., Dash, P. K., & Das, D. P. (2007, September). Stock market prediction of S&P 500 and DJIA using bacterial foraging optimization technique. In Evolutionary Computation, 2007. CEC 2007. IEEE Congress on (pp. 2569-2575). IEEE.

[15] Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K., & Zhang, J. (1998, October). Daily stock market forecast from textual web data. In Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on (Vol. 3, pp. 2720-2725). IEEE

[16] C. de Boor, A Practical Guide to Splines, Springer, New York (1978).

[17] Friedman, J. H. (1991). "Multivariate Adaptive Regression Splines". The Annals of Statistics. 19: 1. doi:10.1214/aos/1176347963. JSTOR 2241837. MR 1091842. Zbl 0765.62064.

[18] - M. Nash and D. Bradford. Parametric and Nonparametric Logistic Regressions for Prediction of Presence/Absence of an Amphibian. EPA Oct. 2001.

[19] Jerome H. Friedman. Fast MARS. Stanford University Department of Statistics, Technical Report 110, 1993.

[20] Regoniel, Patrick A. (November 11, 2012). Example of a Research Using Multiple Regression Analysis.In Simply Educate.Me. Retrieved from http://simplyeducate.me/2012/11/11/example-of-a-research-using-multiple-regression-analysis

[21] Gharehchopogh, F. S., & Khalifehlou, Z. A. (2012). A New Approach in Software Cost Estimation Using Regression Based Classifier. AWERProcedia Information Technology and Computer Science, Vol: 2, pp. 252-256.

[22] Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K., & Zhang, J. (1998, October). Daily stock market forecast from textual web data. In Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on (Vol. 3, pp. 2720-2725). IEEE

[24] S.M. Kim and E. Hovy. Identifying and analyzing judgment opinions. In Proceedings of the Human Language Technology Conference - North American chapter of the Association for Computational Linguistics, New York City, NY, 2006.

[25] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79-86, 2002.

[26] Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. Proceedings of the 23rd International Conference on Computational Linguistics: Posters, pages 36–44.

[27] Turney, P. D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL ’02), pp. 417–24. Stroudsburg, PA: Association for Computational Linguistics.

[28] Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. 2011. Combining lexicon-based and learning-based methods for Twitter sentiment analysis. Technical Report HPL-2011-89.

[29] A Bermingham, AF Smeaton. Classifying sentiment in microblogs: is brevity an advantage?. CIKM, 2010.

[30] Jacob Perkins. 2010. Python Text Processing with NLTK 2.0 Cookbook.

[31]Eugenio Martínezcámara, M. Teresa Martínvaldivia,L. Alfonso Ureñalópez and A Rturo Montejoráez. Sentiment analysis in Twitter. Natural Language Engineering.

[32] Rada Mihalcea. 2004. Co-training and Self-training for Word Sense Disambiguation.