Exploratory Analysis and Predicting Ratings for TED Talks

Richard Paterson and Jon Kaplan
IST 707

Table of Contents

Introduction

Knowledge and ideas are the foundation that drive society and all of its technological advances and progress. With every technological breakthrough or disruptive product, it often begins as a simple idea. Every individual is born with an impressionable brain that is ready for storage. This brain accumulates experiences and ideas to form an ideology and perspective of the world. Individuals acquire these experiences from interacting with their caretakers and through more formal education such as traditional schooling. Once a person has fulfilled educational requirements, though, there can be a gaping hole for further education and learning. In school, children are taught by qualified teachers who are knowledgeable in their expertise and vetted by professionals. During school years, online resources are experienced but mostly for social media purposes as well as informational resources. Once out of school, though, the acquisition of knowledge is a free-for-all where when one has a need for verified and trustworthy content, especially online. That is where TED talks come in. TED talks celebrate the union of public speaking and the sharing of ideas and knowledge by qualified experts in their respective fields.

TED talks provide an outlet for experts in their domain to form an 18 minutes or less speech about a thought-provoking topic. Since its inception in 1984 as a conference named “Technology, Entertainment and Design,” TED talks have taken place annually since 1990. The talks are now distributed freely across the internet and in more than 100 different languages. Over the years, TED has established its brand as one of the best curators of lectures and speeches on the internet. The quality of talks is widely known and respected across the world. In a time of information overload, TED talks provide a high standard for knowledge and the art of public speaking. The talks aim to challenge preconceiving notions and find new ways to express different facets of the human experience.

This report will focus on analyzing the attributes and transcripts of TED talks, and then determining if it is possible to predict the ratings and popularity of TED talks based upon these transcripts and attributes. The analyses and findings from this report will be useful to several parties. For TED.com, it will provide them with better curator abilities and more criteria for judging future submissions. By knowing if a talk will be popular or high-rated, it will allow them to focus their marketing of that talk in a more strategic way. The findings of this project will also be of importance to future participants who want to know what sort of material and speeches become popular and successful on the TED platform. In that way a presenter can then structure talks to maximize the attributes that are popular and reach the widest audience possible. For the analysis, exploratory analysis will be done on TED talks in general followed by the use of machine learning methods to predict popular and highly rated TED talks.

Analysis and Models

About the Data

Source Data

The data set comprises two sets of data. The first dataset (“the main dataset”) comprises 2550 observations with 17 variables. The second data (“the transcript dataset”) contains a transcript of the talk that was translated into English (if the talk was a non-English talk) and then converted into a textual transcript. A URL is used as a link between the datasets to shows which main observation goes together with which transcript.

Note that there are only 2467 transcripts. Reason for this being that some of the talks were not delivered as spoken word, and hence could not be transcribed to text. In other cases, it was not possible to translate the talk to English and then transcribe.

The structure of the raw main data is shown in Figure 1: Structure of TED Main dataset.

```
FP : Start'data.frame': 2550 obs. of  17 variables:
 $ comments          : int  4553 265 124 200 593 672 919 46 852 900 ...
 $ description       : Factor w/ 2550 levels "'I am a mathematician, and I would like to stand on your roof.' Th
Ron Eglash greeted many African fam"| __truncated__,..: 1836 2492 1510 1035 2527 2115 2412 198 1630 1600 ...
 $ duration          : int  1164 977 1286 1116 1190 1305 992 1198 1485 1262 ...
 $ event             : Factor w/ 355 levels "AORN Congress",..: 80 80 80 80 80 80 80 80 80 80 ...
 $ film_date         : int  1140825600 1140825600 1140739200 1140912000 1140566400 1138838400 1140739200 1140652
1138838400 1140825600 ...
 $ languages         : int  60 43 26 35 48 36 31 19 32 31 ...
 $ main_speaker      : Factor w/ 2156 levels " OK Go"," Rodrigo y Gabriela",..: 1132 38 512 1259 773 2062 1075 1
1693 ...
 $ name              : Factor w/ 2550 levels "Aakash Odedra: A dance in a hurricane of paper, wind and light",..
607 1485 918 2442 1270 1246 513 1993 ...
 $ num_speaker       : int  1 1 1 1 1 1 1 1 1 1 ...
 $ published_date    : int  1151367060 1151367060 1151367060 1151367060 1151440680 1151440680 1152490260 1152490
1153181460 1153181460 ...
 $ ratings           : Factor w/ 2550 levels "[{'id': 1, 'name': 'Beautiful', 'count': 100}, {'id': 9, 'name': '
'count': 97}, {'id': 3, 'name': '"| __truncated__,..: 1858 1928 1977 1801 2471 1825 1744 2440 1765 837 ...
 $ related_talks     : Factor w/ 2550 levels "[{'id': 1, 'hero':
'https://pe.tedcdn.com/images/ted/a2194b4ef5170bd9e9b086c5053e09cdf3545960_2880x1620.jpg', '"| __truncated__,..:
810 56 1096 1256 1196 2266 2225 2501 ...
 $ speaker_occupation: Factor w/ 1458 levels " Chairman of the Cordoba Initiative",..: 143 281 1341 14 637 790 2
985 ...
 $ tags              : Factor w/ 2530 levels "['3d printing', 'art', 'bacteria', 'biology', 'biomimicry', 'biote
'collaboration', 'crowdsourcing', 'micro'"| __truncated__,..: 1003 291 1175 1799 121 882 1047 423 1650 1048 ...
 $ title             : Factor w/ 2550 levels " Hidden miracles of the natural world",..: 589 413 1564 731 1691 2
429 1212 171 ...
 $ url               : Factor w/ 2550 levels
"https://www.ted.com/talks/9_11_healing_the_mothers_who_found_forgiveness_friendship\n",..: 1334 44 610 1480 917
1244 517 1989 ...
 $ views             : int  47227110 3200520 1636292 1697550 12005869 20685401 3769987 967741 2567958 3095993 ..
```

Figure 1: Structure of TED Main dataset

The structure of the raw transcript dataset is shown in Figure 2: Structure of TED Transcript dataset.

```
'data.frame':   2467 obs. of  2 variables:
 $ transcript: Factor w/ 2464 levels "'Theme and variations' is one of those forms that require a certain kind of
intellectual activity, because you "| __truncated__,..: 419 1843 84 1143 134 1854 1473 658 1268 738 ...
 $ url       : Factor w/ 2464 levels
"https://www.ted.com/talks/9_11_healing_the_mothers_who_found_forgiveness_friendship\n",..: 1297 43 603 1438 898 2357 1234
1212 512 1926 ...
```

Figure 2: Structure of TED Transcript dataset

A sample of the first 90 characters of transcribed text for the first 40 talks on file is shown in Figure 3: Sample of Transcript data.

```
 [1] "Good morning. How are you?(Laughter)It's been great, hasn't it? I've been blown away by th"
 [2] "Thank you so much, Chris. And it's truly a great honor to have the opportunity to come to "
 [3] "(Music: \"The Sound of Silence,\" Simon & Garfunkel)Hello voice mail, my old friend.(Laughte"
 [4] "If you're here today â€" and I'm very happy that you are â€" you've all heard about how su"
 [5] "About 10 years ago, I took on the task to teach global development to Swedish undergraduat"
 [6] "Thank you. I have to tell you I'm both challenged and excited. My excitement is: I get a c"
 [7] "On September 10, the morning of my seventh birthday, I came downstairs to the kitchen, whe"
 [8] "I'm going to present three projects in rapid fire. I don't have much time to do it. And I "
 [9] "It's wonderful to be back. I love this wonderful gathering. And you must be wondering, \"Wh"
[10] "I'm often asked, \"What surprised you about the book?\" And I say, \"That I got to write it.\""
[11] "I'm going to take you on a journey very quickly. To explain the wish, I'm going to have to"
[12] "I can't help but this wish: to think about when you're a little kid, and all your friends "
[13] "I'm the luckiest guy in the world. I got to see the last case of killer smallpox in the wo"
[14] "I'm really excited to be here today. I'll show you some stuff that's just ready to come ou"
[15] "I've been at MIT for 44 years. I went to TED I. There's only one other person here, I thin"
[16] "(Music)(Music ends)(Applause)(Applause ends)Hi, everyone. I'm Sirena. I'm 11 years old and"
[17] "(Music)(Music ends)(Applause)Thank you!(Applause continues)Thank you very much. Like the s"
[18] "In terms of invention, I'd like to tell you the tale of one of my favorite projects. I thi"
[19] "My name is Lovegrove. I only know nine Lovegroves, two of which are my parents. They are f"
[20] "Charles Van Doren, who was later a senior editor of Britannica, said the ideal encyclopedi"
[21] "I'm Rich Baraniuk and what I'd like to talk a little bit about today are some ideas that I"
[22] "You know, when Chris first approached me to speak at TED, I said no, because I felt like I"
[23] "Over the past couple of days, as I've been preparing for my speech, I've become more and m"
[24] "I'd like to talk today about the two biggest social trends in the coming century, and perh"
[25] "I bet you're worried.(Laughter)I was worried. That's why I began this piece. I was worried"
[26] "We've been told to go out on a limb and say something surprising. So I'll try and do that,"
[27] "My title: \"Queerer than we can suppose: the strangeness of science.\" \"Queerer than we can "
[28] "You'll be happy to know that I'll be talking not about my own tragedy, but other people's "
[29] "I think I was supposed to talk about my new book, which is called \"Blink,\" and it's about "
[30] "When you have 21 minutes to speak, two million years seems like a really long time. But ev"
[31] "I'm going to talk to you about some stuff that's in this book of mine that I hope will res"
[32] "Thank you. It's really an honor and a privilege to be here spending my last day as a teena"
[33] "18 minutes is an absolutely brutal time limit, so I'm going to dive straight in, right at "
[34] "I'll just take you to Bangladesh for a minute.Before I tell that story, we should ask ours"
[35] "I want to start with a story, a la Seth Godin, from when I was 12 years old. My uncle Ed g"
[36] "Video: Narrator: An event seen from one point of view gives one impression. Seen from anot"
[37] "A public, Dewey long ago observed, is constituted through discussion and debate. If we are"
[38] "I want to start off by saying, Houston, we have a problem. We're entering a second generat"
[39] "This is me. My name is Ben Saunders. I specialize in dragging heavy things around cold pla"
[40] "Well, as Alexander Graham Bell famously said on his first successful telephone call, \"Hell"
[41] "Walk around for four months with three wishes  and all the ideas will start to percolate u"
```

Figure 3: Sample of Transcript data

A sample of the transcript URLs is shown in Figure 4: Sample transcript URL data.

```
 [2] https://www.ted.com/talks/al_gore_on_averting_climate_crisis\n
 [3] https://www.ted.com/talks/david_pogue_says_simplicity_sells\n
 [4] https://www.ted.com/talks/majora_carter_s_tale_of_urban_renewal\n
 [5] https://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen\n
 [6] https://www.ted.com/talks/tony_robbins_asks_why_we_do_what_we_do\n
 [7] https://www.ted.com/talks/julia_sweeney_on_letting_go_of_god\n
 [8] https://www.ted.com/talks/joshua_prince_ramus_on_seattle_s_library\n
 [9] https://www.ted.com/talks/dan_dennett_s_response_to_rick_warren\n
[10] https://www.ted.com/talks/rick_warren_on_a_life_of_purpose\n
[11] https://www.ted.com/talks/cameron_sinclair_on_open_source_architecture\n
[12] https://www.ted.com/talks/jehane_noujaim_inspires_a_global_day_of_film\n
[13] https://www.ted.com/talks/larry_brilliant_wants_to_stop_pandemics\n
[14] https://www.ted.com/talks/jeff_han_demos_his_breakthrough_touchscreen\n
[15] https://www.ted.com/talks/nicholas_negroponte_on_one_laptop_per_child\n
[16] https://www.ted.com/talks/sirena_huang_dazzles_on_violin\n
[17] https://www.ted.com/talks/jennifer_lin_improvs_piano_magic\n
[18] https://www.ted.com/talks/amy_smith_shares_simple_lifesaving_design\n
[19] https://www.ted.com/talks/ross_lovegrove_shares_organic_designs\n
[20] https://www.ted.com/talks/jimmy_wales_on_the_birth_of_wikipedia\n
```

Figure 4: Sample transcript URL data

The data across both the TED main as well as the TED transcripts has either a 1:1 cardinality or a 1:many cardinality style.

- Each TED talks observation equates to one talk
- Each TED talks observation is represented by a unique URL.
- The variables “related talks”, “tags”, “speakers”, “ratings” and “comments” each have their own 1:many style represented within their data attribute.
- All other variables have a 1:1 relationship with the TED talks eg “title”, “event”, duration” etc
- The conceptual representation of the data mode is shown in Figure 5: Conceptual Data Model.

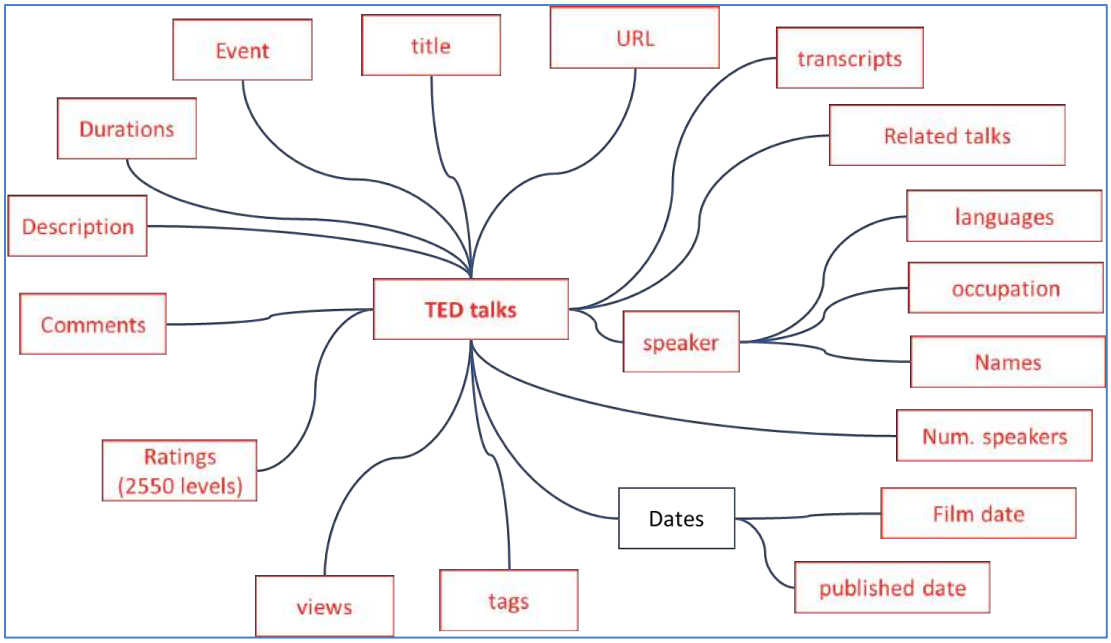
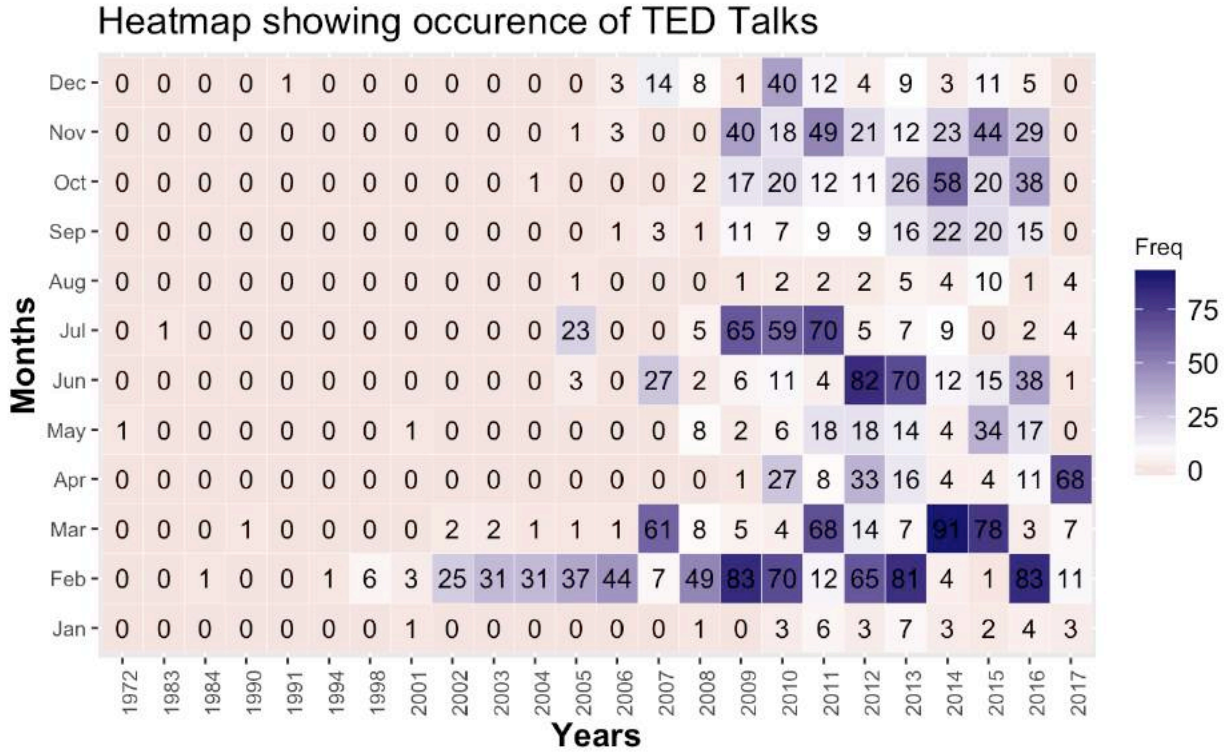


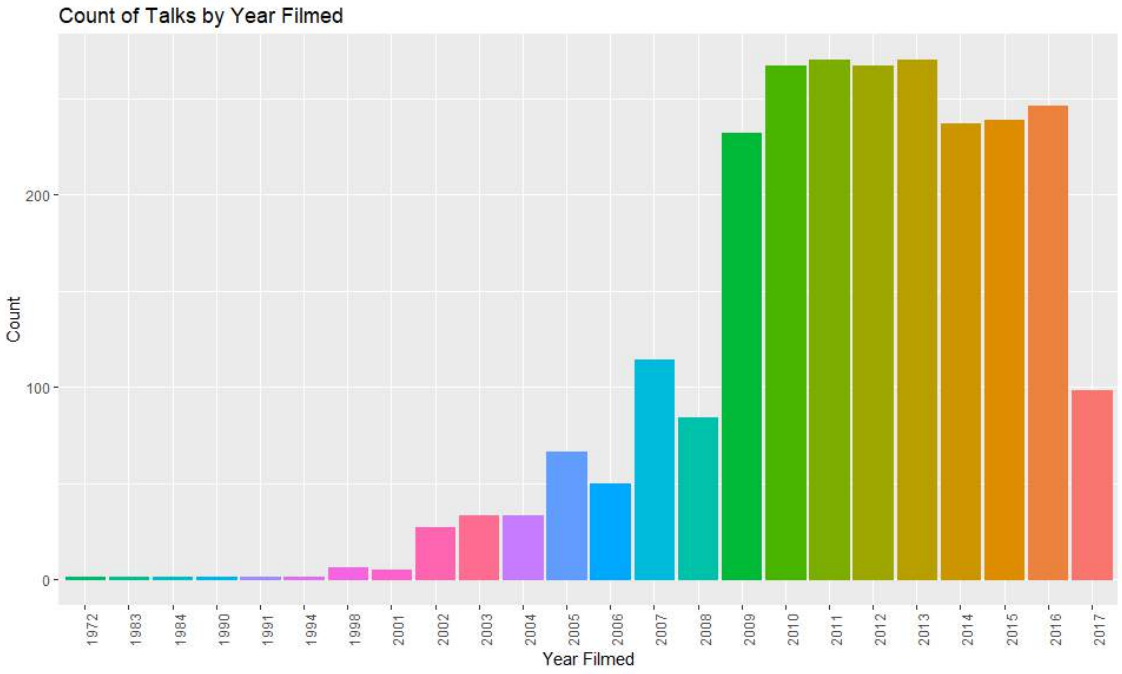
Figure 5: Conceptual Data Model

Understand Date Filmed

Reviewing talks by Date Filmed shows how varied the filming month was. Talks pre-2007 appear to follow a specific publishing schedule where talks post-2007 appear to be published on a more random basis.

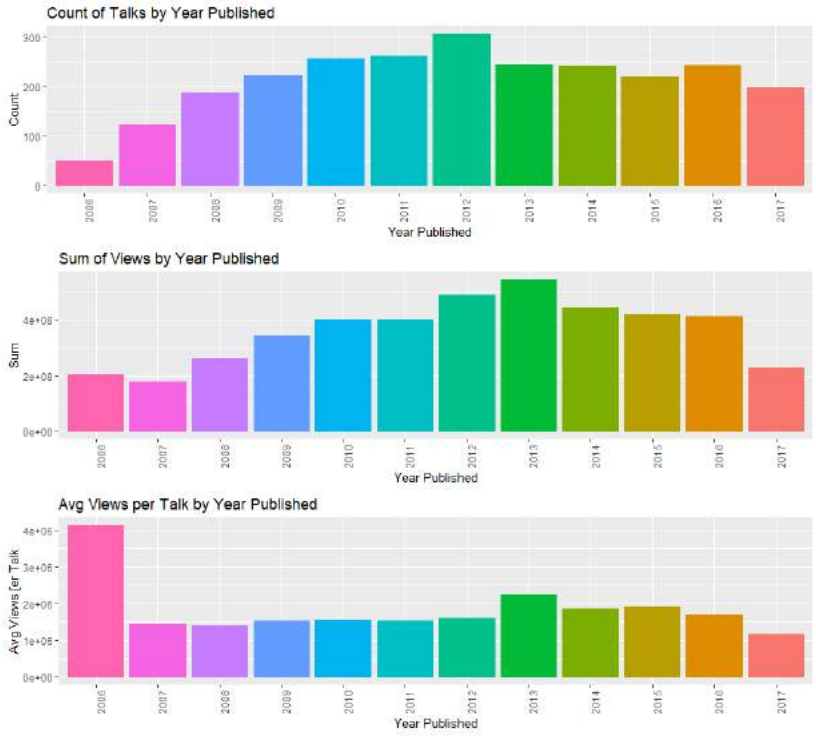


To further understand the filming pattern, a histogram of talks by year filmed was produced, showing that most talks were produced in the 2009 to 2016 year range.



Understand Date Published

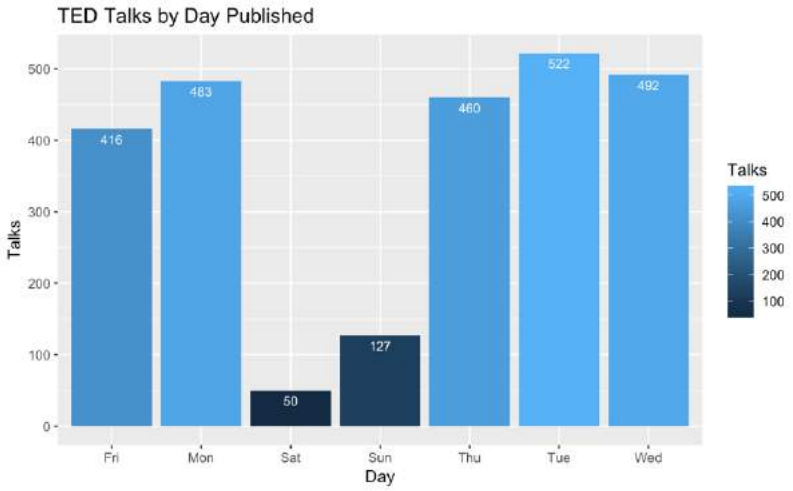
This led to an analysis of talks by year published which shows some strange anomalies.



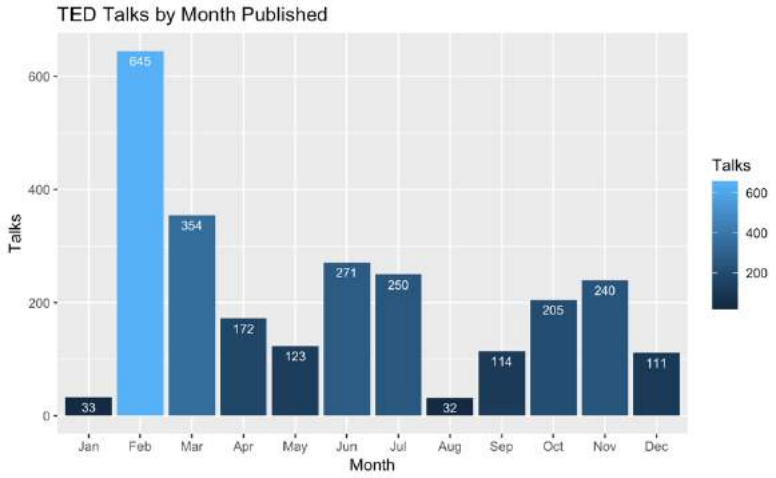
- Firstly, there is a wide disparity between the year talks were filmed and the year they were published.

- Secondly, looking at talks published vs number of views in the same year, it appears as if the talks and views follow the same pattern.
- However, taking the average views per talk, the chart shows that the average views per talk was far higher in 2006 than in any other year.
- It is important to note that 2006 was a pivotal year for TED talks – this was the year in which the organization started publishing free talks.

The publish days and months were also looked at:



- From this visual, there is a trend of TED talks being published on week days and not weekends. This would make sense since most people are out and about on the weekend and probably less likely to want to stay indoors and watch an educational talk.



- Here a trend develops of TED talks mostly being published in February, but this is in part due to TED talks mostly taking place in February before 2006.

Speaker Occupations

The speaker occupation column had an issue with speakers who had multiple occupations. It was difficult to pinpoint one occupation for each speaker. After some transforming, the top 30 speaker occupations are displayed below:

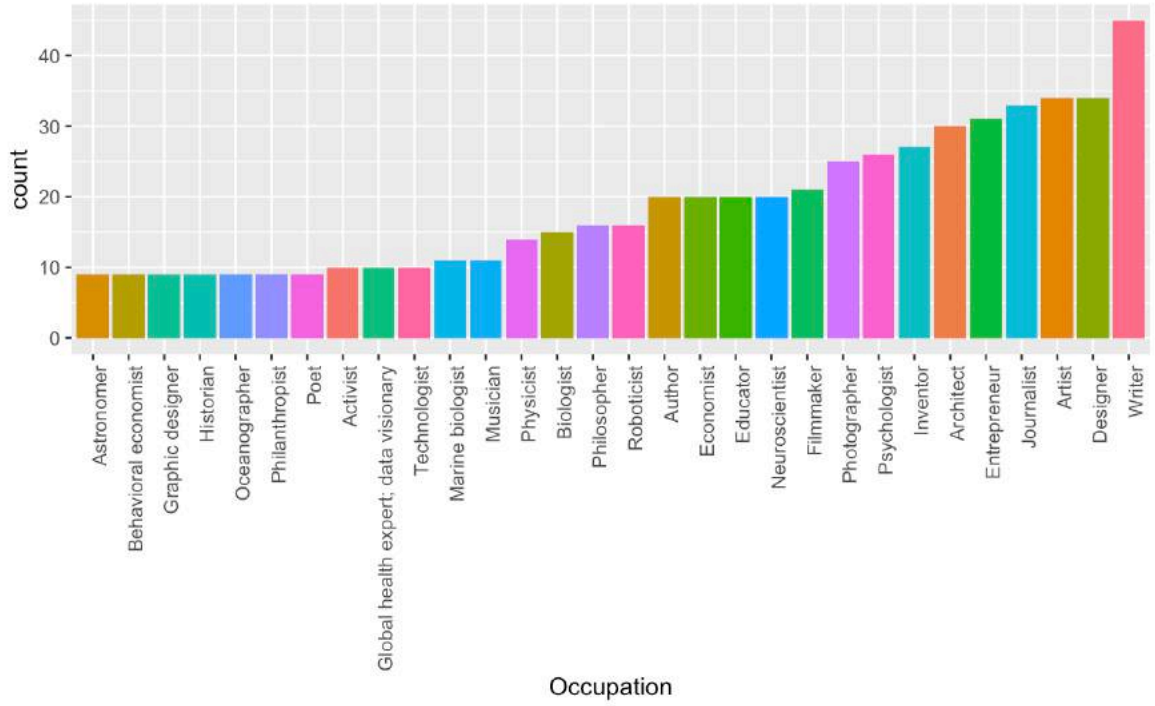


Figure 6, Top 30 Speaker Occupations

Tags

The *tags* column is cleaned up by untokenizing all the words, resulting in **441 unique tags**. A word cloud is then built to visualize the different types of tags associated with each talk:

Figure 7, Tags with minfreq=20 from all TED talks

The word cloud illustrates the diverse range of topics and industries represented by all of the talks.

The top 10 tags are also shown:

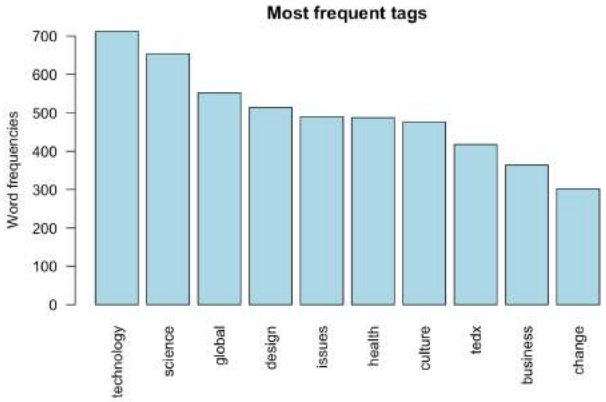


Figure 8, Top 10 tags from all TED talks

Quality

1. Missing Transcripts (2550 TED main vs 2467 TED transcripts). This was handled by reducing the dataset to only those rows where both a talk observation as well as a transcript were present.
2. Some transcript URLs are duplicated.

Transformations

To assist with further analysis, a transformed dataset was created; containing attributes from the original two datasets, with the following added:

- Standardized Ratings
- Ratings Sentiment
- Popularity
- Highrated
- Tag Sentiment
- Transcript Enrichment Extraction

Standardized Ratings

Using data contained in the Ratings variable, a set of Standardized Ratings were created according to defined rules which mapped Original Ratings to Standardized Ratings.

There are 14 original rating types, being:

	Name
1	Unconvincing
2	Informative
3	Inspiring
4	OK
5	Fascinating
6	Ingenious
7	Confusing
8	Obnoxious
9	Beautiful
10	Longwinded
11	Persuasive
12	Jaw-dropping
13	Courageous
14	Funny

A viewer assigns a rating to a talk; the original rating data per talk shows how many times each rating was assigned.

As an example, Original Ratings for the first two talks followed this structure:

```
[1] [{'id': 7, 'name': 'Funny', 'count': 19645}, {'id': 1, 'name': 'Beautiful', 'count': 4573}, {'id': 9, 'name': 'Ingenious', 'cou
nt': 6073}, {'id': 3, 'name': 'Courageous', 'count': 3253}, {'id': 11, 'name': 'Longwinded', 'count': 387}, {'id': 2, 'name': 'Conf
using', 'count': 242}, {'id': 8, 'name': 'Informative', 'count': 7346}, {'id': 22, 'name': 'Fascinating', 'count': 10581}, {'id': 2
1, 'name': 'Unconvincing', 'count': 300}, {'id': 24, 'name': 'Persuasive', 'count': 10704}, {'id': 23, 'name': 'Jaw-dropping', 'cou
nt': 4439}, {'id': 25, 'name': 'OK', 'count': 1174}, {'id': 26, 'name': 'Obnoxious', 'count': 209}, {'id': 10, 'name': 'Inspiring',
 'count': 24924}]
[2] [{'id': 7, 'name': 'Funny', 'count': 544}, {'id': 3, 'name': 'Courageous', 'count': 139}, {'id': 2, 'name': 'Confusing', 'count
': 62}, {'id': 1, 'name': 'Beautiful', 'count': 58}, {'id': 21, 'name': 'Unconvincing', 'count': 258}, {'id': 11, 'name': 'Longwind
ed', 'count': 113}, {'id': 8, 'name': 'Informative', 'count': 443}, {'id': 10, 'name': 'Inspiring', 'count': 413}, {'id': 22, 'name
': 'Fascinating', 'count': 132}, {'id': 9, 'name': 'Ingenious', 'count': 56}, {'id': 24, 'name': 'Persuasive', 'count': 268}, {'id'
: 23, 'name': 'Jaw-dropping', 'count': 116}, {'id': 26, 'name': 'Obnoxious', 'count': 131}, {'id': 25, 'name': 'OK', 'count': 203}]
```


Figure 9: Sample Original Ratings Data

For talk #1, 19645 viewers assigned the rating Funny, 4573 viewers assigned the rating Beautiful, 6073 assigned the rating Ingenious and so on. If a rating was NOT assigned to the talk, it does not appear in the list for that talk.
For talk #2, 544 viewers assigned the rating Funny, etc
And so on, for each talk.

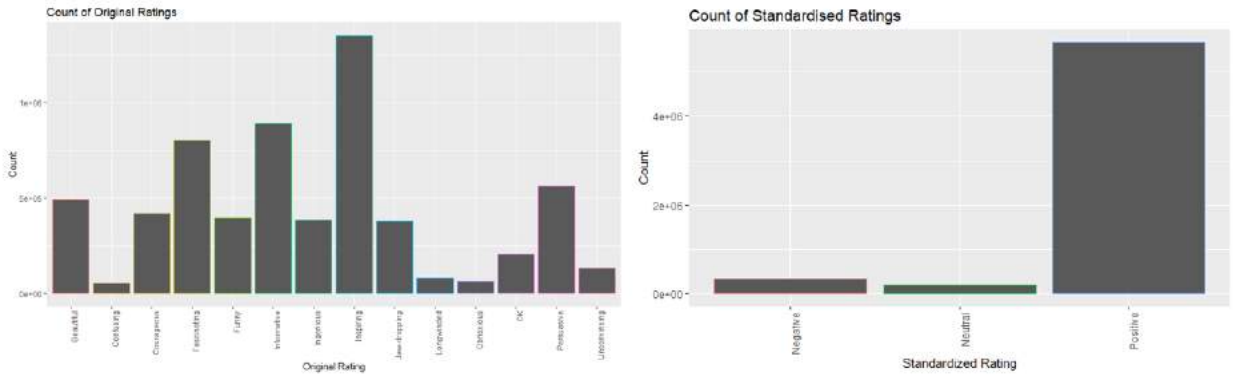
The Positive/Negative mapping was done based on assigning Original Ratings to Standardized Ratings.

There are three Standardized Ratings: Positive, Neutral and Negative.

The Mapping assign Original Rating to Standardized Rating as follows:

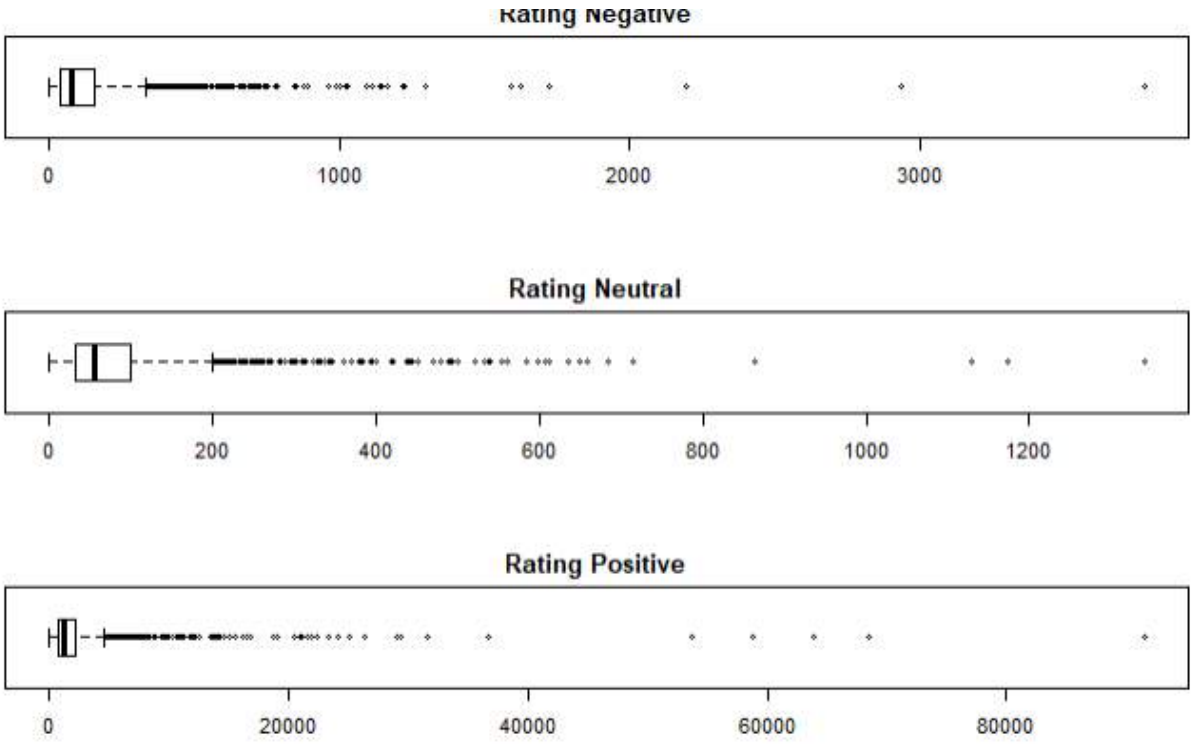
- Positive <- 'Informative', 'Inspiring', 'Fascinating', 'Ingenious', 'Beautiful', 'Persuasive', 'Jaw-dropping', 'Courageous', 'Funny'
- Negative <- 'Unconvincing', 'Confusing', 'Obnoxious', 'Longwinded'
- Neutral <- 'OK'

A summary of Original Ratings counts together with Standardized Ratings counts is shown in the below two charts:



Descriptive Statistics for the Standardised Ratings summarised count are shown in this figure:

```
Descriptive Stats for summarized count of Negative Standardized Sentiment
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0      39      78     133     157    3777

Descriptive Stats for summarized count of Neutral Standardized Sentiment
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0      32      56      81      99    1341

Descriptive Stats for summarized count of Positive Standardized Sentiment
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
     45     729    1279    2228    2254   91538
```



Ratings Sentiment

Using the Standardized Ratings, a count of ratings per Standardized Rating was obtained for each talk and the Ratings Sentiment was derived from this.

A variable called *positivecount* is created as a way to group all positive ratings:

- 'Informative', 'Inspiring', 'Fascinating', 'Ingenious', 'Beautiful', 'Persuasive', 'Jaw-dropping', 'Courageous', 'Funny'

A variable called *negativecount* is created as a way to group all negative ratings:

- 'Unconvincing', 'Confusing', 'Obnoxious', 'Longwinded'

The two neural ratings of ‘OK’ and ‘Neutral’ are omitted from this rating categorization since only two categories were desired. These two new variables will later on be used to create the machine learning formula to classify a talk as high-rated.

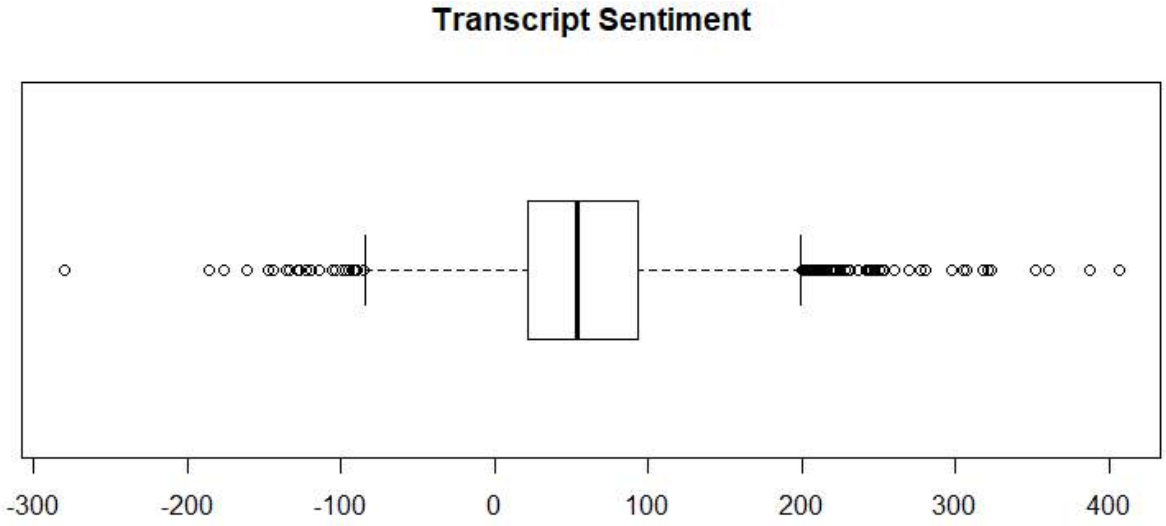
Tag Sentiment

Using the transcript wording, a sentiment value (positive or negative) is derived for each transcript.
This is done by:

1) Assign a sentiment value to each word in the talk using the AFINN word rating scheme from the tidytext library.
2) Summarize all sentiment values for a talk

The Descriptive Statistics of Transcript Sentiment is shown here:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
-280	22	54	60	93	406	86



Transcript Enrichment Extraction

Each transcript is derived from a corresponding talk. Within this talk, there are Enrichment attributes describing audience emotion (eg laughter or clapping) and talk enrichment actions (eg music or video).
These enrichment attributes have been extracted and included in the transformed dataset.
Values that are included are:

1. Talk Has Laughter: Value = 1 for yes and 0 for no

2. Talk Has Clapping: Value = 1 for yes and 0 for no
3. Talk Has Music: Value = 1 for yes and 0 for no
4. Talks Has Video: Value = 1 for yes and 0 for no

Tags Sentiment

The individual tags from each talk are then given a score by the AFINN sentiment score. A total cumulative score was given to each talk. On the left is an example of the different kinds of tags for each unique talk ID. On the right is a cumulative sentiment score for tags associated with each unique talk ID.

	id	tags1
1	1	children
1.1	1	creativity
1.2	1	culture
1.3	1	dance
1.4	1	education
1.5	1	parenting
1.6	1	teaching
2	2	alternative
2.1	2	energy
2.2	2	cars
2.3	2	climate
2.4	2	change
2.5	2	culture
2.6	2	environment
2.7	2	global
2.8	2	issues
2.9	2	science
2.10	2	sustainability
2.11	2	technology
3	3	computers
3.1	3	entertainment
3.2	3	interface
3.3	3	design
3.4	3	media
3.5	3	music
3.6	3	performance
3.7	3	simplicity
3.8	3	software
3.9	3	technology

	id	score
1	4	1.000
2	6	1.000
3	7	1.333
4	9	1.000
5	10	1.667
6	11	-2.000
7	12	2.000
8	18	1.000
9	22	1.500
10	24	3.000
11	25	3.000
12	30	2.000
13	31	2.500
14	32	-1.000
15	33	2.000
16	34	-1.000
17	35	0.000
18	37	-1.000
19	39	1.500
20	40	-1.000
21	42	2.000
22	44	1.000
23	45	1.000
24	47	1.000
25	48	2.500
26	49	3.000
27	50	2.000
28	51	-1.000

Figure 10, Example of tags associated with each talk ID

Figure 11, , Each talk ID is assigned tags sentiment score

This is done to add another variable (*tagscore*) for analysis for the machine learning predictive models.

Popularity and High-Rated Columns

In order to predict popularity and ratings of a talk, all of the talks needed to be classified by creating criteria to differentiate them. The response variable columns of *popularity* and *highrated* are made as defined below:

Popularity

- Using the *views* per talk, a popularity rating is assigned to each talk. If the *views* value for the **talk is in the top 25% of all *views* per talk**, then the talk is classified as a popular talk.
- This is a binary rating – either the talk is popular (value of 1) or the talk is not (value of 0).
- Once created, **617** of the talks are now classified as popular.

High-Rated

- Using the *positivecount* variable (a cumulative count of all positive ratings for each talk), a high-rated rating is assigned to each talk. If the *positivecount* value for the **talk is in the top 25% of *positivecount* values per talk**, then the talk is classified as high-rated talk.
- This is a binary rating – either the talk is high-rated (value of 1) or the talk is not (value of 0).
- Originally, the criteria for a high-rated talk was going to be defined as a talk having more positive ratings than negative ratings, but due to the lopsided amount of positive rating counts, the top 25% quartile method was chosen as the best way to highlight a talk as being exceptionally rated. The possible reason for the imbalance of positive to negative ratings speaks to the quality in general of TED talks.
- Once created, **616** of the talks are now classified as high-rated.

The number of talks that are ***both* popular and high-rated** is **466** talks. These talks are the upper-tier of TED talks.

By separating talks into these subgroups, it is now possible to do more comparative analysis of popular vs non-popular, high-rated vs not high-rated, and also explore the

possible differences between the different categories for aspects such as types of words used in each talk.

Word Cloud for all TED Talks

By analyzing the transcripts of each talk, a word cloud is created in order to have a better understanding of the kinds of words used in all 2,467 TED talks:

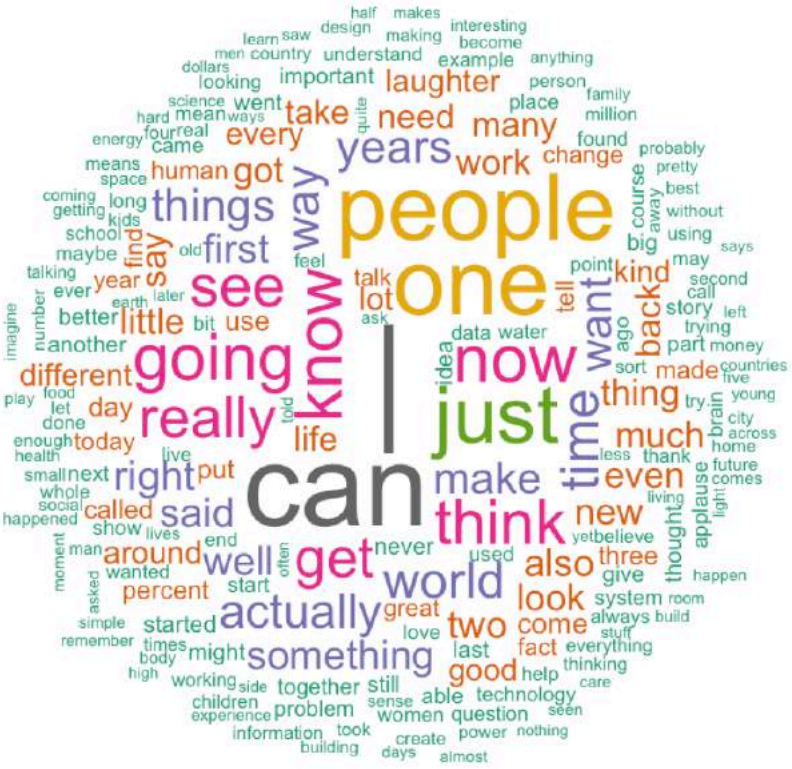
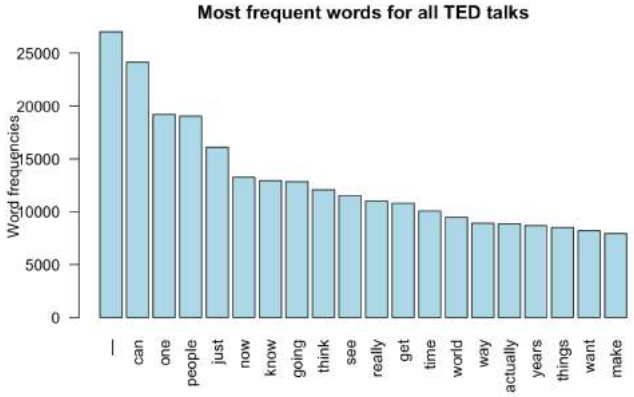


Figure 12, Word cloud for all TED talks, minfreq=1500

The top 20 most frequent words in all TED talks was also found out:



Word Cloud Comparisons

Now that it was possible to specify a talk as popular or high-rated, four different word clouds were made in order to compare the words used in **popular talks** vs **non-popular talks** and **high-rated talks** vs **non high-rated talks**:

Figure 13, Word cloud for popular talks, minfreq=500

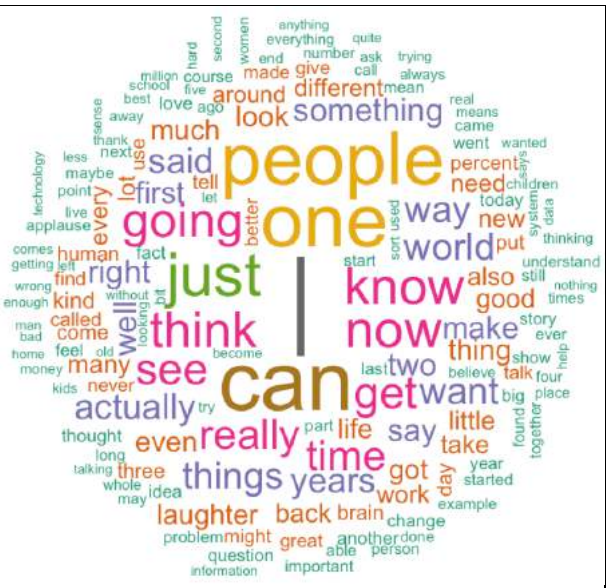


Figure 14, Word cloud for high-rated talks, minfreq=500

Figure 15, Word cloud for non-popular talks, minfreq=1500

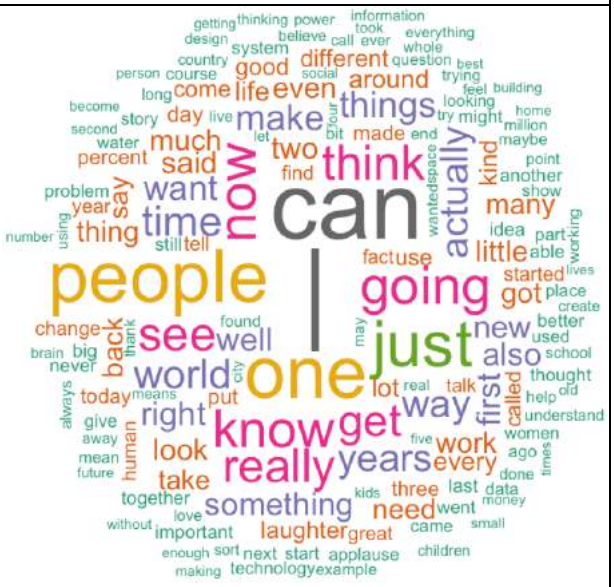
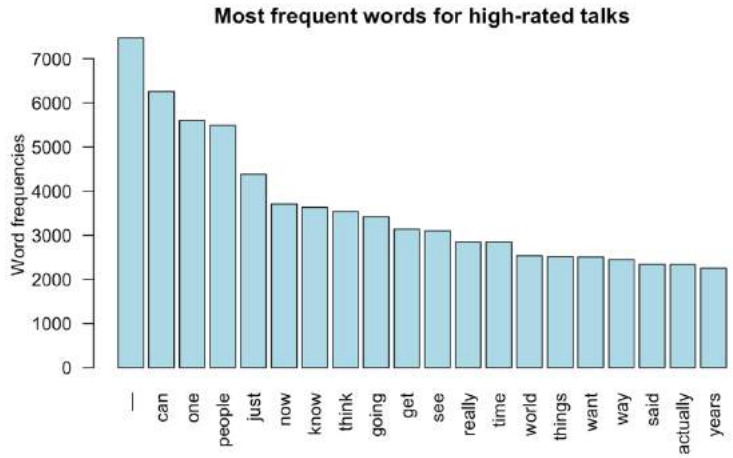
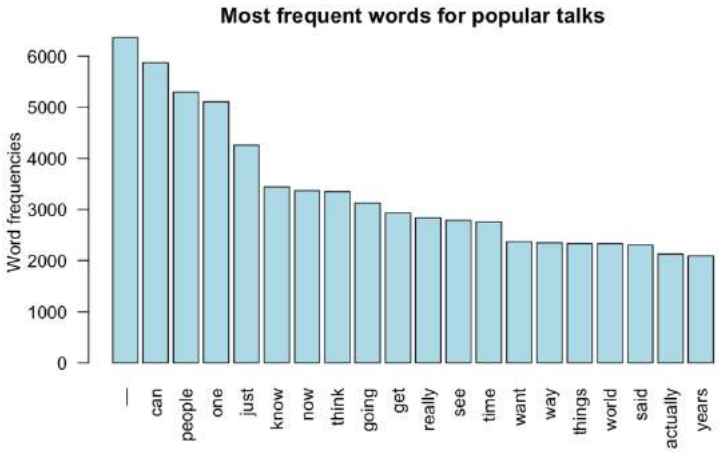


Figure 16, Word cloud for non high-rated talks, minfreq=1350

Top Words for High-Rated and Popular Talks

The top 20 most frequent words for high-rated and popular talks were also examined and end up being very similar:





Model 1: Feature Comparison

An understanding of the relationships between different attributes and popularity/high-rated talks was desired. To do this, logistic regression was run to look at the variable selection and see how variables were ranked in regards to the popularity column. In this case, just the transcript and ratings are compared. It is interesting to see that some specific talks have more of a relationship with the popularity variable than others:

	Variable	Estimate	Std. Error	z value	Pr(>|z|)	Wald ChiSquare	Rank
highrated1	highrated1	3.55e+00	1.27e-01	28.08	1.65e-173	788.57	1.0
positivecount	positivecount	1.13e+02	5.06e+00	22.35	1.22e-110	499.49	2.0
fascinating	fascinating	8.43e+01	4.13e+00	20.41	1.44e-92	416.45	3.0
persuasive	persuasive	2.53e+01	1.30e+00	19.44	3.70e-84	377.82	5.0
inspiring	inspiring	2.53e+01	1.30e+00	19.44	3.70e-84	377.82	5.0
longwinded	longwinded	2.53e+01	1.30e+00	19.44	3.70e-84	377.82	5.0
informative	informative	4.18e+01	2.20e+00	19.01	1.49e-80	361.26	7.5
ingenius	ingenius	4.18e+01	2.20e+00	19.01	1.49e-80	361.26	7.5
languages	languages	1.07e+01	5.86e-01	18.25	2.17e-74	332.96	9.0
comments	comments	3.77e+01	2.19e+00	17.19	3.04e-66	295.57	10.0
negativecount	negativecount	1.77e+01	1.20e+00	14.74	3.66e-49	217.22	11.0
beautiful	beautiful	3.63e+01	2.47e+00	14.71	5.22e-49	216.51	12.0
funny	funny	7.81e+01	5.45e+00	14.35	1.12e-46	205.83	13.0
jawdropping	jawdropping	7.29e+01	5.42e+00	13.44	3.44e-41	180.68	14.0
courageous	courageous	2.78e+01	2.11e+00	13.19	9.99e-40	173.98	15.0
confusing	confusing	9.06e+00	9.24e-01	9.81	1.02e-22	96.24	16.0
unconvincing	unconvincing	1.14e+01	1.39e+00	8.20	2.48e-16	67.18	17.0
obnoxious	obnoxious	1.24e+01	1.62e+00	7.69	1.46e-14	59.15	18.0
sentiment	sentiment	2.09e+00	5.04e-01	4.16	3.23e-05	17.27	19.0
num_tags	num_tags	-1.36e+00	3.64e-01	-3.73	1.91e-04	13.91	20.0
duration	duration	1.05e+00	4.80e-01	2.18	2.93e-02	4.75	21.0
views	views	2.79e+06	2.45e+06	1.14	2.55e-01	1.30	22.0

The same ranking formula was done for popularity but now with the normalized dataframe and more variables:

	Variable	Estimate	Std. Error	z value	Pr(>|z|)	Wald ChiSquare	Rank
highrated1	highrated1	3.55e+00	1.27e-01	2.81e+01	1.65e-173	7.89e+02	1.0
positivecount	positivecount	1.90e-03	8.52e-05	2.23e+01	1.22e-110	4.99e+02	2.0
fascinating	fascinating	5.83e-03	2.86e-04	2.04e+01	1.44e-92	4.16e+02	3.0
persuasive	persuasive	1.89e-02	9.70e-04	1.94e+01	3.70e-84	3.78e+02	5.0
inspiring	inspiring	1.89e-02	9.70e-04	1.94e+01	3.70e-84	3.78e+02	5.0
longwinded	longwinded	1.89e-02	9.70e-04	1.94e+01	3.70e-84	3.78e+02	5.0
informative	informative	4.27e-03	2.25e-04	1.90e+01	1.49e-80	3.61e+02	7.5
ingenius	ingenius	4.27e-03	2.25e-04	1.90e+01	1.49e-80	3.61e+02	7.5
negativecount	negativecount	4.68e-03	3.18e-04	1.47e+01	3.66e-49	2.17e+02	9.0
beautiful	beautiful	3.84e-03	2.61e-04	1.47e+01	5.22e-49	2.17e+02	10.0
funny	funny	3.98e-03	2.77e-04	1.43e+01	1.12e-46	2.06e+02	11.0
jawdropping	jawdropping	4.95e-03	3.68e-04	1.34e+01	3.44e-41	1.81e+02	12.0
courageous	courageous	3.21e-03	2.44e-04	1.32e+01	9.99e-40	1.74e+02	13.0
confusing	confusing	1.71e-02	1.74e-03	9.81e+00	1.02e-22	9.62e+01	14.0
unconvincing	unconvincing	5.18e-03	6.32e-04	8.20e+00	2.48e-16	6.72e+01	15.0
obnoxious	obnoxious	9.15e-03	1.19e-03	7.69e+00	1.46e-14	5.91e+01	16.0
tagscore	tagscore	2.63e-01	4.75e-02	5.54e+00	3.02e-08	3.07e+01	17.0
sentiment	sentiment	3.05e-03	7.34e-04	4.16e+00	3.23e-05	1.73e+01	18.0
id	id	2.07e-05	6.53e-05	3.17e-01	7.51e-01	1.01e-01	19.0
transcript.941	transcript	5.31e+01	4.36e+05	1.22e-04	1.00e+00	1.48e-08	20.0
transcript.1765	transcript	5.31e+01	4.36e+05	1.22e-04	1.00e+00	1.48e-08	21.0
transcript.347	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	22.0
transcript.1554	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	23.0
transcript.2167	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	24.0
transcript.1460	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	25.0
transcript.322	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	26.0
transcript.1839	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	27.0
transcript.1642	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	28.0
transcript.2171	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	29.0
transcript.650	transcript	5.31e+01	5.04e+05	1.05e-04	1.00e+00	1.11e-08	30.0

Figure 17, logistic regression for popularity in Norm_TED dataframe

These chi-square rankings help get another perspective of the important variables that play a role in predicting popularity. It also makes sense that the more negative ratings are some of the lowest ranked features.

Model 2: Correlation

A correlation matrix is also created to see the correlation between variables with a focus on views since views is the most important attribute for determining popularity:

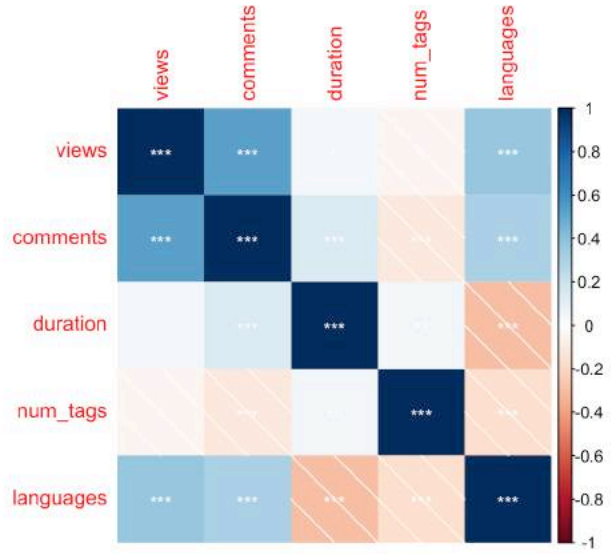


Figure 18, Correlation matrix for TED talk attributes

- There seems to be slight correlation between the pairs of views/comments and views/languages.

Another correlation matrix to examine only the ratings and views is shown:

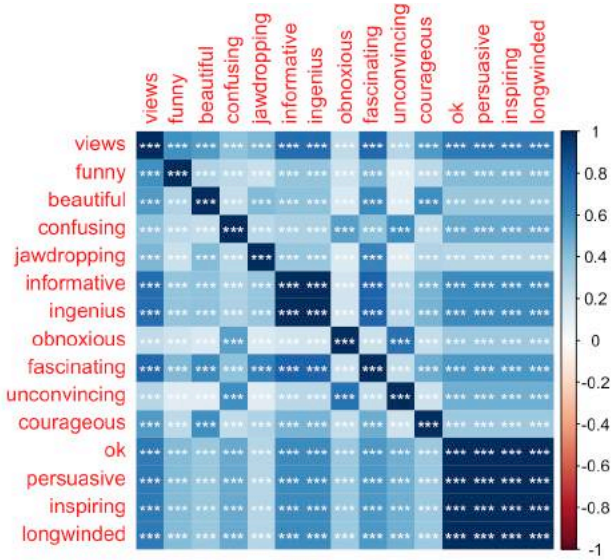


Figure 19, Correlation matrix for TED talk ratings categories

- Most of the ratings show as being slightly correlated to views so it is hard to determine specific patterns with certain ratings.

Model 3: Associated Rule Mining

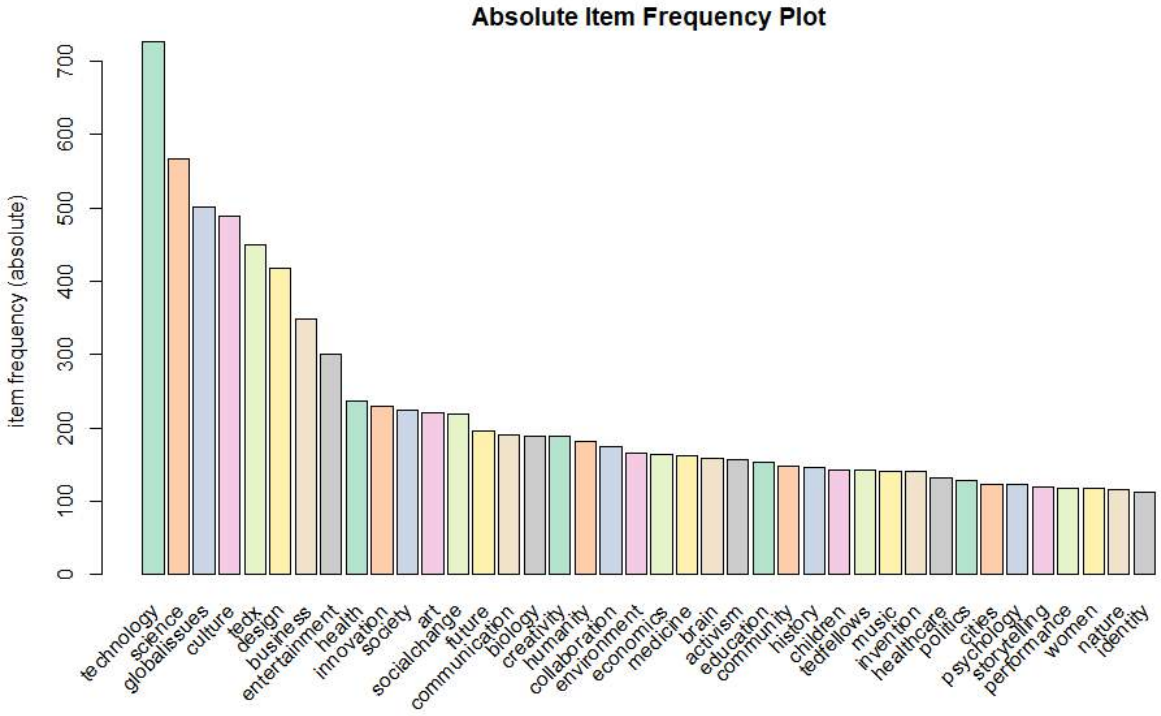
The objective of Association Rule mining is to determine, from a basket of rules, which appear together and with what level of confidence, support and lift.

- Confidence is an indication of how often the rule has been found to be true.
- Support is an indication of how frequently the itemset appears in the dataset.
- Lift is a measure of importance of a rule. If lift > 1, then items are dependent on each other. If lift < 1 then the items substitute each other.

The tags (used as the basket of rules) initially looked as follows, per talk:

```
[1] "children,creativity,culture,dance,education,parenting,teaching"
[2] "alternativeenergy,cars,climatechange,culture,environment,globalissues,science,sustainability,technology"
[3] "computers,entertainment,interfacedesign,media,music,performance,simplicity,software,technology"
[4] "macarthurgrant,activism,business,cities,environment,green,inequality,politics,pollution"
[5] "africa,asia,google,demo,economics,globaldevelopment,globalissues,health,math,statistics,visualizations"
[6] "business,culture,entertainment,goal-setting,motivation,potential,psychology"
```

These tags were parsed to identify which tags were the most frequently used.

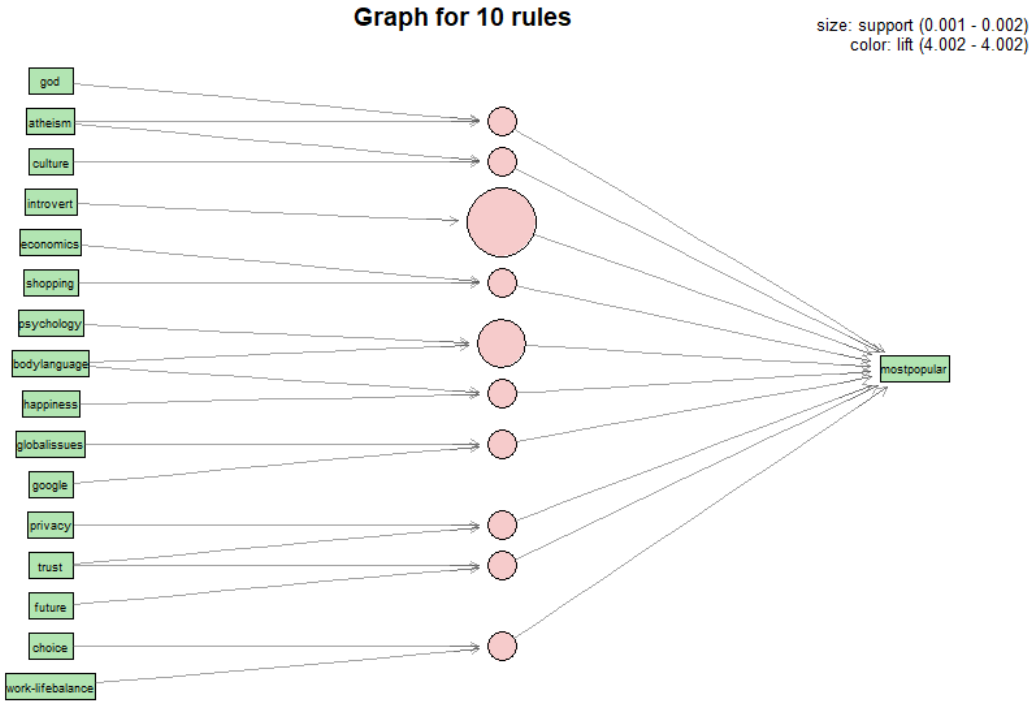


As part of this Association Rules mining exercise, two additional attributes per talk were appended to each basket. These attributes were (1) either **mostpopular** or **lesspopular** and (2) either **highlyrated** or **nothighlyrated**.

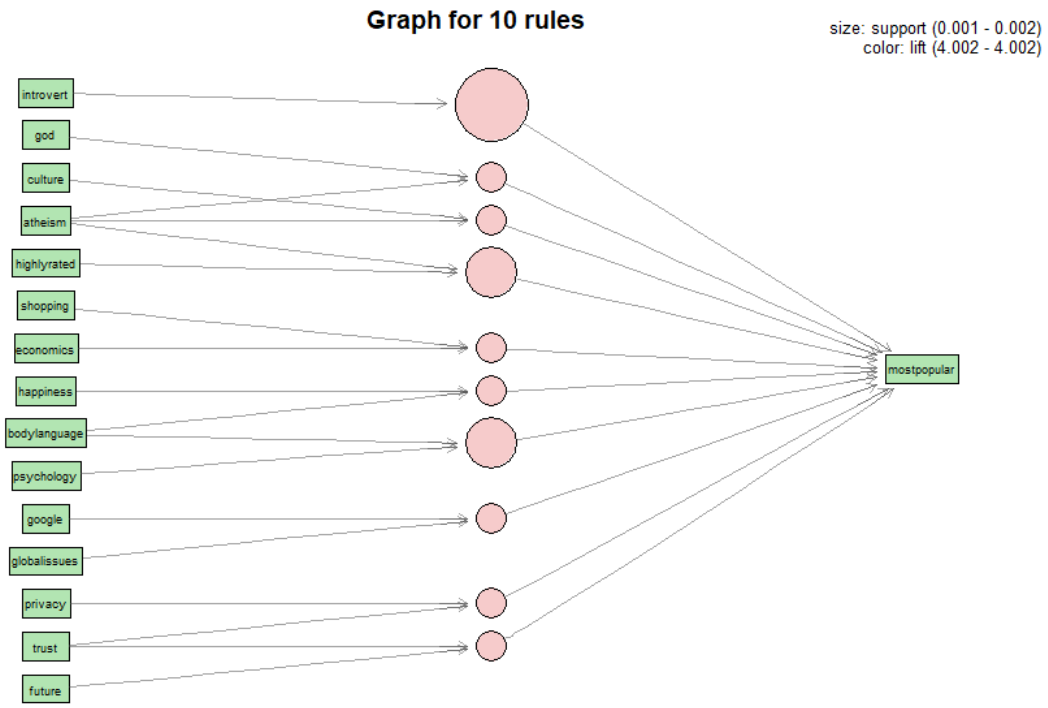
The eventual intention being to determine which of the rules from the tags appeared with which of the newly added behavioral attributes.
Four sets of Association Rules were run, to determine which of the tags appeared as the top tags for the four different rule sets. The four sets are as follows:

1. Rules where RHS contains **mostpopular** as well as **highlyrated**
2. Rules where RHS contains **mostpopular** as well as **nothighlyrated**
3. Rules where RHS contains **lesspopular** as well as **highlyrated**
4. Rules where RHS contains **lesspopular** as well as **nothighlyrated**

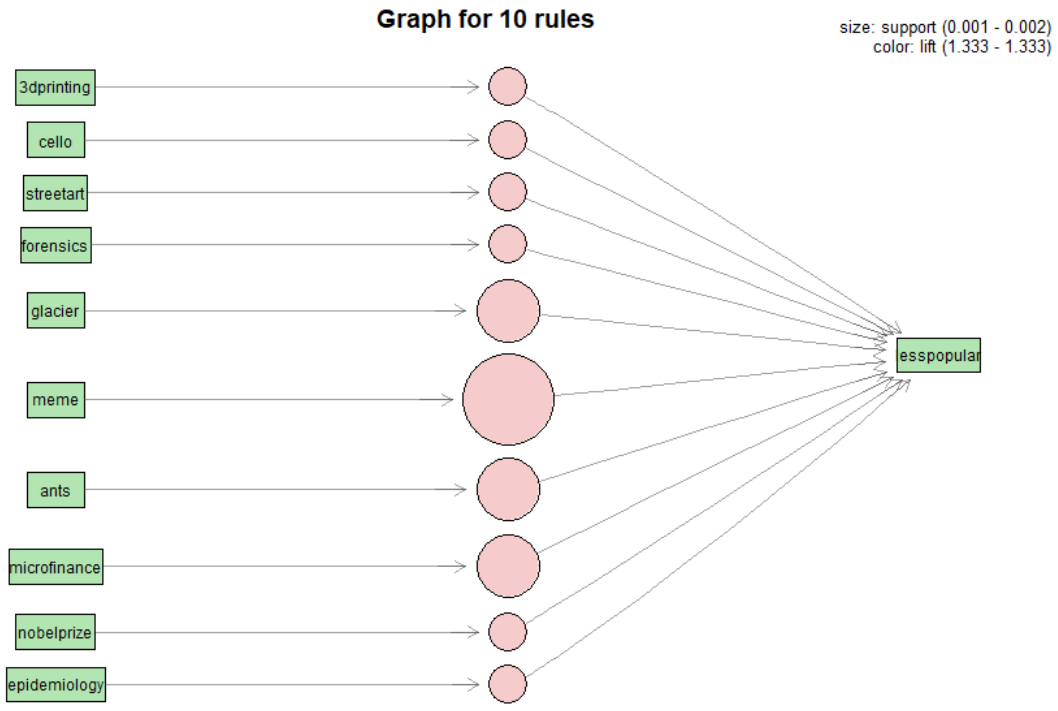
For Set 1 (**mostpopular** and **highlyrated**), the top 10 rules were extracted



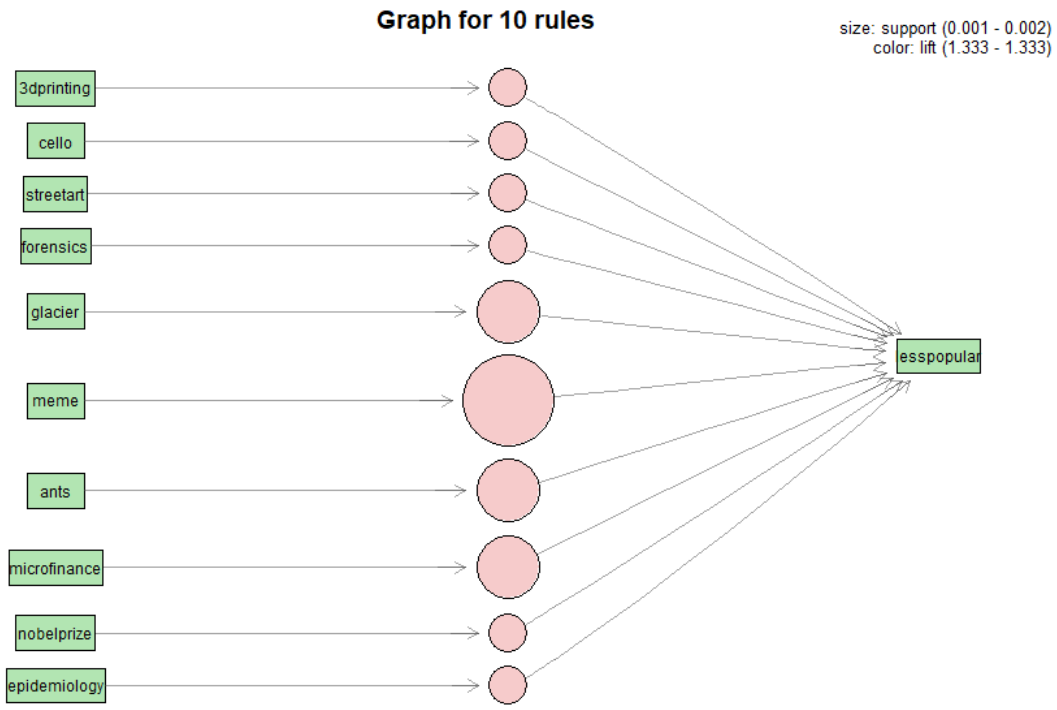
For Set 2 (**mostpopular** and **nothighlyrated**), the top 10 rules were extracted



For Set 3 (**lesspopular** and **highlyrated**), the top 10 rules were extracted



For Set 4 (**lesspopular** and **nothighlyrated**), the top 10 rules were extracted


Graph for 10 rules
size: support (0.001 - 0.002)
color: lift (1.333 - 1.333)
3dprinting
cello
streetart
forensics
glacier
meme
ants
microfinance
nobelprize
epidemiology
esspopular

Model 4: Decision Tree

Decision Tree Mining was run to predict both Popular talks as well as Highly Rated talks.

For predicting whether a talk is Highly Rated or not, an accuracy rating of 99.49% is possible. Output from the j48 model, with 10 fold validation, is as follows:

```
=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances        2540               99.4908 %
Incorrectly Classified Instances        13                0.5092 %
Kappa statistic                          0
Mean absolute error                      0.0101
Root mean squared error                  0.0712
Relative absolute error                 95.9651 %
Root relative squared error             99.9982 %
Total Number of Instances             2553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area
PRC Area  Class
                 0.000    0.000    ?          0.000    ?          ?        0.419
0.004     0
                 1.000    1.000    0.995      1.000    0.997      ?        0.419
0.994     1
Weighted Avg.    0.995    0.995    ?          0.995    ?          ?        0.419
0.989

=== Confusion Matrix ===

    a    b   <-- classified as
    0   13 |    a = 0
    0 2540 |    b = 1
```

For predicting whether a talk is Popular or not, an accuracy rating of 75.00% was achieved. Output from the j48 model, with 10-fold validation, is as follows:

```
=== 10 Fold Cross Validation ===

=== Summary ===

Correctly Classified Instances        1915               75.0098 %
Incorrectly Classified Instances       638               24.9902 %
Kappa statistic                          0
Mean absolute error                      0.3749
Root mean squared error                  0.433
Relative absolute error                 99.971  %
Root relative squared error            100      %
Total Number of Instances              2553

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area
PRC Area  Class
                 1.000    1.000    0.750      1.000    0.857      ?        0.498
0.749     0
                 0.000    0.000    ?          0.000    ?          ?        0.498
0.249     1
Weighted Avg.    0.750    0.750    ?          0.750    ?          ?        0.498
0.624

=== Confusion Matrix ===

    a    b   <-- classified as
 1915    0 |    a = 0
  638    0 |    b = 1
```

Model 5: Support Vector Machines

SVM is a linear classifier and can solve linear separable and non linear separable problems. It relies on decision boundaries, or hyperplane, to separate two categories. The goal of the hyperplane is to maximize the margin between the two categories. The two main things to look for in SVM is setting a large margin and having a low misclassification rate. The most important parameters are C, or the cost of misclassification, and the type of kernel used (linear, polynomial, radial, etc). A larger value for C means a smaller margin. With the change in C comes with the change in margin. SVM is useful since it is flexible and scalable, but it has parameters that need to be tweaked in order to find the best setting for predictions. In this case, SVM will be used to predict whether a talk is popular or high-rated. Both the normalized and original dataset will be used for the SVM models.

Predicting “Popularity” with SVM:

Tuning was done on the normalized dataset with linear, radial, and polynomial kernels in order to find the best C-value and kernel for SVM when predicting “popularity”. In this case, C=100 was found to be the best option. To determine the best kernel in general, the linear kernel was selected due to it having the lowest amount of support vectors (55) compared to the other two kernels. The lower the amount of support vectors, the less complicated the model. After the tuning is done, **C=100** and **linear kernel** will be used for the models.

Feature selection was also done in SVM in order to find out the important features in predicting the response variable of “popularity.” First, the features of the non-normalized training set were evaluated. A regression tree function was also used to compare the attributes as a way to compare important features across a different method:

```
Regression tree case (popularity, non-normalized dataset):
       views positivecount     highrated   fascinating   informative      ingenius
    697.2368      386.0238      374.0541      243.8832      215.4551      215.4551

SVM model case (popularity, non-normalized dataset):
          views     positivecount       fascinating       persuasive        inspiring       longwinded      jawdropping
      116.9200669        69.3378515        64.4920210        56.0195302        56.0195302        56.0195302        55.9132409
      informative          ingenius             funny          comments       courageous         beautiful         languages
       53.0131147        53.0131147        46.2516882        44.9225997        40.7754265        40.0800429        36.3614305
        confusing        highrated1         obnoxious     negativecount     unconvincing    published_date              year
       29.5152238        25.8616583        20.4788573        17.6680319        15.2987930         9.9459235         9.7353241
                id TransHasLaughter1          num_tags          duration            daySat         sentiment TransHasApplause1
        7.2935651         5.3067497         4.5012418         3.2873358         2.9156940         2.0492959         1.9186036
           dayFri             dayMon    TransHasVideo1            daySun             month             dayTue         film_date
        1.8336148         1.4310765         1.4224397         1.3726554         1.1969389         1.1592885         0.9284921
     filming_year    filming_month            dayThu            dayWed    TransHasMusic1
        0.8841316         0.8096395         0.4039707         0.1583935         0.0000000
```

From this analysis, views, comments, positivecount, and highrated are among the important variables along with positive rating categories such as “fascinating”,

“informative”, and “ingenius”. This makes sense that positive ratings tend to result in videos that are popular.

The same features evaluation for popularity is done with the normalized version of the dataset:

```
Regression tree case (popularity, normalized dataset):
       views positivecount     highrated   fascinating   informative      ingenius
    694.2497      361.3697      341.8768      235.4151      209.9243      209.9243

SVM model case (popularity, normalized dataset):
       views positivecount   fascinating    persuasive     inspiring    longwinded   informative      ingenius      comments
 111.8848377    57.4459759    52.9152968    49.9069167    49.9069167    49.9069167    47.9714684    47.9714684    44.1204561
       funny   jawdropping    courageous     languages     beautiful  unconvincing    highrated0    highrated1 negativecount
  40.8246507    39.5983023    37.6260424    30.1920674    29.9545369    20.4291601    15.9076830    15.9076830    10.7707502
    obnoxious     num_tags     sentiment     confusing      duration
  10.0906689     5.8026969     5.4420392     3.3681799     0.7530752
```

Although there not as many variables in the normalized dataset, the result is very similar. Notably, “duration” has decreased from 3.29 to 0.75 - it is now the least important variable in the normalized dataset.

Predicting High-Rated with SVM:

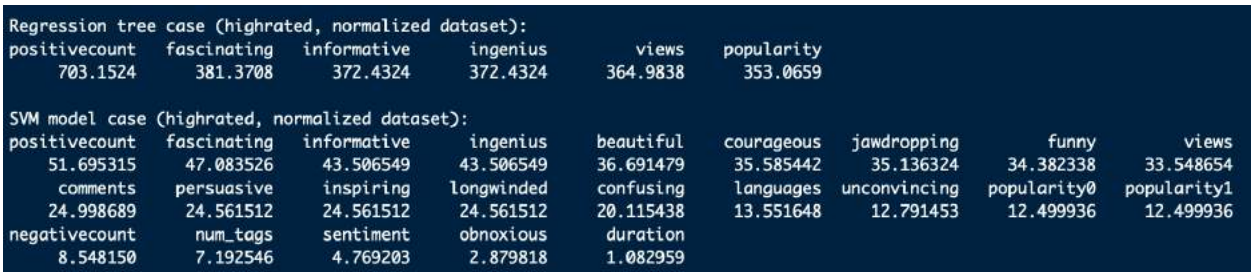
Tuning was done with linear, radial, and polynomial kernels in order to find the best C-value and kernel for SVM when predicting high-rated talks. In this case, C=0.01 was found to be the best option. To determine the best kernel in general, the linear kernel was selected due to it having the lowest amount of Support Vectors (10) compared to the other two kernels. The lower the amount of Support Vectors, the less complicated the model. After the tuning is done, **C=0.01** and **linear kernel** will be used for the models.

Feature selection is also done on the non-normalized original dataset in order to find the features that are useful in predicting high-rated talks:

```
Regression tree case (highrated, non-normalized dataset):
positivecount         views     popularity   informative      ingenius    fascinating
     695.2476      386.5816       371.5978       364.1059      364.1059       349.1222

SVM model case (highrated, non-normalized dataset):
    positivecount       fascinating       informative          ingenius        courageous       jawdropping             views
      78.48314804       67.55399698       65.30057692       65.30057692       58.68362111       57.14501665       55.34530980
        beautiful             funny          comments         persuasive          inspiring        longwinded       unconvincing
      52.24603930       47.86833854       39.00245638       35.44444258       35.44444258       35.44444258       34.66851417
      popularity1         confusing         languages          obnoxious TransHasLaughter1          num_tags    published_date
      22.25400633       20.03592212       19.90117439       19.20600150        9.35405944        9.35206858        7.68209323
             year      filming_year                id         film_date          duration             month         sentiment
       7.42425546        6.61850186        6.43618167        6.38547158        5.34638663        3.18631018        3.08491251
           dayThu     filming_month            daySat            dayWed            dayMon TransHasApplause1    TransHasVideo1
       3.05284957        2.53316390        2.37794869        2.20909040        1.77870573        0.94851478        0.67869208
           dayFri            dayTue            daySun     negativecount    TransHasMusic1
       0.44772682        0.11348355        0.08972706        0.06680592        0.00000000
```

Positivecount, popularity, views and positive rating categories such as “fascinating” and “ingenius” are at the top of the ranking of important variables.

The same feature selection is done on the normalized dataset to see if this trend is continued:



```
Regression tree case (highrated, normalized dataset):
positivecount   fascinating   informative      ingenius         views    popularity
     703.1524      381.3708      372.4324      372.4324      364.9838      353.0659

SVM model case (highrated, normalized dataset):
positivecount   fascinating   informative      ingenius     beautiful    courageous   jawdropping         funny         views
    51.695315     47.083526     43.506549     43.506549     36.691479     35.585442     35.136324     34.382338     33.548654
     comments    persuasive      inspiring    longwinded     confusing     languages   unconvincing   popularity0   popularity1
    24.998689     24.561512     24.561512     24.561512     20.115438     13.551648     12.791453     12.499936     12.499936
negativecount      num_tags     sentiment      obnoxious      duration
     8.548150      7.192546      4.769203      2.879818      1.082959
```

Comments had a big decrease from 39.00 down to 25.00. Many of the other variables also had a big drop. It seems that the feature comparison lowered the rating for the normalized version for the SVM model comparison, but the regression tree numbers stayed mostly the same. Due to this drop, a SVM model based on the non-normalized data was decided to be included in the results as well.

Results (Popularity)

SVM Model 1

In R, a SVM model is built to run on the training data (train3) created from the normalized dataset. Train3 was broken up into subgroups for the popularity label. The popularity column is the response variable and the rest below are the predictor variables:

```
 [1] "views"        "comments"     "duration"     "languages"    "num_tags"     "funny"        "beautiful"    "confusing"
 [9] "jawdropping"  "informative"  "ingenius"     "obnoxious"    "fascinating"  "unconvincing" "courageous"   "persuasive"
[17] "inspiring"    "longwinded"   "positivecount" "negativecount" "sentiment"   "highrated"
```

The model with C=100 and linear kernel ends up having 277 support vectors. The result is very good with a **96.43%** accuracy:

```
SVM model for predicting Popularity: 96.43%:

Call:
svm(formula = popularity ~ ., data = train3, kernel = "linear", cost = 100, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  100
      gamma:  0.04347826

Number of Support Vectors:  277
```

The confusion matrix confirms that the model is very accurate at classifying a talk as popular or not:

```
PredPopularity   0   1
             0 462  21
             1   1 133
[1] 0.9643436
```

Other kernels were attempted such as the radial kernel with an accuracy of 92.06% and the polynomial kernel with an accuracy of 89.30%. The 96.43% with the linear kernel ultimately proved the best option for the normalized dataset.

SVM Model 2 was also built using the non-normalized dataset for comparison purposes yet was only 65.64% accurate with the linear kernel at predicting popularity. Oddly enough, the radial kernel was the best kernel with the non-normalized dataset with a 74.88% accuracy.

Results (High-Rated)

SVM Model 3

In R, a SVM model is built to run on the training data (train4) created from the normalized dataset. Train4 was broken up into subgroups for the high-rated label. The high-rated column is the response variable and the rest below are the predictor variables:

```
 [1] "views"        "comments"     "duration"     "languages"    "num_tags"     "funny"        "beautiful"    "confusing"
 [9] "jawdropping"  "informative"  "ingenius"     "obnoxious"    "fascinating"  "unconvincing" "courageous"   "persuasive"
[17] "inspiring"    "longwinded"   "positivecount" "negativecount" "sentiment"   "popularity"
```

The model with C=100, and linear kernel ends up having 164 support vectors. The result is very good with a **97.41%** accuracy:

```
SVM model for predicting High-rated: 97.41%:

Call:
svm(formula = highrated ~ ., data = train4, kernel = "linear", cost = 100, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  100
      gamma:  0.04347826

Number of Support Vectors:  164
```

The confusion matrix confirms that the model is very accurate at classifying a talk as high-rated or not:

```
PredHighRated   0   1
            0 447   9
            1   7 154
[1] 0.9740681
```

Other kernels were attempted such as the radial kernel with an accuracy of 95.30%, and the polynomial kernel with an accuracy of 86.87%. The 97.41% with the linear kernel ultimately proved the best option for the normalized dataset.

SVM Model 4

In R, another SVM model is built to run on the training data (train2) created from the non-normalized dataset. Train2 has 35 variables in it compared to the normalized dataset which has 23 variables. Below are the variables included in the training set as predictor variables for the response variable of high-rated:

```
 [1] "id"               "views"            "comments"         "duration"         "languages"        "published_date"
 [7] "month"            "year"             "day"              "film_date"        "filming_month"    "filming_year"
[13] "num_tags"         "funny"            "beautiful"        "confusing"        "jawdropping"      "informative"
[19] "ingenius"         "obnoxious"        "fascinating"      "unconvincing"     "courageous"       "persuasive"
[25] "inspiring"        "longwinded"       "positivecount"    "negativecount"    "sentiment"        "TransHasVideo"
[31] "TransHasApplause" "TransHasMusic"    "TransHasLaughter" "highrated"        "popularity"
```

The model with C=100, and linear kernel ends up having 39 support vectors. The 39 support vectors is a big drop from the 164 support vectors of the previously normalized model. Not only does this mean the model is less complex, but the resulting accuracy of **98.05%** ends up surpassing the 97.41% accuracy from before:

```
SVM model for predicting High-rated: 98.05% (non-normalized dataset):

WARNING: reaching max number of iterations

Call:
svm(formula = highrated ~ ., data = train2, kernel = "linear", cost = 100, scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  100
      gamma:  0.025

Number of Support Vectors:  39
```

The confusion matrix confirms that the model is very accurate at classifying a talk as high-rated or not:

```
PredHighRated   0   1
            0 462   9
            1   3 143
[1] 0.9805511
```

Other kernels were attempted such as the radial kernel with an accuracy of 75.52%, and the polynomial kernel with an accuracy of 33.06%. The other kernels were noticeably worse with the non-normalized dataset. The linear kernel once again was proven to be the best option regardless of whether or not the data was normalized.

Model 6: Naïve Bayes

Naive Bayes is a linear classifier like SVM. Since in Naïve Bayes features are only compared to each other, normalization is often not necessary. Regardless, different models will be run against the normalized and non-normalized training sets to see which has the best accuracy. Naïve Bayes is run on the default setting since there are no parameters to tune.

Results (Popularity)

NB Model 1

In R, a Naïve Bayes model is built to run on the training data (train) created from the **original non-normalized dataset**. This dataset has all original 43 variables. The popularity column is the response variable and the rest below are the predictor variables:

```
 [1] "id"               "title"            "main_speaker"      "speaker_occupation" "views"            "comments"
 [7] "duration"         "languages"        "description"       "transcript"         "event"            "published_date"
[13] "month"            "year"             "day"               "film_date"          "filming_month"    "filming_year"
[19] "tags"             "num_tags"         "tagscore"          "funny"              "beautiful"        "confusing"
[25] "jawdropping"      "informative"      "ingenius"          "obnoxious"          "fascinating"      "unconvincing"
[31] "courageous"       "persuasive"       "inspiring"         "longwinded"         "positivecount"    "negativecount"
[37] "sentiment"        "TransHasVideo"    "TransHasApplause"  "TransHasMusic"      "TransHasLaughter" "highrated"
[43] "popularity"
```

The NB classifier is created and used to predict against the test set. The result is a confusion matrix and an accuracy of **88.33%**:

```
NB popularity model (non-normalized dataset):
Confusion Matrix and Statistics


PredPopularity   0   1
             0 438  44
             1  28 107

               Accuracy : 0.8833
                 95% CI : (0.8553, 0.9076)
    No Information Rate : 0.7553
    P-Value [Acc > NIR] : 1.035e-15

                  Kappa : 0.6726

 Mcnemar's Test P-Value : 0.0771

            Sensitivity : 0.9399
            Specificity : 0.7086
         Pos Pred Value : 0.9087
         Neg Pred Value : 0.7926
             Prevalence : 0.7553
         Detection Rate : 0.7099
   Detection Prevalence : 0.7812
      Balanced Accuracy : 0.8243

       'Positive' Class : 0
```

Figure 20, Naive Bayes "Popularity" Confusion Matrix/Output for Non-Normalized Data

NB Model 2

A NB classifier is also created to run against the **normalized version of the dataset** with these predictor variables below for the popularity response variable:

```
 [1] "views"        "comments"     "duration"     "languages"    "num_tags"     "funny"        "beautiful"    "confusing"
 [9] "jawdropping"  "informative"  "ingenius"     "obnoxious"    "fascinating"  "unconvincing" "courageous"   "persuasive"
[17] "inspiring"    "longwinded"   "positivecount" "negativecount" "sentiment"   "highrated"
```

The model is used to predict against the test set and ends up having a slightly higher accuracy of **90.11%:**

```
NB popularity model (normalized dataset):
Confusion Matrix and Statistics


PredPopularity   0   1
             0 445  43
             1  18 111

               Accuracy : 0.9011
                 95% CI : (0.8748, 0.9235)
    No Information Rate : 0.7504
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.721

 Mcnemar's Test P-Value : 0.00212

            Sensitivity : 0.9611
            Specificity : 0.7208
         Pos Pred Value : 0.9119
         Neg Pred Value : 0.8605
             Prevalence : 0.7504
         Detection Rate : 0.7212
   Detection Prevalence : 0.7909
      Balanced Accuracy : 0.8410

       'Positive' Class : 0
```

Figure 21, Naive Bayes "Popularity" Confusion Matrix/Output for Normalized Data

Although this accuracy is slightly higher, the reason it is higher may due to having to deal with less variables.

Results (High-Rated)

NB Model 3

In R, a Naïve Bayes model is built to run on the training data (train) created from the original **non-normalized dataset**. This dataset has all original 43 variables. The "highrated" column is the response variable and the rest below are the predictor variables:

```
 [1] "id"              "title"           "main_speaker"      "speaker_occupation" "views"           "comments"
 [7] "duration"        "languages"       "description"       "transcript"         "event"           "published_date"
[13] "month"           "year"            "day"               "film_date"          "filming_month"   "filming_year"
[19] "tags"            "num_tags"        "tagscore"          "funny"              "beautiful"       "confusing"
[25] "jawdropping"     "informative"     "ingenius"          "obnoxious"          "fascinating"     "unconvincing"
[31] "courageous"      "persuasive"      "inspiring"         "longwinded"         "positivecount"   "negativecount"
[37] "sentiment"       "TransHasVideo"   "TransHasApplause"  "TransHasMusic"      "TransHasLaughter" "highrated"
[43] "popularity"
```

The NB classifier is created and used to predict against the test set. The result is a confusion matrix and an accuracy of **94.00%**:

```
NB high-rated model (non-normalized dataset):
Confusion Matrix and Statistics

PredHighRated   0   1
            0 449  21
            1  16 131

               Accuracy : 0.94
                 95% CI : (0.9183, 0.9574)
    No Information Rate : 0.7536
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8367

 Mcnemar's Test P-Value : 0.5108

            Sensitivity : 0.9656
            Specificity : 0.8618
         Pos Pred Value : 0.9553
         Neg Pred Value : 0.8912
             Prevalence : 0.7536
         Detection Rate : 0.7277
   Detection Prevalence : 0.7618
      Balanced Accuracy : 0.9137

       'Positive' Class : 0
```

Figure 22, Naive Bayes "High-Rated" Confusion Matrix/Output for Non-Normalized Data

NB Model 4

A NB classifier is also created to run against the **normalized version of the dataset** (train4) with the predictor variables below for the high-rated response variable:

```
 [1] "views"        "comments"     "duration"     "languages"    "num_tags"     "funny"        "beautiful"    "confusing"
 [9] "jawdropping"  "informative"  "ingenius"     "obnoxious"    "fascinating"  "unconvincing" "courageous"   "persuasive"
[17] "inspiring"    "longwinded"   "positivecount" "negativecount" "sentiment"   "popularity"
```

The model is used to predict against the test set and ends up having a slightly lower accuracy than before with **93.52%:**

```
NB high-rated model (normalized dataset):
Confusion Matrix and Statistics


PredHighRated   0   1
            0 438  24
            1  16 139

               Accuracy : 0.9352
                 95% CI : (0.9128, 0.9533)
    No Information Rate : 0.7358
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8306

 Mcnemar's Test P-Value : 0.2684

            Sensitivity : 0.9648
            Specificity : 0.8528
         Pos Pred Value : 0.9481
         Neg Pred Value : 0.8968
             Prevalence : 0.7358
         Detection Rate : 0.7099
   Detection Prevalence : 0.7488
      Balanced Accuracy : 0.9088

       'Positive' Class : 0
```

Figure 23, Naive Bayes "High-Rated" Confusion Matrix/Output for Normalized Data

Model 7: Random Forest

Random Forest is an ensemble method based on the decision tree algorithm and creates subsets with decision trees that combine into a forest that makes prediction based on majority vote. For this analysis, the non-normalized version of the dataset will be used without the columns that have over 2000 levels to them. Random Forest only allows factor variables with a maximum of 32 levels. The variables removed in this smaller dataset include main_speaker, tagscore, tags, title, transcript, speaker_occupation, event, and description. First, variable importance is done in Random Forest.

Predicting “Popularity” with Random Forest:

Important features for popularity were analyzed with the default Random Forest fit model:

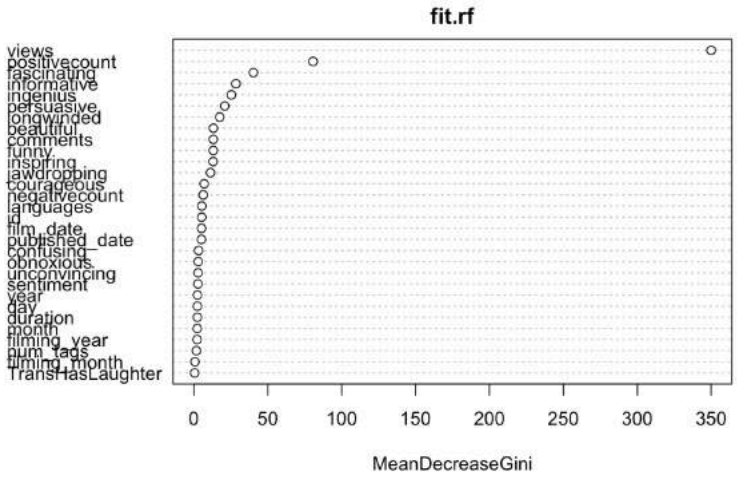


Figure 24, Random Forest Popularity Variable Importance Plot

Predicting High-Rated with Random Forest:

Important features for high-rated talks were analyzed with the default Random Forest fit model:

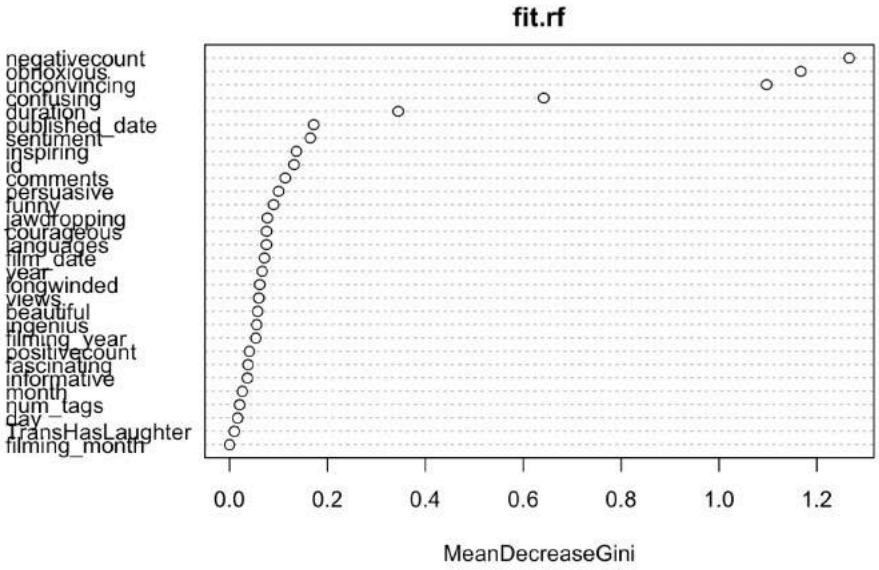


Figure 25, Random Forest High-Rated Variable Importance Plot

The importance of each variable when predicting the high-rated talks was also done:

	0	1	MeanDecreaseAccuracy	MeanDecreaseGini
id	0.000	1.003	1.003	1.39e-02
views	0.723	2.218	2.436	6.97e+01
comments	-1.003	-1.003	-1.003	1.06e+00
duration	0.000	0.000	0.000	0.00e+00
languages	0.000	-1.003	-1.003	4.80e-03
published_date	0.000	-1.003	-1.003	9.49e-02
month	0.000	0.000	0.000	1.27e-02
year	0.000	1.003	1.003	1.91e-02
day	0.000	0.000	0.000	0.00e+00
film_date	0.000	1.370	1.366	6.98e-02
filming_month	0.000	0.000	0.000	0.00e+00
filming_year	1.003	1.420	1.403	3.39e-02
num_tags	0.000	0.000	0.000	8.32e-03
funny	1.116	0.000	1.116	1.77e-01
beautiful	1.216	1.360	1.490	5.83e-01
confusing	-1.003	1.003	1.003	4.32e-02
jawdropping	0.000	1.005	0.999	2.02e-01
informative	3.063	0.603	3.085	2.10e+01
ingenius	2.543	-1.089	2.518	1.91e+01
obnoxious	0.000	0.000	0.000	0.00e+00
fascinating	3.252	0.909	3.268	1.24e+01
unconvincing	-1.003	-1.003	-1.003	1.41e-02
courageous	2.637	0.582	2.691	8.28e-01
persuasive	0.000	0.000	0.000	0.00e+00
inspiring	0.000	0.000	0.000	0.00e+00
longwinded	0.000	0.000	0.000	0.00e+00
positivecount	219.586	109.768	175.408	5.39e+02
negativecount	0.000	0.000	0.000	0.00e+00
sentiment	0.000	0.000	0.000	0.00e+00
TransHasVideo	0.000	0.000	0.000	0.00e+00
TransHasApplause	0.000	0.000	0.000	0.00e+00
TransHasMusic	0.000	0.000	0.000	0.00e+00
TransHasLaughter	0.000	0.000	0.000	0.00e+00
popularity	1.428	1.715	2.253	3.00e+01

A few variables stand out such as the “fascinating”, “courageous”, “ingenius”, and “informative” ratings. These ratings were more involved when assigning a talk a 0, or not high-rated. The variables that stuck out when assigning a talk as high-rated include the views and the “positivecount” variable. This is as expected since a higher rated TED talk will often get more views since it will be advertised and displayed more.

After getting a sense of what the default Random Forest model thinks of the important features, tuning is done for each of the response variables, popularity and high-rated. This tuning process allows for the best parameters to be chosen for each Random Forest model.

Results (Popularity)

RF Model 1

A non-normalized training set (train2) is used for predicting the popularity category with Random Forest. The predictor variables are below:

```
 [1] "id"              "views"           "comments"        "duration"        "languages"       "published_date"
 [7] "month"           "year"            "day"             "film_date"       "filming_month"   "filming_year"
[13] "num_tags"        "funny"           "beautiful"       "confusing"       "jawdropping"     "informative"
[19] "ingenius"        "obnoxious"       "fascinating"     "unconvincing"    "courageous"      "persuasive"
[25] "inspiring"       "longwinded"      "positivecount"   "negativecount"   "sentiment"       "TransHasVideo"
[31] "TransHasApplause" "TransHasMusic"  "TransHasLaughter" "highrated"      "popularity"
```

The process of tuning for popularity begins with the mtry parameter. First, a mtry selection formula is made with a cross-validation of 10-fold. The outputted number is 9:

```
Random Forest

1850 samples
  34 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1666, 1665, 1666, 1665, 1666, 1664, ...
Resampling results across tuning parameters:

  mtry  Accuracy  Kappa
   1    0.922     0.773
   2    0.969     0.915
   3    0.989     0.970
   4    0.998     0.996
   5    0.999     0.997
   6    0.999     0.999
   7    0.999     0.999
   8    0.999     0.999
   9    1.000     1.000
  10    1.000     1.000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 9.
```

This value of mtry=9 is used to find the best maxnodes value for the parameter in Random Forest. In this case, maxnodes=15 is best:

```
Call:
summary.resamples(object = results_mtry)

Models: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Number of resamples: 10

Accuracy
    Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
5  0.952   0.973  0.984 0.979   0.988    1    0
6  0.968   0.995  1.000 0.994   1.000    1    0
7  0.989   0.995  0.997 0.997   1.000    1    0
8  0.995   1.000  1.000 0.999   1.000    1    0
9  0.995   1.000  1.000 0.999   1.000    1    0
10 0.995   1.000  1.000 0.999   1.000    1    0
11 0.995   1.000  1.000 0.999   1.000    1    0
12 0.995   1.000  1.000 0.999   1.000    1    0
13 0.995   1.000  1.000 0.999   1.000    1    0
14 0.995   1.000  1.000 0.999   1.000    1    0
15 0.995   1.000  1.000 0.999   1.000    1    0

Kappa
    Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
5  0.875   0.926  0.958 0.944   0.968    1    0
6  0.911   0.986  1.000 0.984   1.000    1    0
7  0.972   0.986  0.993 0.991   1.000    1    0
8  0.985   1.000  1.000 0.999   1.000    1    0
9  0.985   1.000  1.000 0.999   1.000    1    0
10 0.985   1.000  1.000 0.999   1.000    1    0
11 0.985   1.000  1.000 0.999   1.000    1    0
12 0.985   1.000  1.000 0.999   1.000    1    0
13 0.985   1.000  1.000 0.999   1.000    1    0
14 0.985   1.000  1.000 0.999   1.000    1    0
15 0.985   1.000  1.000 0.999   1.000    1    0
```

Finally, the ntree size is found out with a formula:

```
Call:
summary.resamples(object = results_tree)

Models: 250, 300, 350, 400, 450, 500, 550, 600, 800, 1000, 2000
Number of resamples: 10

Accuracy
      Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
250  0.995       1      1 0.999       1    1    0
300  0.995       1      1 0.999       1    1    0
350  0.995       1      1 0.999       1    1    0
400  0.995       1      1 0.999       1    1    0
450  0.995       1      1 0.999       1    1    0
500  0.995       1      1 0.999       1    1    0
550  0.995       1      1 0.999       1    1    0
600  0.995       1      1 0.999       1    1    0
800  0.995       1      1 0.999       1    1    0
1000 0.995       1      1 0.999       1    1    0
2000 0.995       1      1 0.999       1    1    0

Kappa
      Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
250  0.986       1      1 0.999       1    1    0
300  0.986       1      1 0.999       1    1    0
350  0.986       1      1 0.999       1    1    0
400  0.986       1      1 0.999       1    1    0
450  0.986       1      1 0.999       1    1    0
500  0.986       1      1 0.999       1    1    0
550  0.986       1      1 0.999       1    1    0
600  0.986       1      1 0.999       1    1    0
800  0.986       1      1 0.999       1    1    0
1000 0.986       1      1 0.999       1    1    0
2000 0.986       1      1 0.999       1    1    0
```

It seems that Random Forest is perfect with predicting the popularity variable since the accuracy is 100% for every tree size tested.

The best parameters are now found:

- Maxnodes=15
- Mtry=9
- Ntree=200

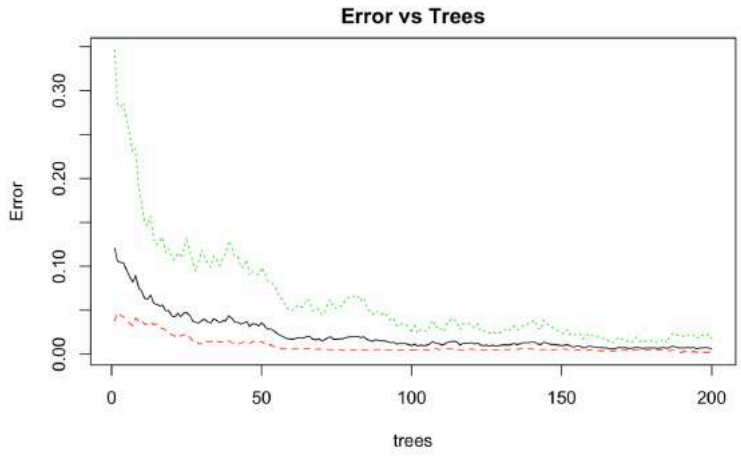
The best parameters are inputted into the Random Forest fit model and an accuracy of **99.80%** is achieved for predicting the popularity response variable:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 466   1
         1   0 150

               Accuracy : 0.998
                 95% CI : (0.991, 1)
    No Information Rate : 0.755
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.996

 Mcnemar's Test P-Value : 1

            Sensitivity : 1.000
            Specificity : 0.993
         Pos Pred Value : 0.998
         Neg Pred Value : 1.000
             Prevalence : 0.755
         Detection Rate : 0.755
   Detection Prevalence : 0.757
      Balanced Accuracy : 0.997

       'Positive' Class : 0
```

Figure 26, RF Confusion Matrix/Output for Popular Talks

The accuracy for the model is **near perfect** at predicting the popularity of TED talks.

An error rate vs tree size plot is also created with the model and shows how after a tree size of 50 or so, the error rate starts going down closer to 0 with more trees added:



Results (High-Rated)

RF Model 2

A non-normalized training set (train2) is used for predicting the high-rated category with Random Forest. The predictor variables are below:

```
 [1] "id"              "views"           "comments"        "duration"        "languages"       "published_date"
 [7] "month"           "year"            "day"             "film_date"       "filming_month"   "filming_year"
[13] "num_tags"        "funny"           "beautiful"       "confusing"       "jawdropping"     "informative"
[19] "ingenius"        "obnoxious"       "fascinating"     "unconvincing"    "courageous"      "persuasive"
[25] "inspiring"       "longwinded"      "positivecount"   "negativecount"   "sentiment"       "TransHasVideo"
[31] "TransHasApplause" "TransHasMusic"  "TransHasLaughter" "highrated"      "popularity"
```

The process of tuning for popularity begins with the mtry parameter. First, a mtry selection formula is made with a cross-validation of 10-fold. The outputted number is 6:

```
Random Forest

1850 samples
  34 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 1666, 1666, 1665, 1665, 1664, 1665, ...
Resampling results across tuning parameters:

  mtry  Accuracy  Kappa
   1    0.957     0.878
   2    0.994     0.984
   3    0.999     0.999
   4    0.999     0.997
   5    0.999     0.997
   6    1.000     1.000
   7    1.000     1.000
   8    1.000     1.000
   9    1.000     1.000
  10    1.000     1.000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 6.
```

This value of mtry=6 is used to find the best maxnodes value for the parameter in Random Forest. In this case, maxnodes=15 is best:

```
Call:
summary.resamples(object = results_mtry)

Models: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Number of resamples: 10

Accuracy
    Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
5  0.995       1      1 0.999       1    1    0
6  0.995       1      1 0.999       1    1    0
7  0.995       1      1 0.999       1    1    0
8  1.000       1      1 1.000       1    1    0
9  1.000       1      1 1.000       1    1    0
10 1.000       1      1 1.000       1    1    0
11 1.000       1      1 1.000       1    1    0
12 1.000       1      1 1.000       1    1    0
13 1.000       1      1 1.000       1    1    0
14 1.000       1      1 1.000       1    1    0
15 1.000       1      1 1.000       1    1    0

Kappa
    Min. 1st Qu. Median  Mean 3rd Qu. Max. NA's
5  0.985       1      1 0.997       1    1    0
6  0.985       1      1 0.999       1    1    0
7  0.985       1      1 0.999       1    1    0
8  1.000       1      1 1.000       1    1    0
9  1.000       1      1 1.000       1    1    0
10 1.000       1      1 1.000       1    1    0
11 1.000       1      1 1.000       1    1    0
12 1.000       1      1 1.000       1    1    0
13 1.000       1      1 1.000       1    1    0
14 1.000       1      1 1.000       1    1    0
15 1.000       1      1 1.000       1    1    0
```

Finally, the ntree size is found out with a formula:

```
Call:
summary.resamples(object = results_tree)

Models: 20, 30, 50, 100, 200, 300, 400, 500, 600, 1000, 2000
Number of resamples: 10

Accuracy
     Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
20      1       1      1    1       1    1    0
30      1       1      1    1       1    1    0
50      1       1      1    1       1    1    0
100     1       1      1    1       1    1    0
200     1       1      1    1       1    1    0
300     1       1      1    1       1    1    0
400     1       1      1    1       1    1    0
500     1       1      1    1       1    1    0
600     1       1      1    1       1    1    0
1000    1       1      1    1       1    1    0
2000    1       1      1    1       1    1    0

Kappa
     Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
20      1       1      1    1       1    1    0
30      1       1      1    1       1    1    0
50      1       1      1    1       1    1    0
100     1       1      1    1       1    1    0
200     1       1      1    1       1    1    0
300     1       1      1    1       1    1    0
400     1       1      1    1       1    1    0
500     1       1      1    1       1    1    0
600     1       1      1    1       1    1    0
1000    1       1      1    1       1    1    0
2000    1       1      1    1       1    1    0
```

It seems that Random Forest is perfect with predicting the high-rated variable since the accuracy is 100% for every tree size tested.

The best parameters are now found:

- Maxnodes=15
- Mtry=6
- Ntree=200

The parameters are inputted into the Random Forest fit model and an accuracy of **100%** is achieved for predicting the high-rated response variable:

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 465   0
         1   0 152

               Accuracy : 1
                 95% CI : (0.994, 1)
    No Information Rate : 0.754
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.000
            Specificity : 1.000
         Pos Pred Value : 1.000
         Neg Pred Value : 1.000
             Prevalence : 0.754
         Detection Rate : 0.754
   Detection Prevalence : 0.754
      Balanced Accuracy : 1.000

       'Positive' Class : 0
```

Figure 27, RF Confusion Matrix/Output for High-Rated Talks

The accuracy for the model is **perfect** at predicting a TED talk as high-rated

An error rate vs tree size plot is also created with the model and shows how after a tree size of 20 or so, the error rate starts going down closer to 0 with more trees added:

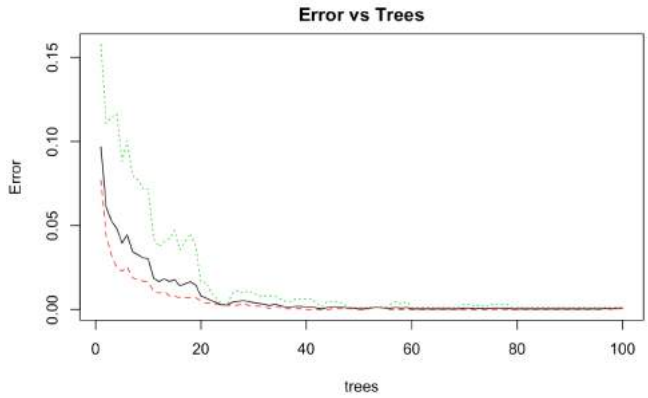


Figure 28, RF "High-Rated" Errors vs Trees Plot

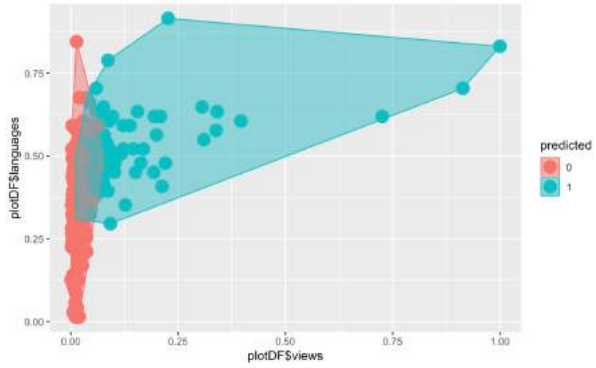
Model 8: k-Nearest Neighbors

K-NN is an instance-based learner, aka “lazy learner.” The algorithm does not learn a model but rather learns the training instances in the prediction phase. It also does not make any assumptions. The most important value is the **K-value**, or the number of neighbors that vote on the test example’s class. For this model, the K-value is achieved by getting the value of the square root of the number of rows of the dataframe. **K=50** is used for both models.

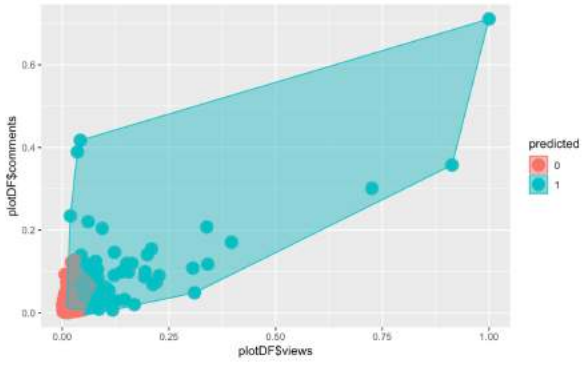
k-NN Model 1

Visualizing k-NN for Popularity:

A k-NN plot is visualized in relation to views and languages. The clusters of each are displayed as well as the boundary points. Views is chosen since it is the most relevant variable for classifying popularity:



A k-NN plot is also visualized in relation to views and comments:



Results (Popularity)

An accuracy of 88.99% is achieved for prediction of popular talks. A confusion matrix shows the breakdown:

```
kNN_fit   0   1
      0 450  32
      1  27 108
```

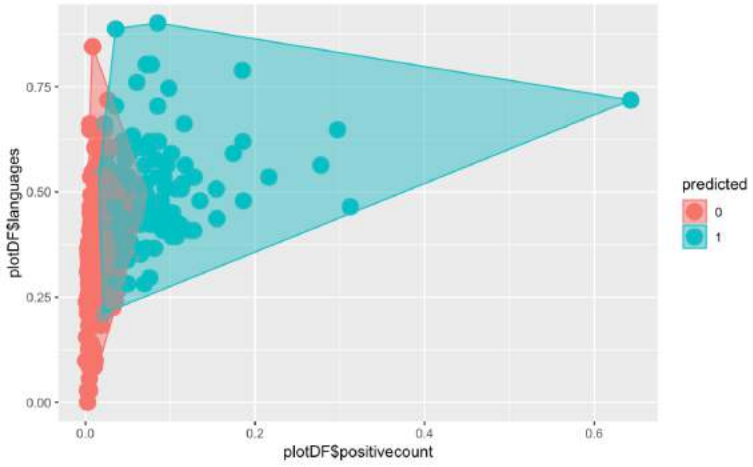
```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  617


                         | kNN_fit
test3_labels$popularity |         0 |         1 | Row Total |
------------------------|-----------|-----------|-----------|
                      0 |       450 |        27 |       477 |
                        |     0.943 |     0.057 |     0.773 |
                        |     0.934 |     0.200 |           |
                        |     0.729 |     0.044 |           |
------------------------|-----------|-----------|-----------|
                      1 |        32 |       108 |       140 |
                        |     0.229 |     0.771 |     0.227 |
                        |     0.066 |     0.800 |           |
                        |     0.052 |     0.175 |           |
------------------------|-----------|-----------|-----------|
           Column Total |       482 |       135 |       617 |
                        |     0.781 |     0.219 |           |
------------------------|-----------|-----------|-----------|
```

Figure 29, k-NN Popularity Model Output

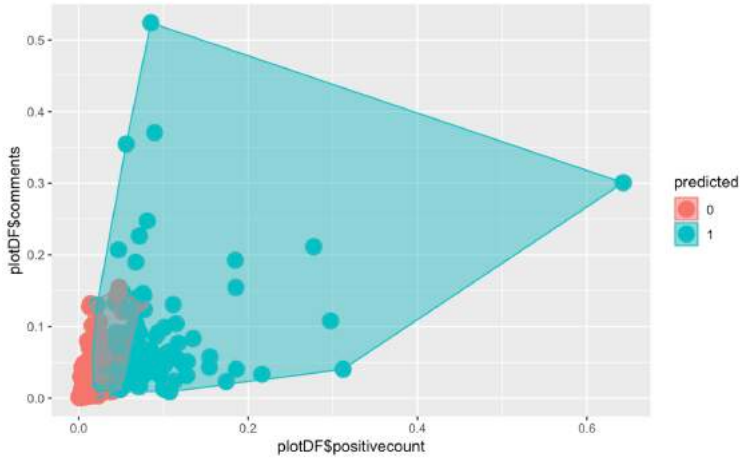
k-NN Model 2

Visualizing k-NN for High-Rated:

A k-NN plot is visualized in relation to count of positive ratings and languages. Positivecount is chosen as the main variable to compare against since it is the most relevant variable for classifying a talk as high-rated:



A k-NN plot is visualized in relation to count of positive ratings and comments:



Results (High-Rated)

An accuracy of **86.76%** is achieved for prediction of high-rated talks. A confusion matrix shows the breakdown:

kNN_fit	0	1
0	418	45
1	26	128

Cell Contents
N
N / Row Total
N / Col Total
N / Table Total

Total Observations in Table: 617

test4_labels$highrated	kNN_fit 0	kNN_fit 1	Row Total
0	417	27	444
	0.939	0.061	0.720
	0.901	0.175	
	0.676	0.044	
1	46	127	173
	0.266	0.734	0.280
	0.099	0.825	
	0.075	0.206	
Column Total	463	154	617
	0.750	0.250	

Figure 30, k-NN High-Rated Model Output

Results

Comparison of all Prediction Algorithms

The table below shows the accuracy achieved with each of the models that were run:

	Popularity	High-Rated
Decision trees	75.00%	99.49%
SVM	96.43%	97.41%
	74.88%	98.05%
Naïve Bayes	88.33%	94.00%
	90.11%	93.52%
Random Forest	99.80%	100.00%
kNN	88.99%	86.76%

As can be seen, the most accurate Model for deriving Popularity is the Random Forest Model and the most accurate Model for deriving High-Rated is also the Random Forest Model.

Analysis as well as result details for each of the Prediction models are giving in the corresponding model sections.

Elite talks vs Normal talks

An examination of the types of words in the 466 elite TED talks (both high-rated and popular) are also looked at:

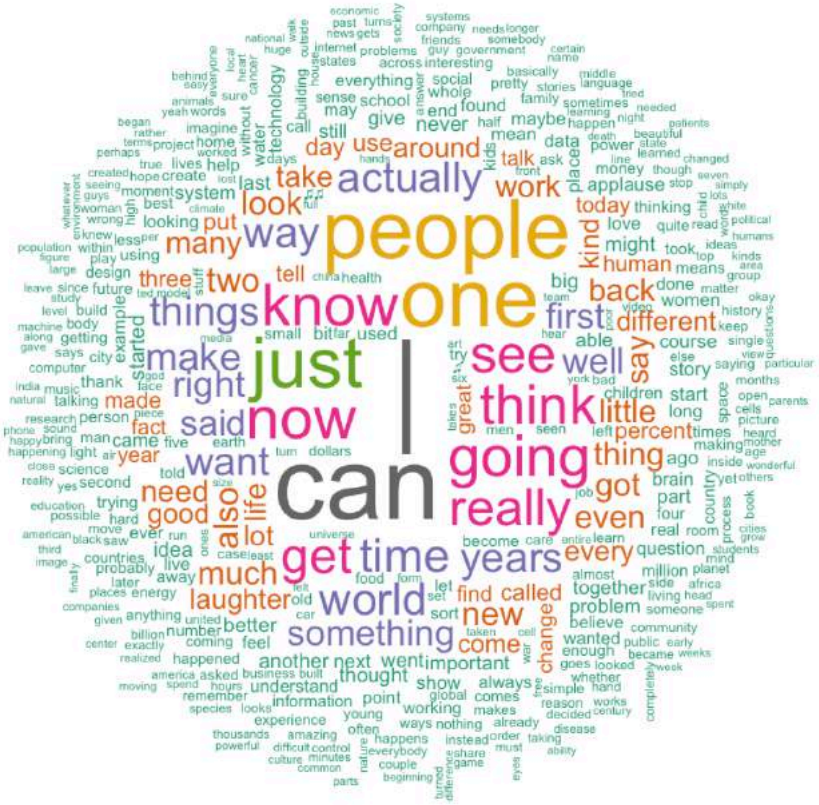
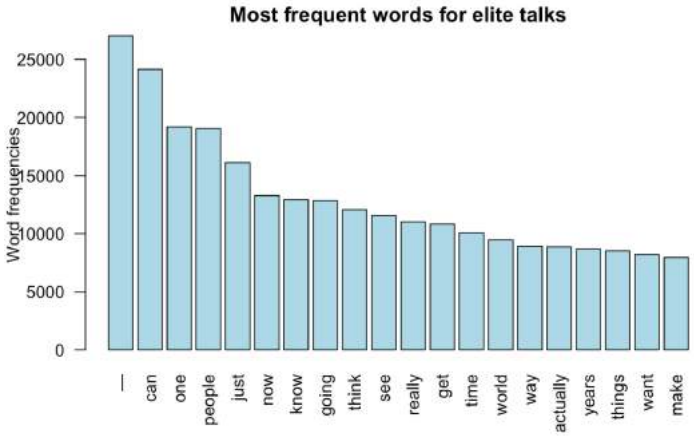


Figure 31, Word cloud for elite talks, minfreq=1500



The attributes of elite TED talks were compared against normal talks:

	Elite talks	**All talks**
Views (MIN)	1,736,329	155,895
Views (AVG)	4,587,641	1,740,295
Positive ratings count (MIN)	1,989	93
Positive ratings count (AVG)	42	5
Negative ratings count (MIN)	5151.298	1839.204
Negative ratings count (AVG)	354.807	179.521
Comments (AVG)	445.873	192.571
Languages (AVG)	35.481	28.291
Duration (AVG)	874.367 (14.6 minutes)	821.760 (13.7 minutes)
Number of Tags (AVG)	6.940	7.553

- An average elite TED talk will have **4,587,641** views, **5,151** positive ratings, **355** negative ratings, **446** comments, **35** languages, **7** tags, and a duration of **14.6** minutes.

- An average normal TED talk will have **1,740,295** views, **1,839** positive ratings, **179** negative ratings, **192** comments, **28** languages, **7** tags, and a duration of **13.7** minutes.

The wide gap between elite TED talks and normal TED talks is displayed with these comparisons.

- It is interesting to note that the elite TED talks are slightly longer than the normal TED talks by about a minute
- Tags does not seem to play a big factor as it is the same in both.
- Comments are over twice as large for elite talks.

Views and positive ratings are much larger between the two as expected.

Conclusions

General

Knowledge and curiosity about the world are fundamental human traits. It is therefore best to find ways to learn and communicate information in the best way possible. TED is at the forefront of this with its in-person and online speeches given by experts who aim to inspire, challenge, and make an impact on the world through the knowledge they share. The important features of what makes a TED talk resonate as popular or highly-rated can often be an enigma, but there are certain trends that can be useful to understand.

Talks that would be considered informative, inspiring, fascinating, ingenious, persuasive, jaw-dropping, courageous or funny are the ones that are most likely to be popular.

In addition, talks about God, culture, economics, shopping, psychology, body language, happiness, global issues, google, privacy, trust, future, choice or work life balance are also most likely to be popular.

To make more conclusive findings, it would be best in the future to analyze the audio of each TED talk and analyze the pitch and tonality of the speaker. Every talk is multifaceted and aspects such as the transcript of a talk, or the occupation of a speaker are not the best predictors alone of whether or not a talk will be successful. This report touched on one dimension of TED talks and therefore is not fully comprehensive. Although, the trends and patterns associated with popular, highly-rated, and elite-talks are definitely beneficial to understanding how to classify TED talks in the future based off multiple talk attributes highlighted in this report.

References

Data and supporting material was obtained from
https://www.kaggle.com/rounakbanik/ted-talks/