Exploring potentially usable predictors to assess the relative popularity of videos on the YouTube platform with similar characteristics

S. Pariente

08/07/2021

Summary

This is a capstone project for the HarvardX PH125.9x course. This project aims at looking into possible predictors to assess whether a video posted on the YouTube platform is likely to get a number of views higher than what would be expected.
After looking into what the main drivers behind YouTube video views are, using data extracted from the YouTube API for the purpose of this project, we will explore a set of potential predictors among the data generated on the platform.
Using this data exploration, we will select the elements most likely to be drivers or indicators of success among the videos in our sample. Several potential models will be used to determine if we can predict whether a given video will receive more views than would be expected.
Final results show that such a metric can be predicted with ca. 70% accuracy, using a combination of models, by using only non-visual data available on the YouTube platform.

Introduction

Context

Over the past two decades, social media has become an integral part of our lives. Whether active on any specific network or not, hard is it to deny the influence this relatively new field had now gathered, with over 50 % of the global eligible population having at least 1 social media account (source : Social Network Usage & Growth Statistics). Focusing on the United States, 72 % of adults say they use at least 1 social media site, up from 5 % in 2005, with younger generations more likely to be active than their elders (source : Social Media Fact Sheet).

Aside from their value as a vector of human interaction, social media platforms live and die through the data they generate, and how it can be used to generate a sustainable revenue stream.
In this project, we will be focusing on one of the largest social media platforms worldwide : YouTube. Founded in 2005, the platform has become a staple of the world wide web, now ranking as the second most visited website on the internet, and the second most-used social media platform (source : Digital 2021: Global Overview Report).
While not aiming to propose a complete forecasting tool, we will be exploring some of the key relevant metrics for the platform, and look into potential indicators that could be used to predict whether a given video is likely to be / remain successful, compared to other similar ones.

If usable patterns can be identified, they might be useful for content creators looking to increase viewership and subscription metrics, advertisers looking to maximize the reach of a specific campaign on the platform,

or the platform's marketing team's ability to more precisely adjust advertisement spot pricing on otherwise similar videos. This analysis can also be used as a first step towards a full prediction model for the number of views a video can expect to get.

Overview of the data used

Presentation of the data gathering process

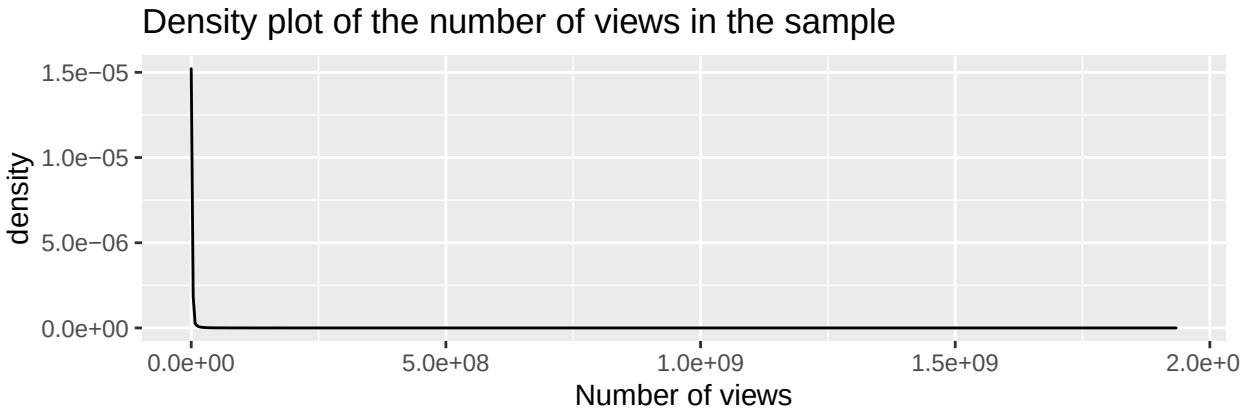
For the purpose of this project, information on a sample of 56,878 videos was gathered in June and July 2021. The data gathering relied on the YouTube Data API, using the following iterative process to try and approach randomness during the sampling :

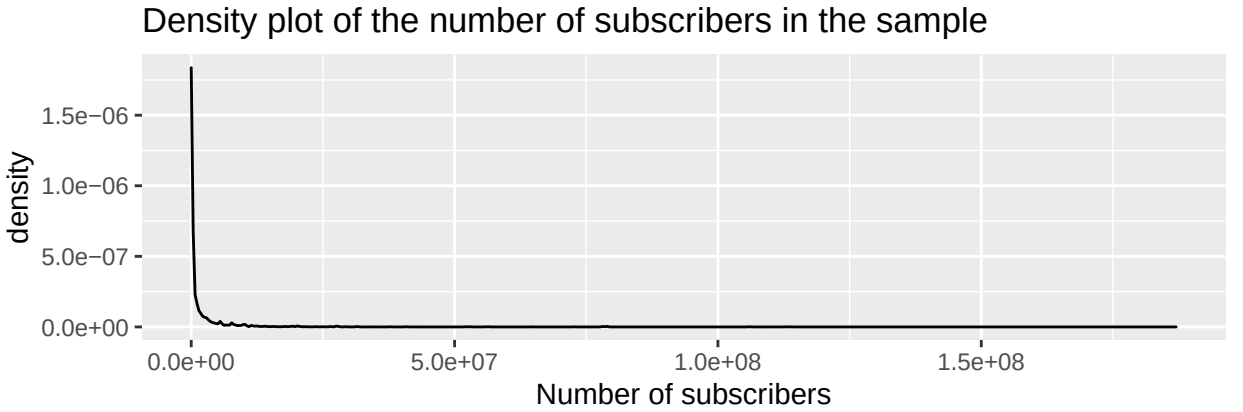
1. Generate a list of random word strings.

2. Feed this list to the aforementioned API, filtering for videos published after 01/01/2020, and select the first page of results for each string, representing up to 50 videos per string, from which the following identifiers were extracted:
    - ID of the video, a string of 11 characters, numbers, and signs
    - time at which the video was uploaded, from which the age of the video at the point of extraction, in days, was also computed,
    - ID of the content creator's channel, a string of 24 characters, numbers, and signs
3. For each of the random word strings, the list of video IDs was concatenated, and then used as input for a second interface of the API, to gather additional information on the video:
    - full title of the video
    - number of views, likes, dislikes, and comments on the video
    - list of topics associated with the video, which to the best of this writer's knowledge is assigned by a specific YouTube algorithm
    - full description of the video, made by its creator
    - ID of the video category, assigned by the creator
4. Similarly, for each of the random word strings, the list of channel IDs was concatenated, and then used as input for a third interface of the API, to gather additional data on the channel:
    - total number of subscribers
    - total number of videos
    - list of topics associated with the channel, assigned similarly to the videos'
5. The data thus gathered was then assembled into a single data set, from which observations with missing data, represented by NAs, was filtered out, for wrangling and exploration.

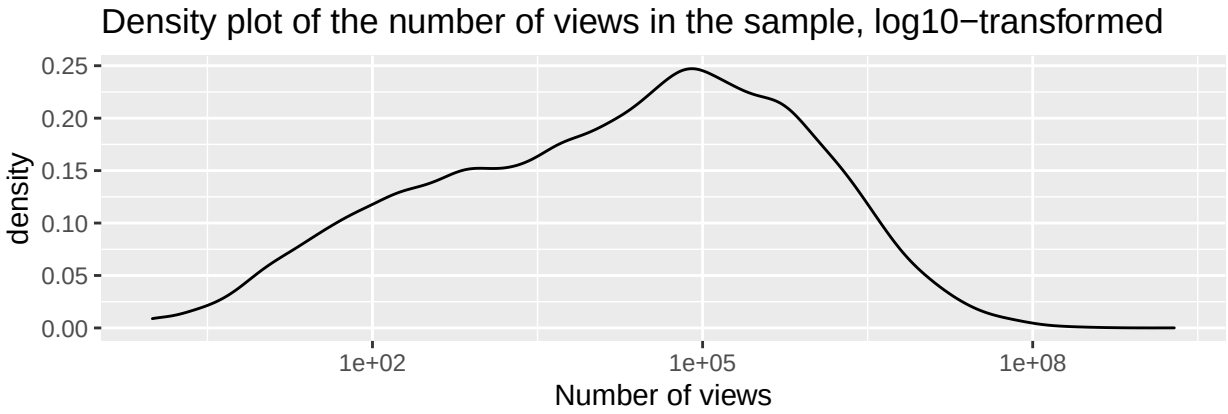
Preliminary data exploration

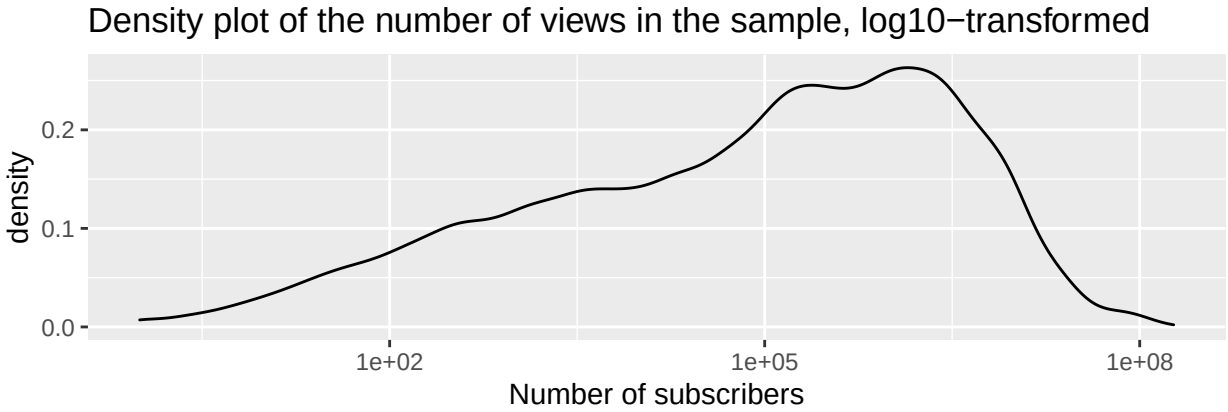
Every social media platform has its own measure of success. For YouTube, commonly accepted metrics are focused around the number of views for videos, and the number of subscribers for channels.
From the data set previously assembled, we will start by looking at the density plots for these two metrics over our full data set.





We quickly notice that both metrics are not normally distributed, with extremely large values being reported, compared to the mean. We can however look at a log10 transformation of the density plots :





This allows us to better glance at one of the first challenges we may be facing : the distribution of views and subscribers, after log10 transformation, does not closely follow a normal distribution. We can make two potential hypotheses to explain the distribution in our sample : either the overall video and channel base has a distribution similar to the one we have gathered from true random sampling process, or the actual distribution is a different one, and the observed shape results from a non-truly random sampling approach. With limited information available on either the overall data or the search algorithm itself, we will carefully theorize that our data set is likely skewed due to only looking at the first 50 results in the initial API call. These results are given by the sophisticated YouTube search algorithm, which, aside from looking for videos close to the search words entered, would also seem to favor videos with more views and / or channels with a higher amount of subscribers than what could be expected in a random sampling.
Nevertheless, we should consider two elements here before continuing our analysis :

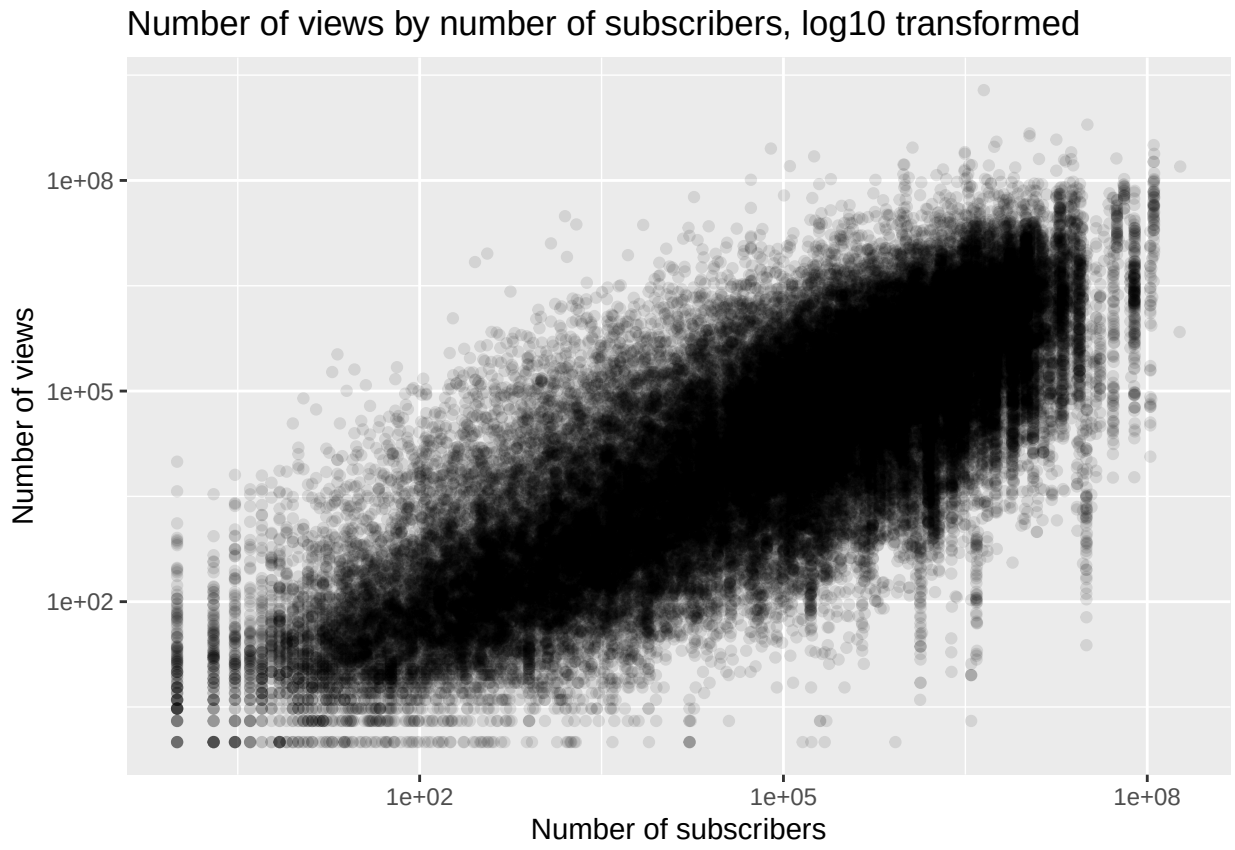
- As mentioned in the introduction, we do not aim to create a model able to fully forecast the number of views for a given video. Instead, we will focus on trying to separate videos that over-perform from videos that under-perform, under similar circumstances. This will require additional steps, after which we will revisit the distribution of our data.

- To make the model usable, we may want to only look at videos that are likely to be viewed. According to a 2021 digital marketing study on the Google search engine, *"71.33% of searches resulted in a page 1 Google organic click"*, which contains only 10 items on average, *"while pages 2 and 3 only get 5.59% of clicks"*. While exact behavior may be different when browsing a more specialized website such as YouTube, we can expect that the large majority of views are generated on the first 50 results.

To summarize, the data we extracted does not follow a normal distribution across these two metrics, but should nevertheless be reflective of the videos that are most likely to be viewed by a human browsing the

website. Moreover, considering we will not be focusing on this raw, non-normally distributed data for our final analysis, we will continue with our analysis.

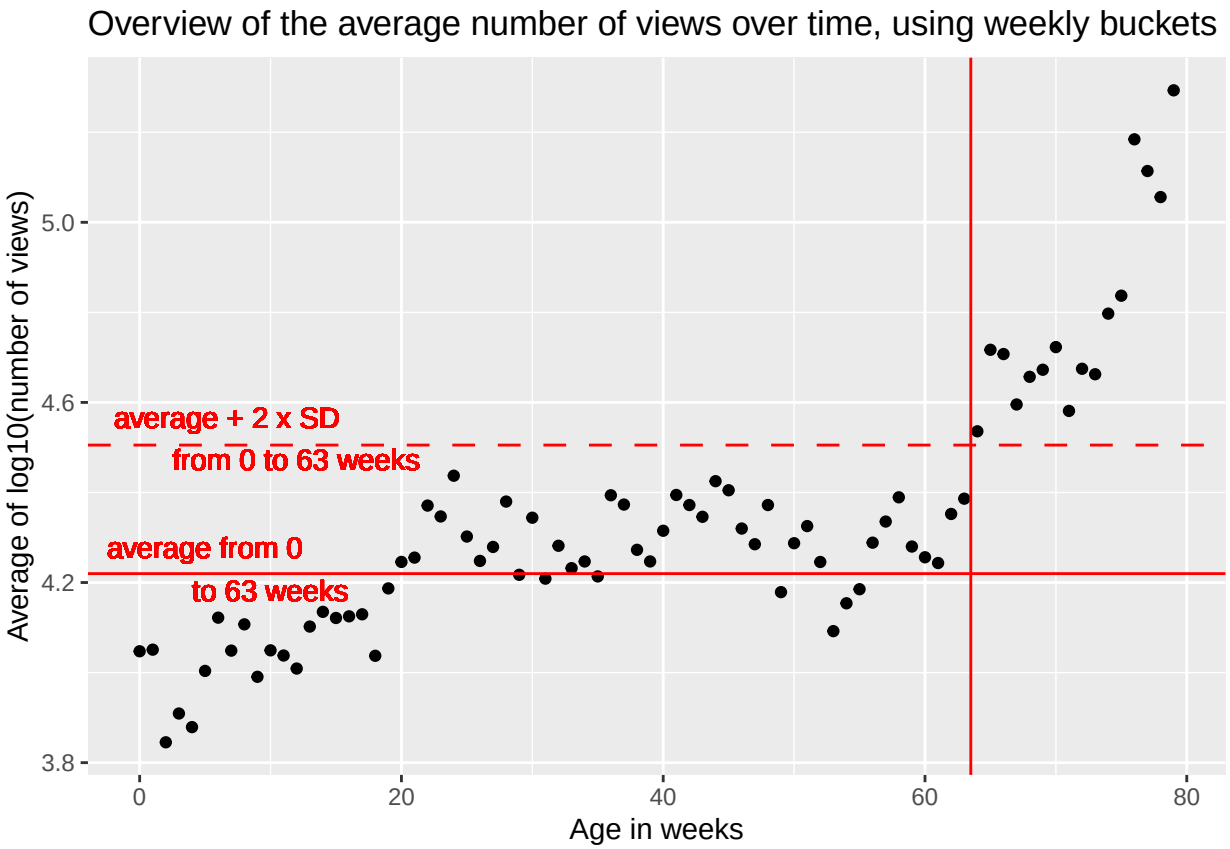
Another characteristic we may be noticing is that both density curves seem to have a similar shape, possibly indicating some correlation between the two. We can see this more clearly when plotting both items against one another, as such:



We can infer from the shape of the plot a correlation between the two metrics, possibly of a linear nature. A strong link between the two makes sense from a theoretical standpoint: a channel with more subscribers, that are kept aware of new videos being posted and / or are actively looking for more videos from this specific channel, is more likely to get a higher number of views on new videos than a channel with fewer subscribers. Moving forwards, we will be using log10-transformed data for the number of views and subscribers, and hence will add these metrics to our data table.

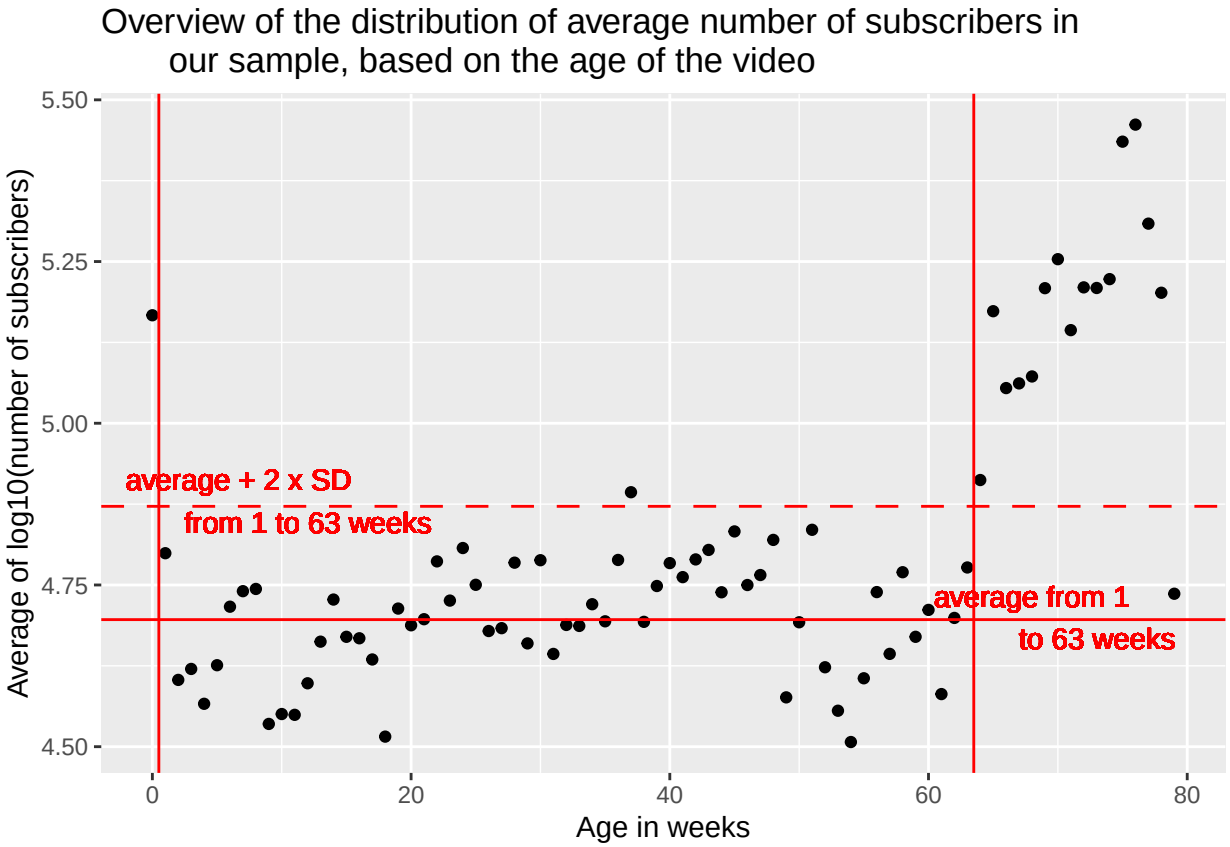
Doing so requires filtering out observations where either the number of views or subscribers is equal to 0, removing 380 observations from our data.

With our new sample, we can explore another relationship we would expect to observe: views as a function of the age of the video. To simplify our analysis, we can group the data by weekly buckets, starting at 0, and look at the mean of log10(views) for each bucket.



From the shape of the plot, we can say that the number of views tends to increase over the first 20 or so weeks, before reaching a plateau where marginal growth is more limited. This makes sense from a theoretical standpoint when looking at the life cycle of a video: new videos are initially promoted by the content creator, and immediately visible on the subscribers' feed, which contributes to the initial momentum, but is progressively replaced by newer videos as time goes by, explaining the decreasing marginal growth over time.

After week 63, we notice a strong pick up in the number of views, with all average number of views standing over 2 standard errors above the mean of the newer videos. As it is hard to explain from the life cycle stated above, we can look at our other metric, number of subscribers, to see whether a similar pattern can be observed.



Once again, we can notice a strong gap in the average number of subscribers between videos younger than 63 weeks and the older ones, as well as a strong preference for higher-subscribed channels in the first week. As there is no specific reason to explain these strong jumps between weeks 0 and 1, or between weeks 63 and 64, we will assume they are linked to the selectivity introduced by the YouTube search algorithm. To limit its impact on our analysis, we will remove these data points from our sample.

By doing so, our sample now contains 43,258 observations.

Defining additional possible predictors from our data

Before moving forwards, we will compute and add some possible predictors to our data set:

- number of votes, defined as the sum of likes and dislikes for a given video
- engagement ratio, defined as the sum of the number of comments and votes, divided by the number of views for a given video
- likes (and dislikes) per view, defined as the ratio of likes (respectively dislikes) by the number of views
- ratings odds, defined as the ratio between likes and dislikes
- number of characters in the title, knowing that the limit is 100, and that only up to 70 are visible without going to the video itself
- number of hashtags in the title and the description, which are built as a way to categorize videos by the creator using custom categories
- number of words in the title and the description
- proportion of capitalized letters in the title
- proportion of capitalized words in the title

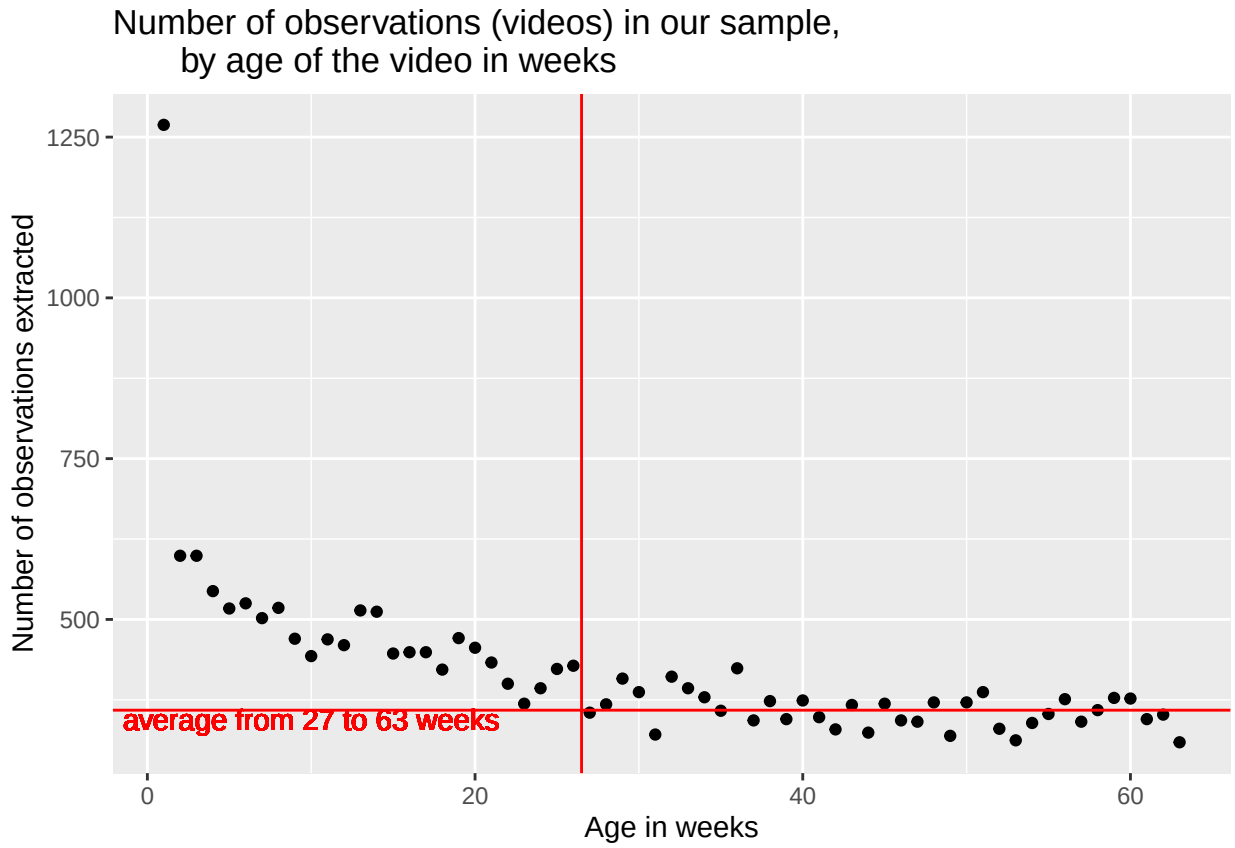
We will also remove observations with 0 values in the number of likes, dislikes, or comments, and add the names of the categories to facilitate readability.

The data set now contains 30,472 observations.
Since YouTube assigns topics to the videos and the channels, we can also consider looking at the fit between the two, which we will define as a binary item taking the value 1 if at least one of the video topics is also a channel topic, and 0 otherwise. To avoid making unjustified assumptions, we will remove all observations where no topic is assigned.

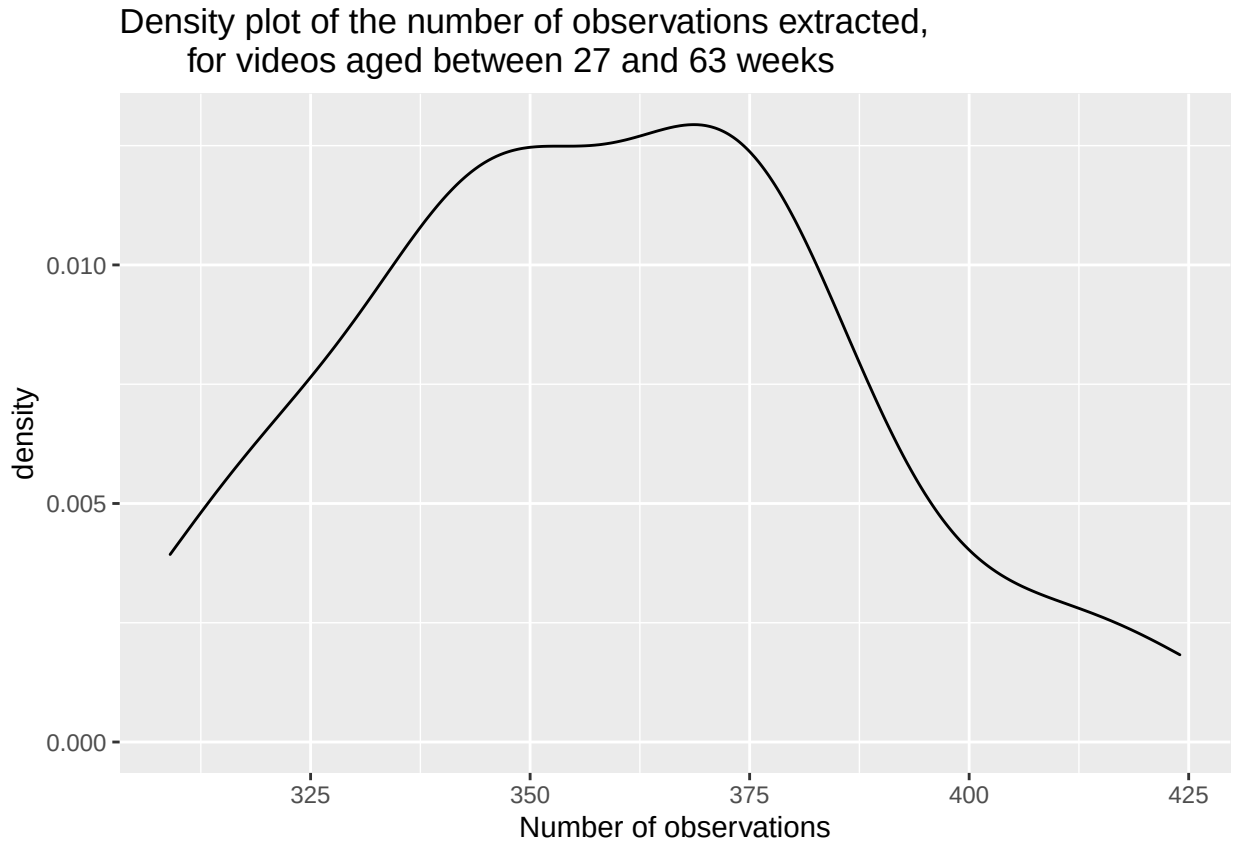
The data set now contains 26,360 observations.

Removing the time selection bias

Earlier, we noticed that our sample had an unexpected distribution of views and subscribers through time, which we attributed to bias introduced by looking at the first 50 results of the YouTube search algorithm. Another similar bias we may expect is that the number of results returned by the search algorithm may not be uniformly distributed across time, i.e. a trend may be seen linking the number of results to the age of the video. We can explore this by looking at the number of observations for each weekly age bucket in our data.



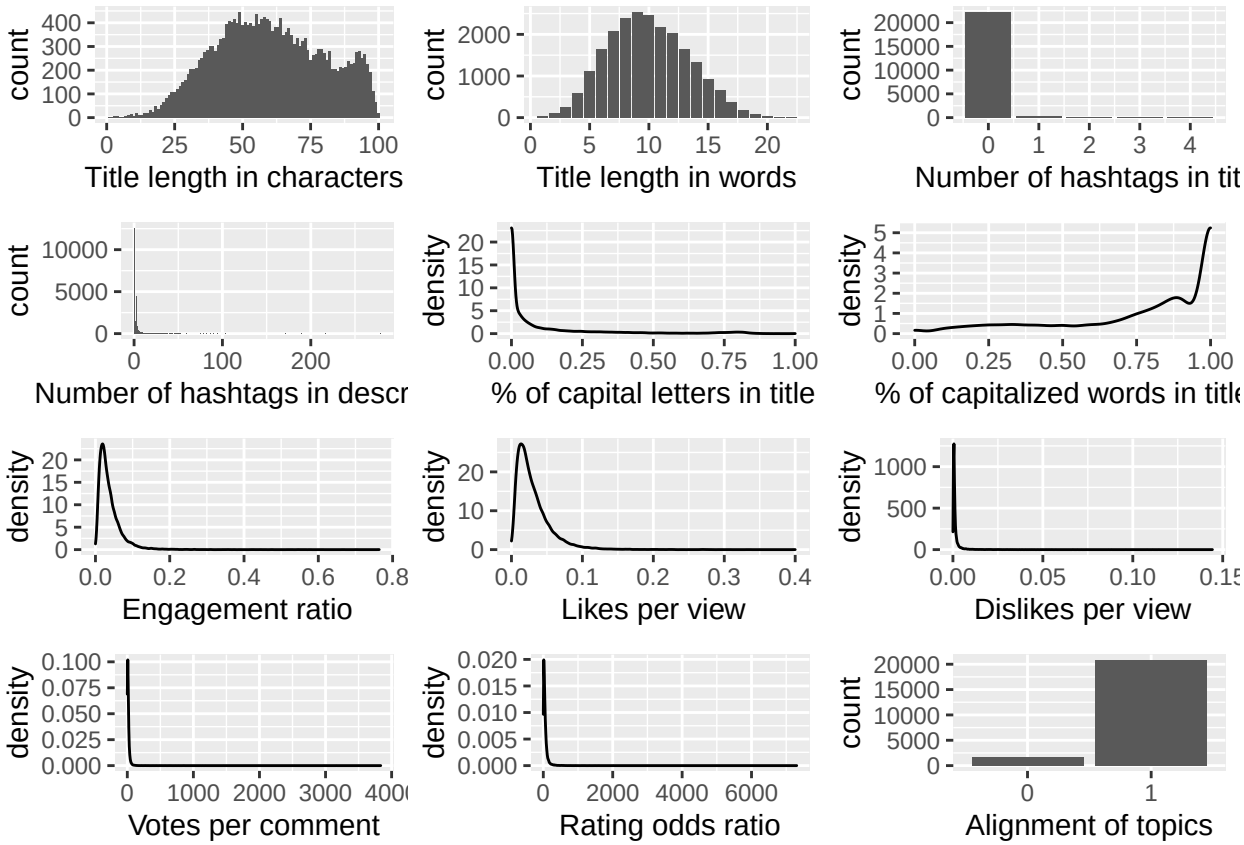
Aside from an expected random variability, we can see a decreasing trend, clearly visible in the first 27 weeks. After that point, the number of data points seem to be normally distributed, as shown in the plot below.



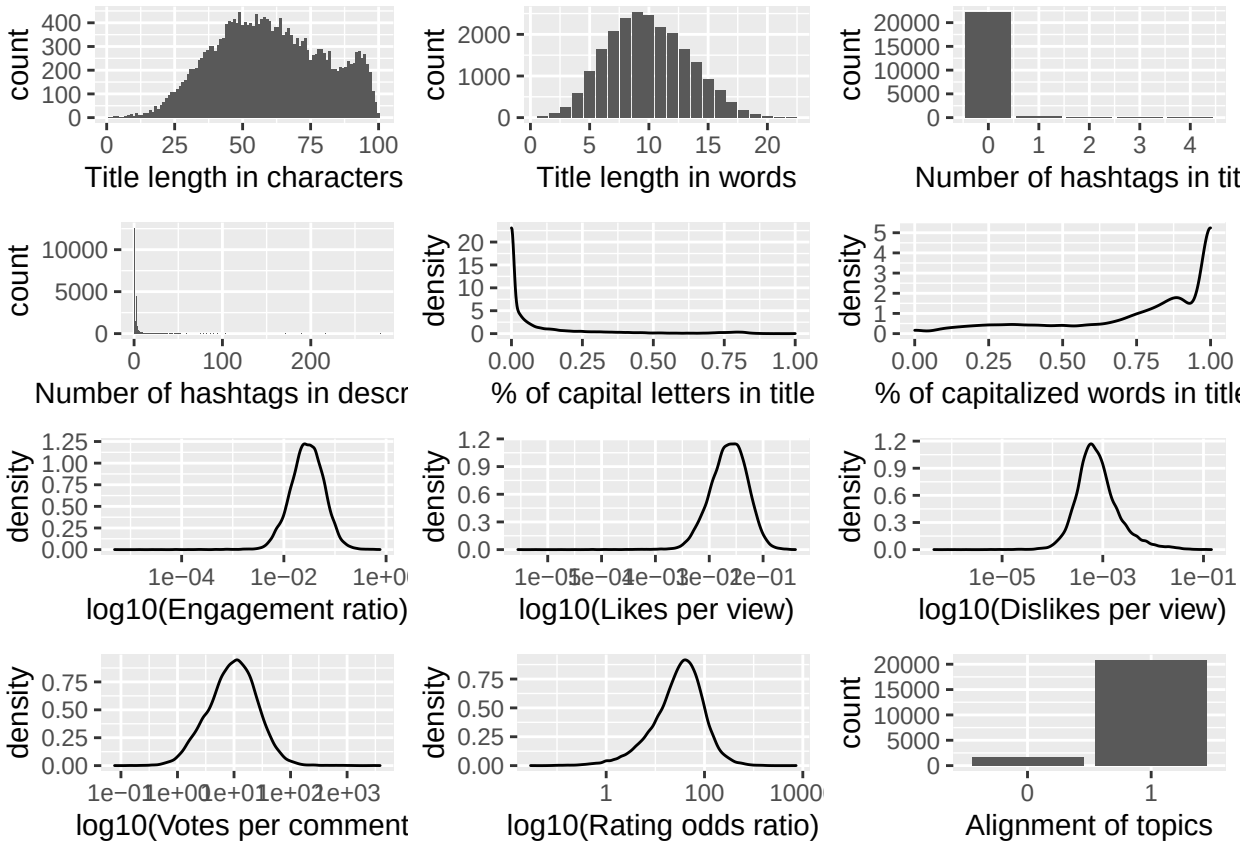
A Shapiro-Wilk normality test on this sample gives a p-value of 0.772, not rejecting the null hypothesis. To mitigate the bias introduced in the number of results before that date, we will randomly select a number of observation for each week before week 27 that is equal to the average number of observations for weeks 30 onward, i.e. 359, while keeping the full sample if the number of observations are lower than this amount.

The data set now contains 22,613 observations.

To finalize the data wrangling phase, we can explore the distribution of our newly created ratios and potential predictors, to assess whether further filters or transformations are useful.



After log10-transforming the x-axis for engagement, likes per view, dislikes per view, votes per comment, and ratings odds, we get the following view.



From these plots, we can see that engagement, likes per view, dislikes per view, votes per comment, and the rating odds ratio seem to be log10-normally distributed in our sample. We can also see that the number of characters in the title may follow a bi-modal normal distribution, while the number of words in the title looks closer to a Poisson distribution or a normal distribution with a positive skew. We can also see that very few videos use hashtags in their titles, that few tend to capitalize all title letters but many capitalize most words, and that the topics of most videos is broadly aligned with the topics of the associated channel. To simplify our analysis onward, we will log10-transform the relevant metrics in our data set to create our final wrangled data set.

Data exploration

Preliminary modelling

As previously stated, the goal of this project is not to create a full predictive model for views, but rather to identify whether the relative success of a video can be at least partly predicted using the video's own characteristics.
For the purpose of this analysis, we need to define what relative success means, i.e. what can be used as a baseline. From our preliminary data exploration, we determined that, at least in our sample, log10(views) can be seen as a function of time and log10(subscribers). We will write this function as follows:

$$\log 10(views) = f(time, log10(subscribers)) + \varepsilon$$

which we will assume at this stage can be simplified into

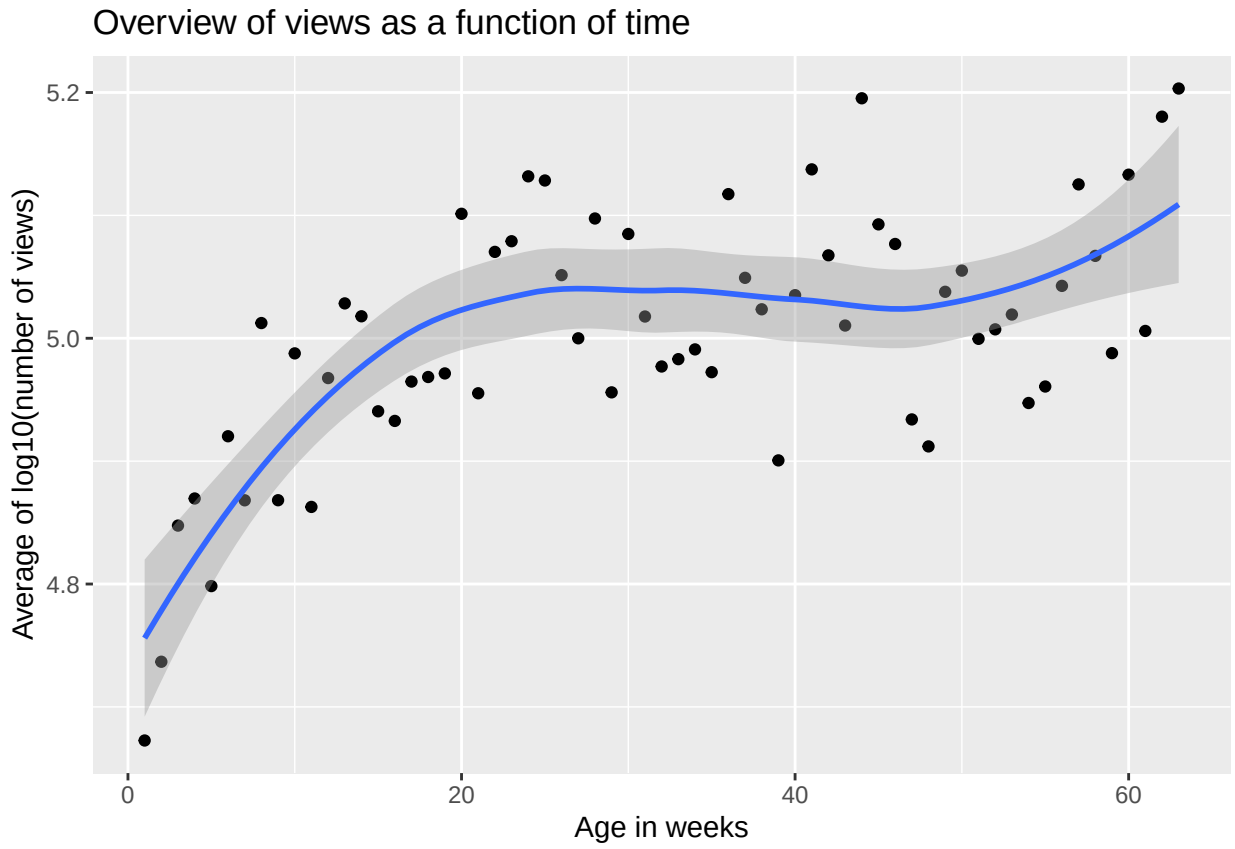
$$\log 10(views) = f_1(time) + f_2(log10(subscribers)) + \varepsilon$$

Where f_1 and f_2 are independent functions that we would expect to be monotonically non-decreasing.

For the purpose of our project, we will define relative success as the ε in the above function, i.e. the variability or log10(views) for a given age and number of subscribers. Our goal will then be to determine whether we can identify predictors that will help determine whether this ε takes a positive or a negative value, i.e. whether log10(views) is likely to be higher than its expected value for a given video age and a given number of channel subscribers.

Due to the assumed non-random nature of our API-based sampling approach, we are unlikely to be able to determine the accurate value of either of the baseline functions for the whole population of YouTube videos. We can however try to neutralize their impact in our sample.

We will start by looking at log10(views) as a function of the age of the video.



As a simple approach, we are considering weekly time buckets, and will use them to estimate the impact of time as follows:

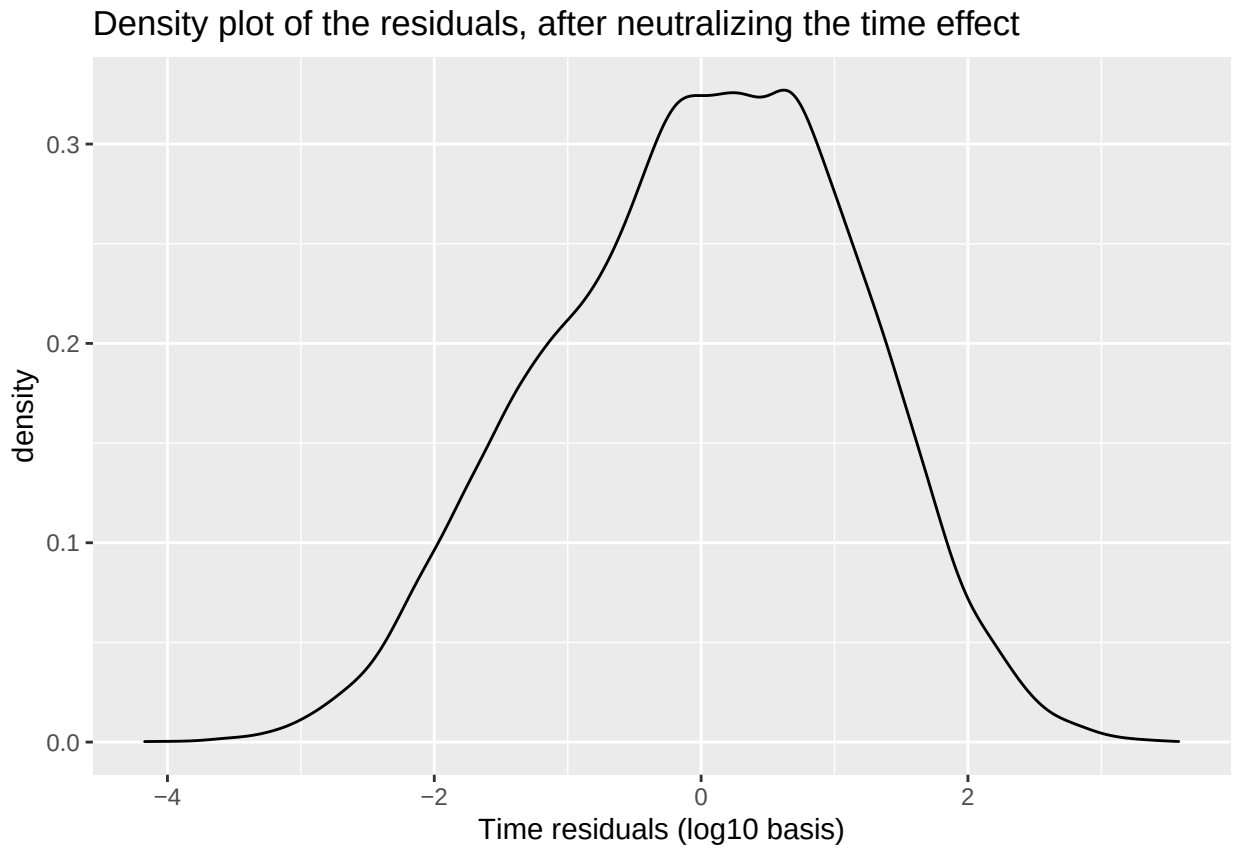
$$\hat{f}_1(time) = \frac{1}{n_w} \sum_{i=1}^{n_w} \log 10(views_i)$$

where n_w represents the number of observations in our sample for the week where *time* is located, and $views_i$ is the number of views for videos with an age contained within the weekly bucket of *time*. As we are not trying to model this part of the relationship, but are only looking to neutralize the impact on our data, we will use the full sample to define it. We are not interested here in the accuracy of this part of the model. If we were looking to create a full predictive model for the number of views, we would need to go through the regular model training and testing process, using different data sets for both.

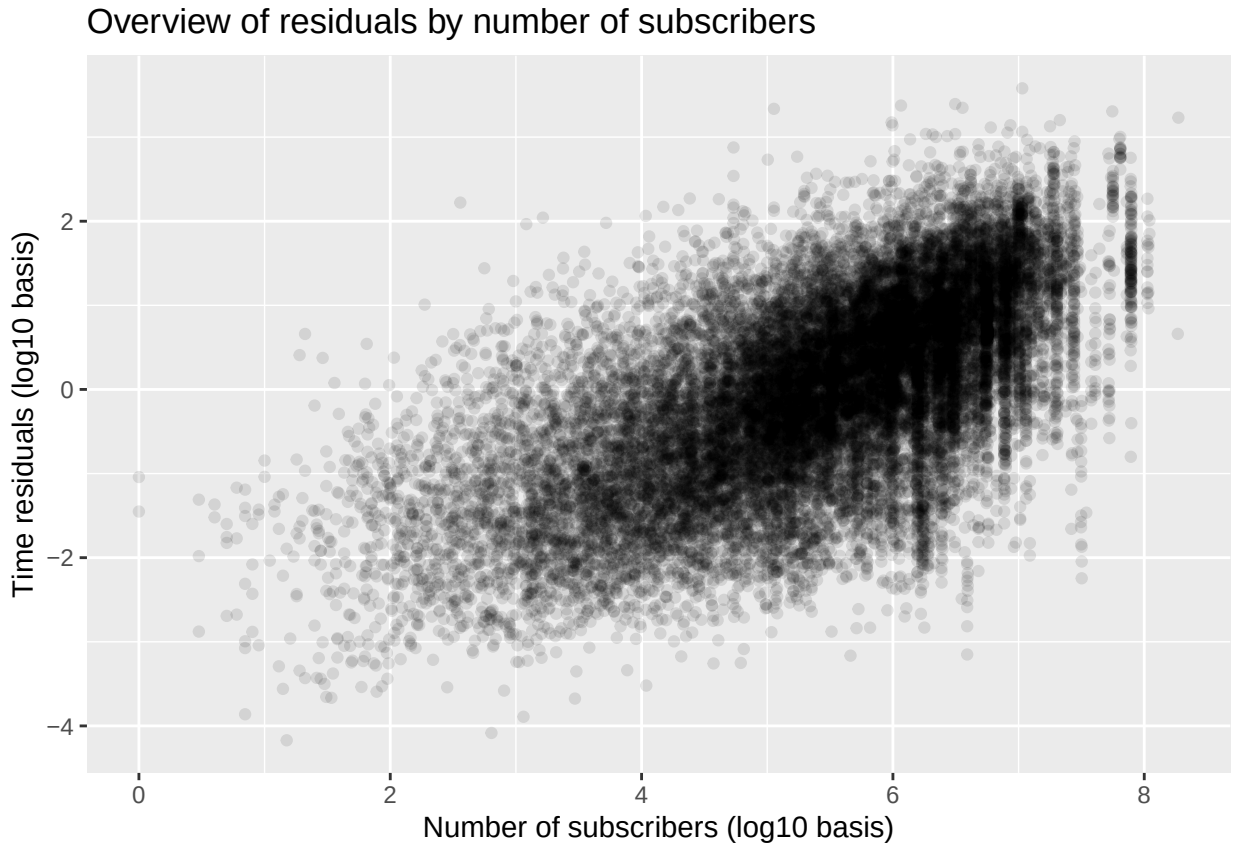
Using this estimate for f_1, we will compute the time residuals $residuals_t$ as follows:

$$residuals_t = \log 10(views) - \hat{f}_1(time) = f_2(log10(subscribers)) + \varepsilon$$

and add the values to our data set. We can then plot the density of these values.



Next, we will try to determine f_2 for our sample, i.e. the relationship between the time residuals and log10(subscribers). Plotting the data can give us an idea of the relationship.



From the shape of the data, we can infer a linear relationship between the two. We will hence use a linear regression model to approach f_2 for our sample.

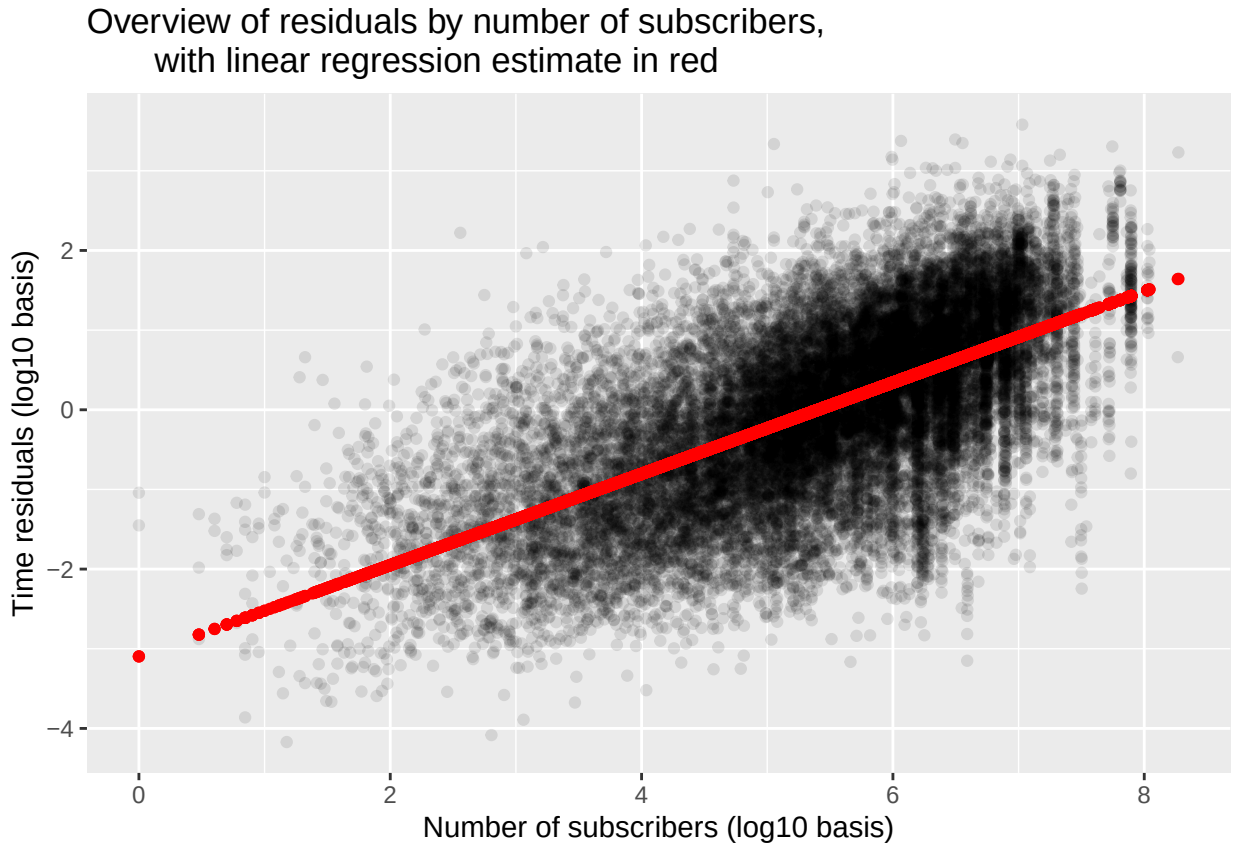
Once again using the full data set, which would be overtraining if we were looking to define this part of the model for general purposes, we get the following equation, only applicable in our sample:

$$\hat{f}_2(log10(subscribers)) = -3.096 + 0.573 \times log10(subscribers)$$

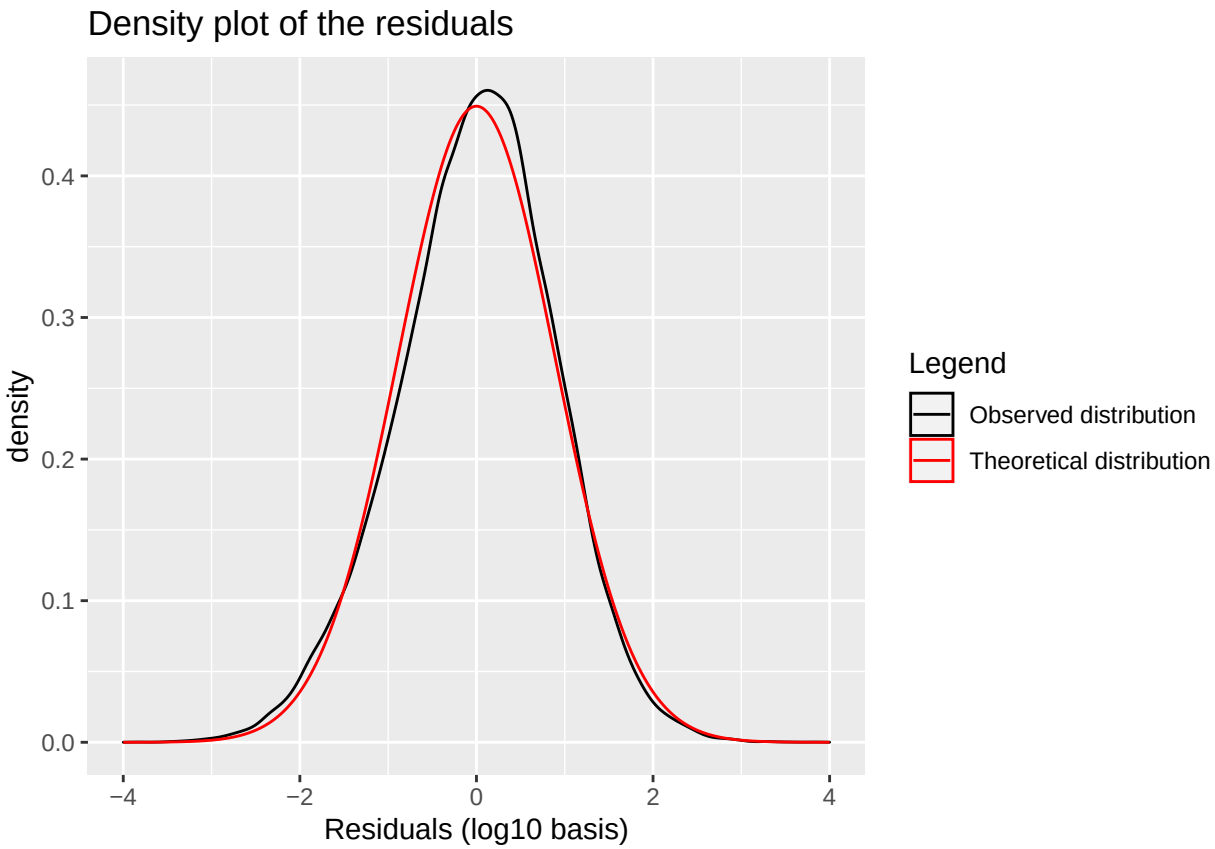
Using this estimate for f_2, we will compute the residuals, i.e.

$$residuals = \log 10(views) - \hat{f}_1(time) - \hat{f}_2(log10(subscribers)) = \hat{\varepsilon}$$

and add the values to our data set. We also show the shape of our linear regression and compare it to our time residuals.



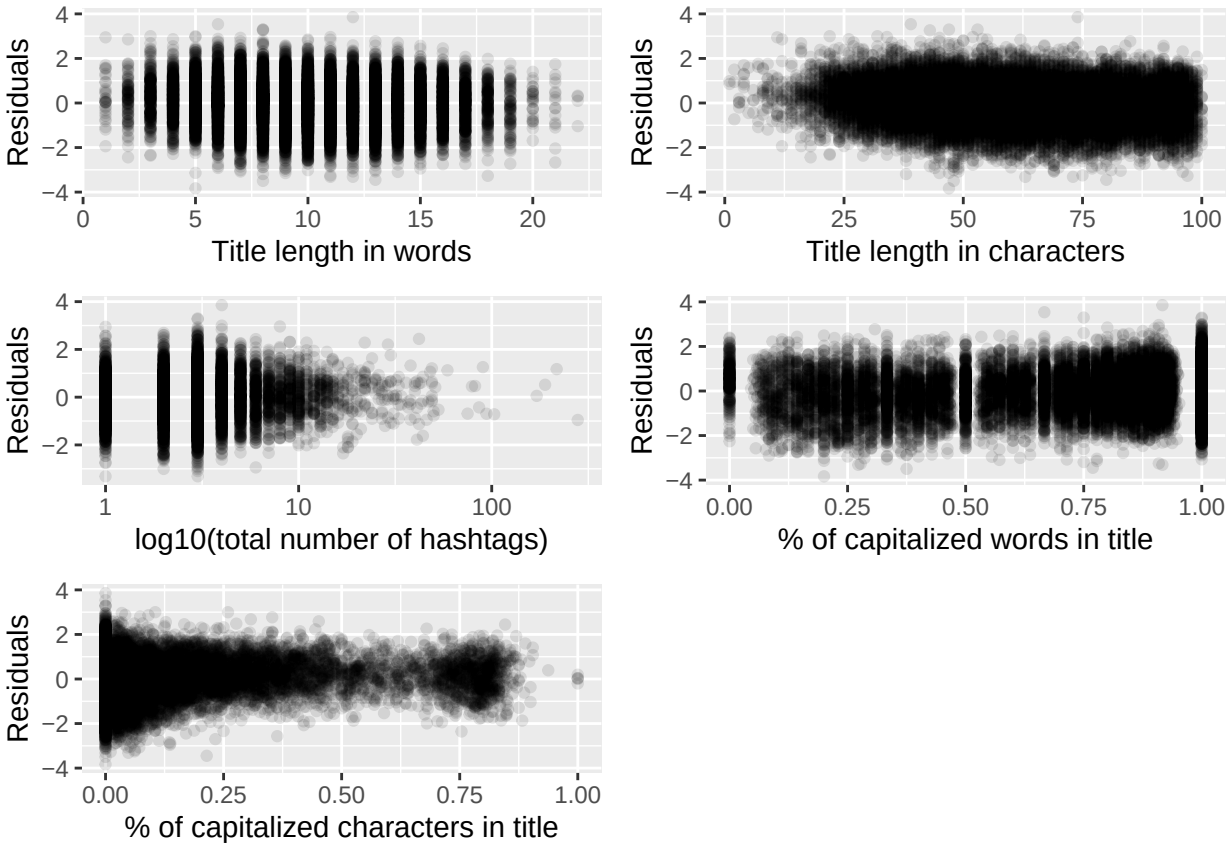
The remaining residuals should now represent an estimate of the ε in our sample. We can look at their distribution density and assess whether they are close to following a normal distribution.



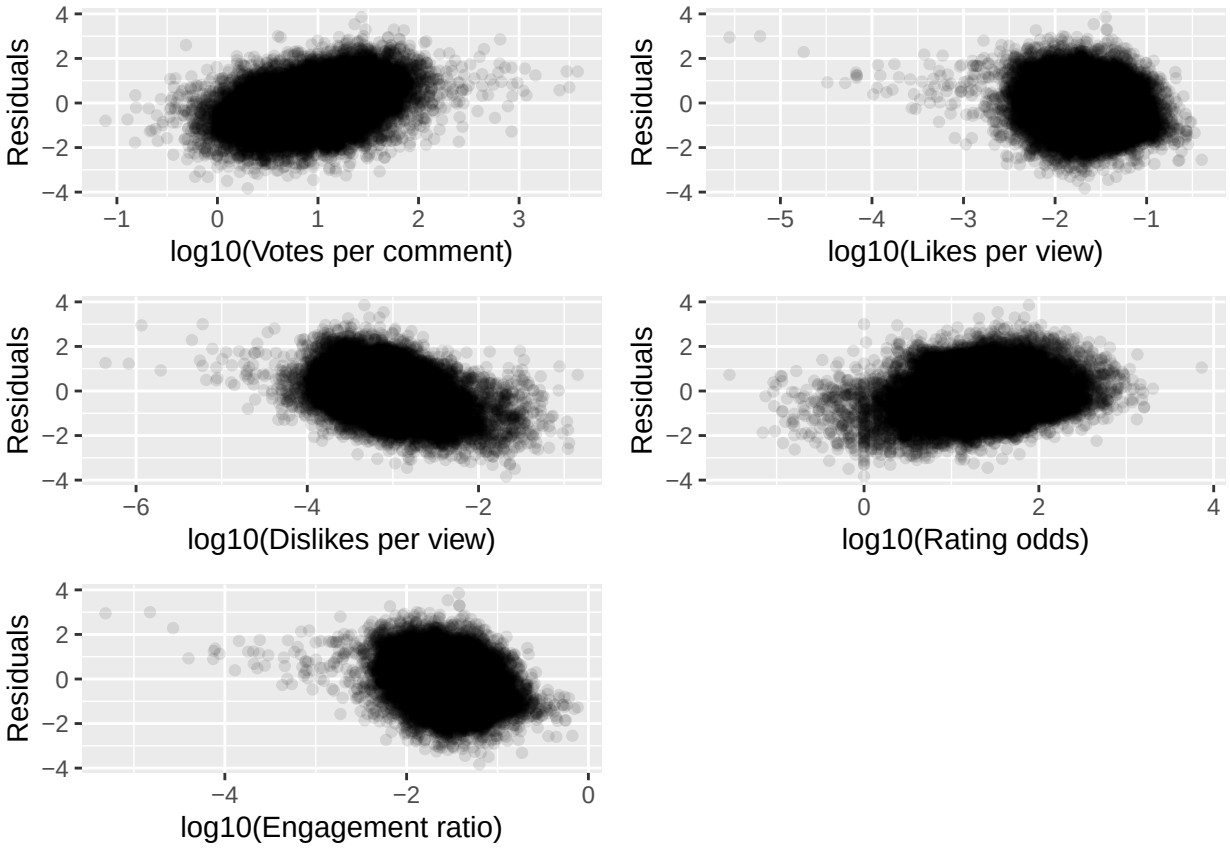
The result seems to be close to a normal distribution centered around 0 with a standard deviation of 0.888, with a slightly negative skew (skewness = -0.176) that can be seen in the distribution curve, and a mesokurtic distribution (kurtosis = 3.141).

Exploring possible relationships

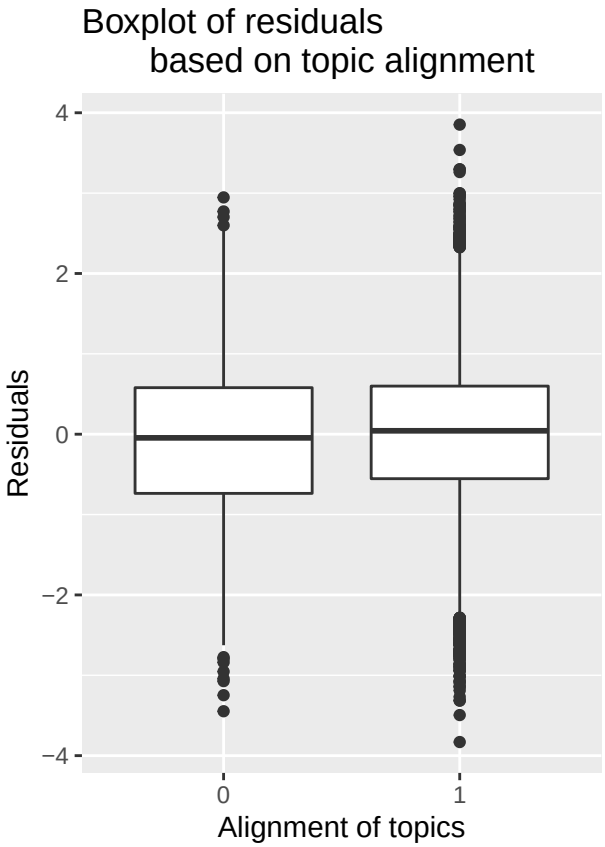
Using our residuals as an estimate of ε for our sample, we can now start looking for possible relationships between them and our other metrics. To begin with, we will look at plots of our residuals against every one of the verbal-based possible predictors, to assess whether visible relationships exist.

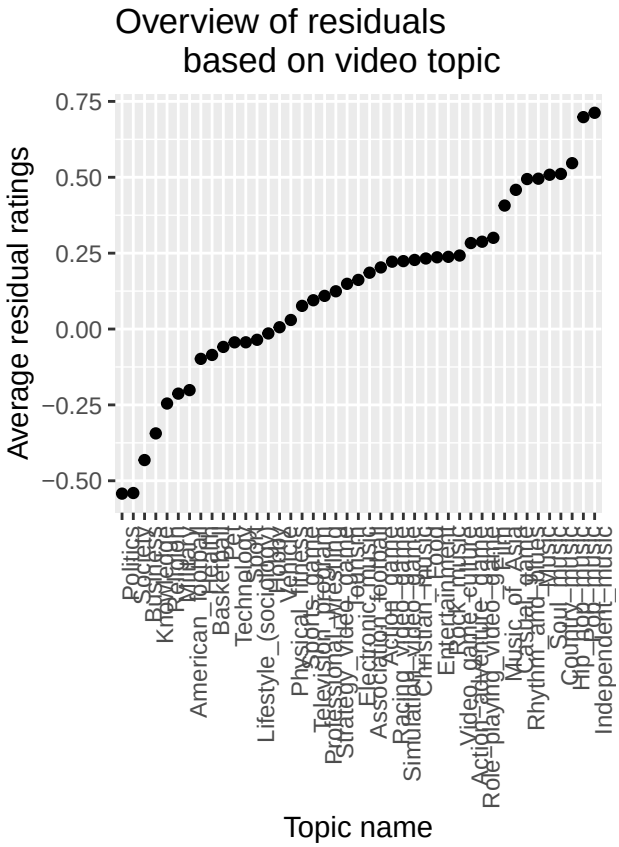


We next have a similar look at the numeric-based possible predictors.

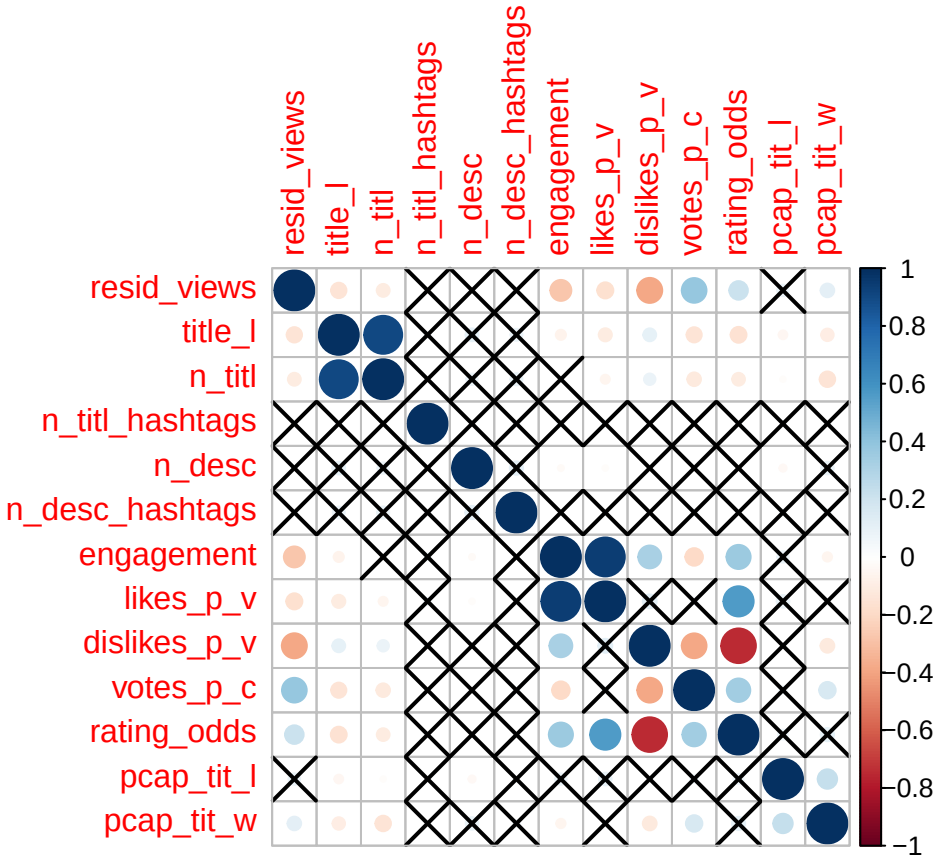


Finally, we have a look at whether the fit of categories, or specific categories, impact the number of views in a material way.

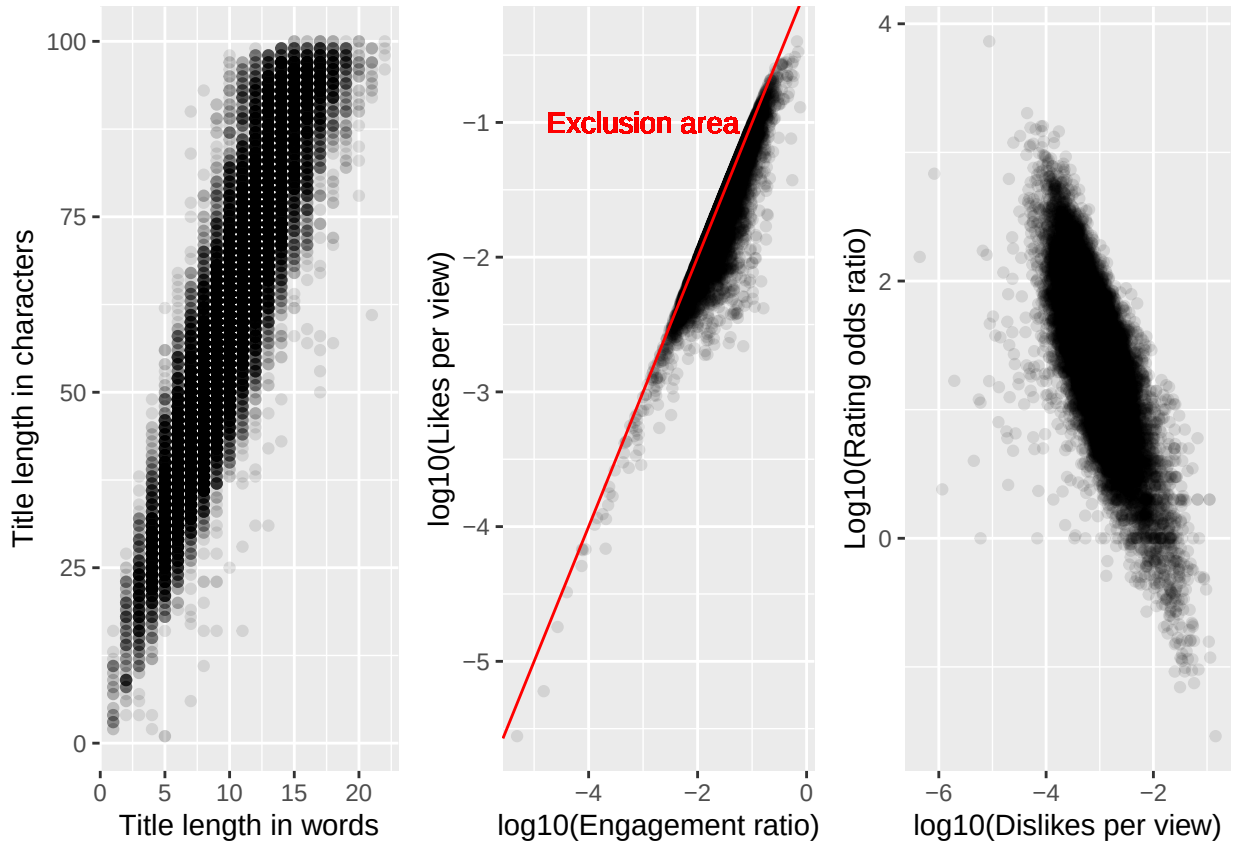




From this data, we may visually infer a relationship between our residuals and the number of votes per comment, the number of dislikes per view, and the rating odds. Less evident relationships may exist between our residuals and the the number of characters and words in the title, and the engagement ratio. However, the alignment between video and channel topics, the number of total hashtags, and the percentage of capitalized words or letters seem to have almost no impact on the comparative success in our sample.
We can visualize all correlations between our variables in the following matrix, where we numerically highlight relationships where the significant level is higher than 0.5, to exclude them from the analysis.



Aside from the relationships we identified earlier, we can see strong correlations between the number of words in the title and the number of characters in the title, the engagement ratio and the number of likes per view, and the rating odds and the number of likes and dislikes per view. These are expected, considering how these metrics were built. We can visually check whether they follow a normal bivariate distribution.

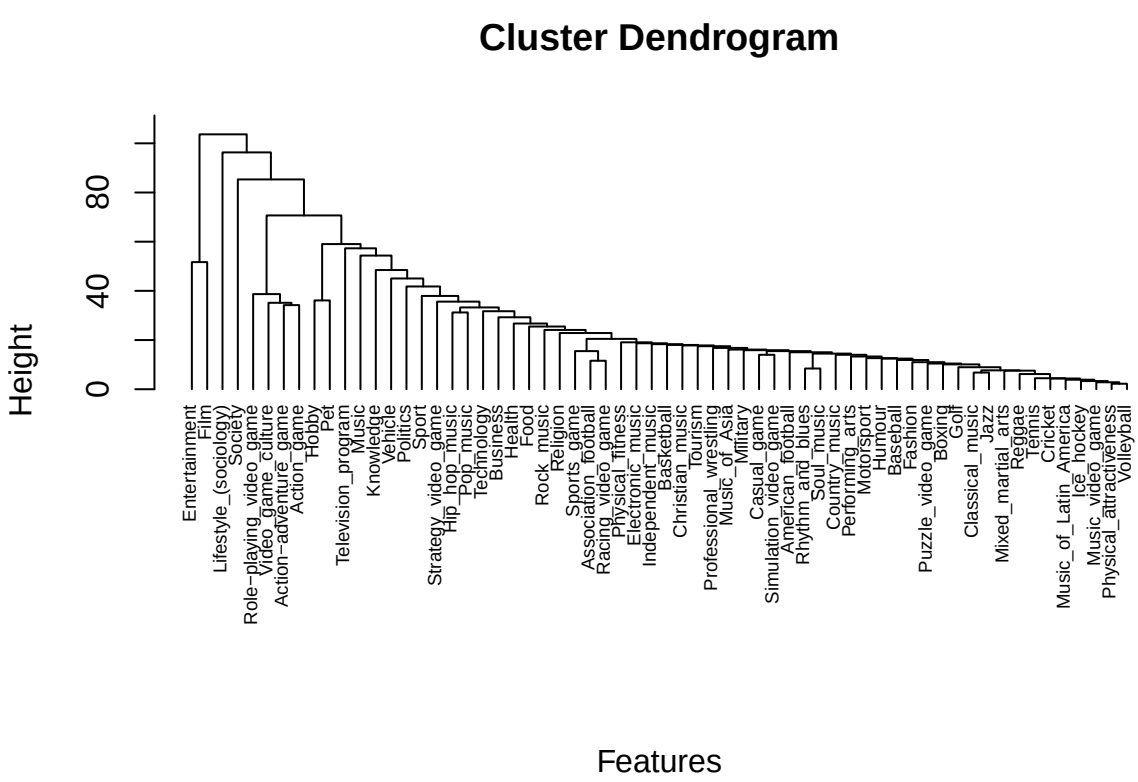


We will remove the values that have either a negligible correlation with our residuals (significant level above 0.5), i.e. the number of hashtags, the length of the description, the proportion of capitalized words in the title, and values that are largely explained by others, i.e. the number of characters in the title, the number of likes per view, and the rating odds, keeping only the variables with the highest correlation to our residuals.

Exploring video topics

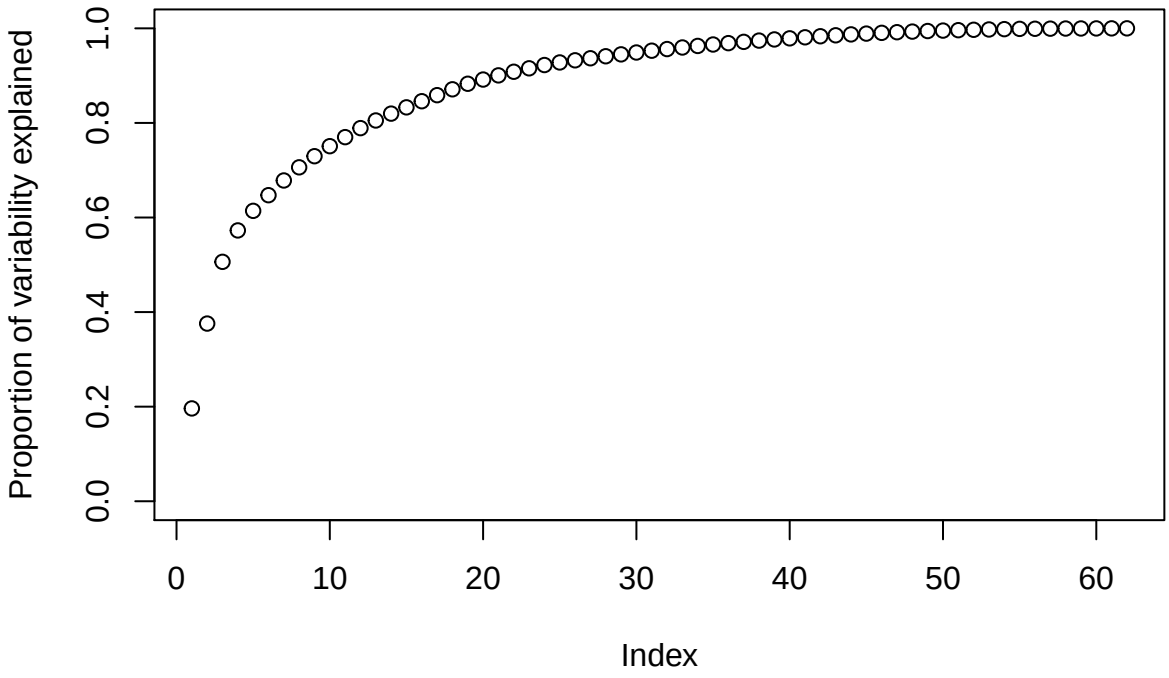
A piece of information we have given little attention to so far is the information we have on topics associated with a given video. Using our sample, we will first transform the topics information into a matrix, where each row represents a video, and each row one of the categories available in our sample. For each row, we will assign 1 if the topic is associated with the video, and 0 otherwise.

We now have a matrix of 22,613 rows (the number of videos in our final sample) and 62 columns (the number of different topics associated with these videos). We can take a glance at a cluster dendrogram of these topics.



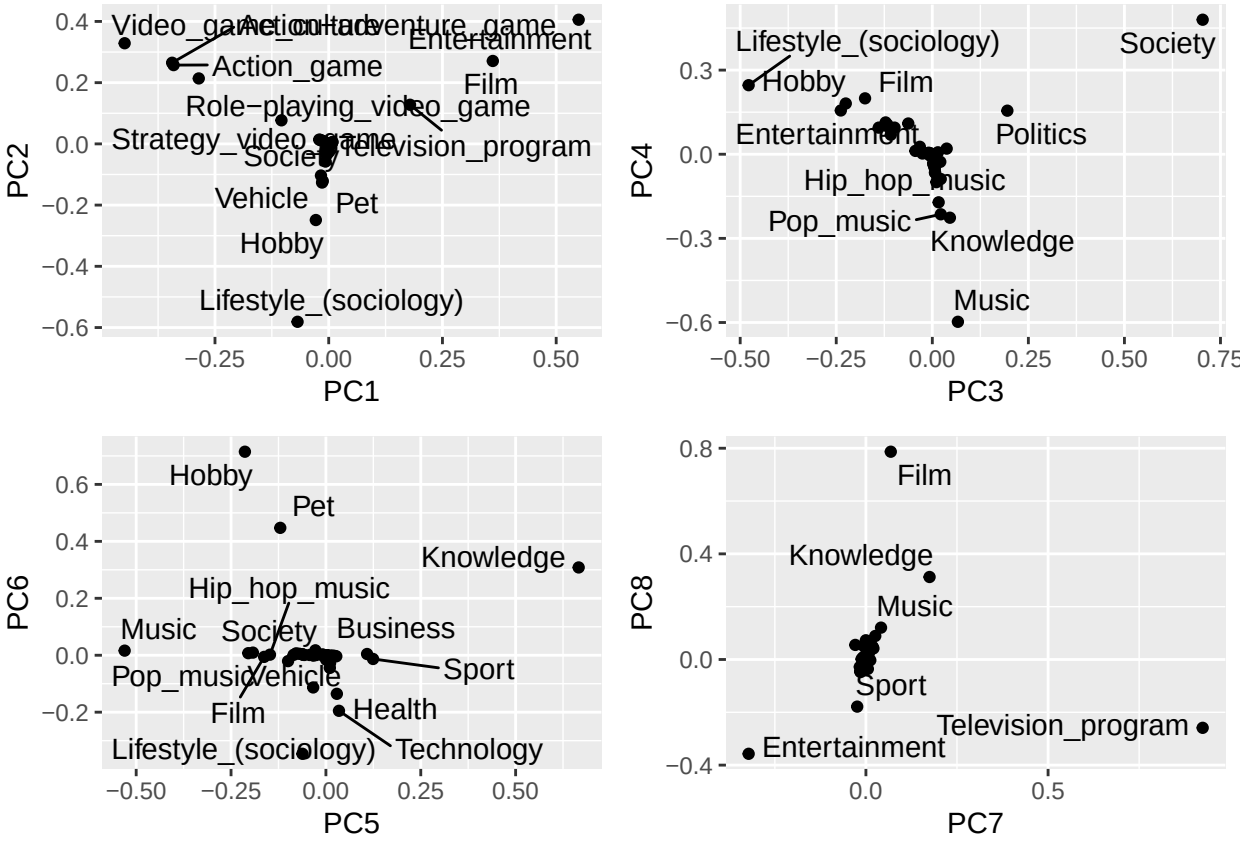
Given the size of the matrix, the next step will be to simplify it for use in future models. We will do this through a dimension reduction through a matrix decomposition.

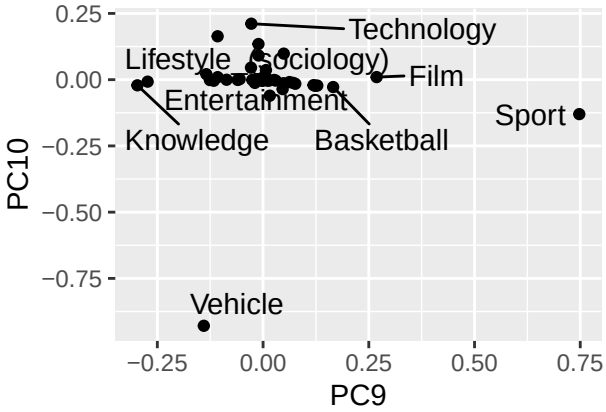
For the purpose of this analysis, we will use a principal component analysis. Using this approach, we can see that using the 13 first dimensions, we can explain more than 80% of the total variability of our original matrix, as shown in the following plot.

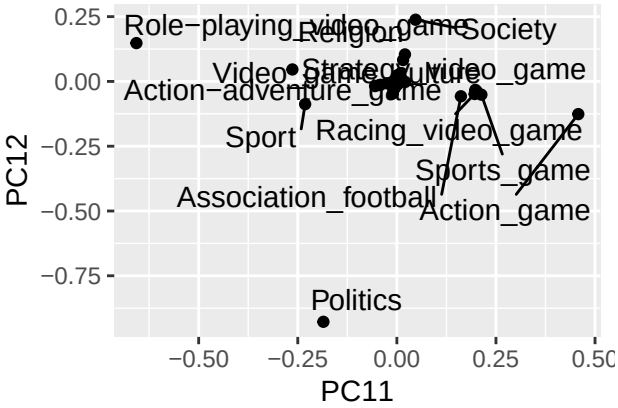


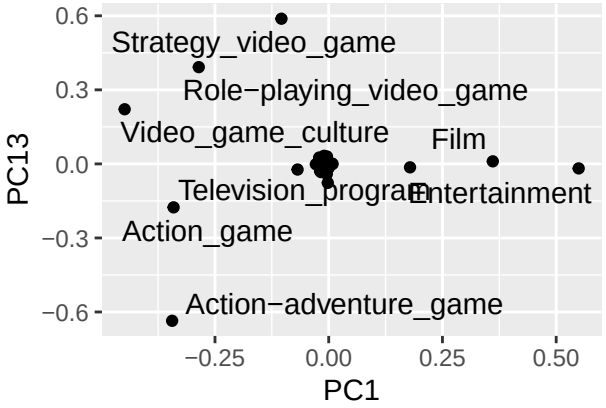
We can hence keep the first 13 dimensions of our transformed matrix, to speed up our algorithmic approach later on, without losing much information.

We should also determine what these dimensions represent. As a first approach, this can be done by plotting them two by two, as follows.


Video_game
Action_adventure_game
Entertainment
Action_game
Film
Role−playing_video_game
Strategy_video_game
Society
Television_program
Vehicle
Pet
Hobby
Lifestyle_(sociology)
PC1
PC2
Lifestyle_(sociology)
Society
Hobby
Film
Entertainment
Politics
Hip_hop_music
Pop_music
Knowledge
Music
PC3
PC4
Hobby
Pet
Knowledge
Hip_hop_music
Music
Society
Business
Sport
Pop_music
Vehicle
Film
Health
Lifestyle_(sociology)
Technology
PC5
PC6
Film
Knowledge
Music
Sport
Television_program
Entertainment
PC7
PC8







Based on these plots, we can infer that:

- PC1 differentiates between generic entertainment or movies and video games
- PC2 differentiates between personal topics or hobbies and more generic topics
- PC3 differentiates between personal and political topics
- PC4 differentiates between societal or personal topics and music or knowledge
- PC5 differentiates between knowledge and generic topics
- PC6 differentiates between "show and tell" type videos and generic ones
- PC7 differentiates between TV programs and others
- PC8 differentiates between films and TV or generic entertainment videos
- PC9 differentiates between sports and non-sport videos
- PC10 differentiates between vehicles and non-vehicles videos
- PC11 differentiates between types of video games, from action / sports-oriented to role-playing ones
- PC12 differentiates between politics-focused and other videos
- PC13 differentiates between types of video games, from slower paced strategy games to faster paced action-adventure games

Modeling relative popularity of videos using the previously defined predictors

With the data set created so far, we will start looking into fitting models to our problem. First, we should refine our outcomes into a binary outcome, based on the statement "the residual is strictly higher than 0", where 1 will be the positive (statement is true), and 0 the negative (statement is false) for each observation

in our sample. In other words, we are using the binary $f(\varepsilon)$, defined as:

$$f(\varepsilon) = \begin{cases} 1 & \text{if } \varepsilon > 0 \\ 0 & \text{if } \varepsilon \leq 0 \end{cases}$$

where ε represents the residuals after neutralizing the $time$ and $\log 10(subscribers)$ impacts on $\log 10(views)$, i.e.

$$\varepsilon = \log 10(views) - \hat{f}_1(time) + \hat{f}_2(log10(subscribers))$$

Given that $\log 10(x)$ is a continuous monotonically non-decreasing function for all $x \in \mathbb{R}_+^*$, and the number of views is a positive integer (strictly positive in our sample), this means that if ε is higher than 0, the video has received more views than a video of a similar age and from a channel with a similar number of subscribers would be expected to receive, based on our sample.

We next split the data into a training and a test set, to use in the modeling phase. For this step, we use a random data partition, keeping 20% of the sample for testing. Since we will be looking into numerical categorization algorithms as well, we create a dedicated subset of training and test sets, by removing the column containing category names from the regular training and test sets.

Assess baseline accuracy, through random guessing

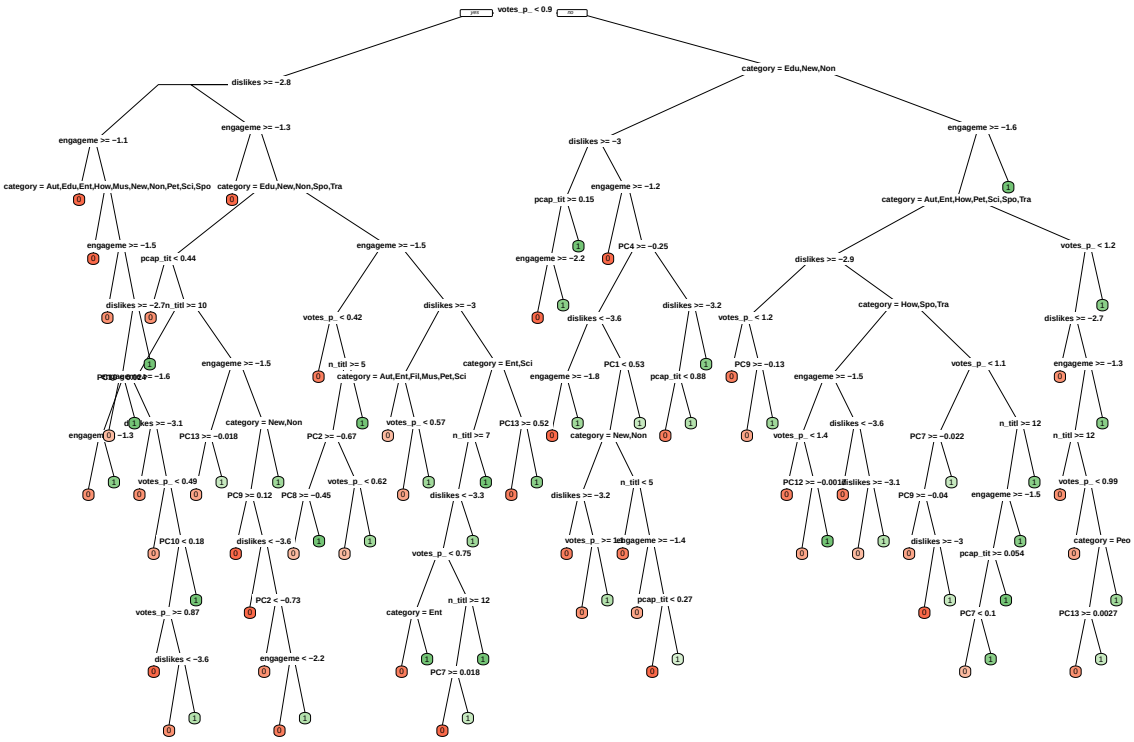
To assess the benefits of our models, we need to define a simple approach to use as baseline. To this end, we will randomly guess whether the video has an above or below average observation, i.e. assign it a 0 or 1 value in our approach.

With this approach, we reach an accuracy of 0.509 on our test sample.

Classification tree

As our first approach, we will use a classification tree algorithm to try and determine if a decision tree can help categorize the data between higher than expected and lower than expected views. Classification trees, or decision trees, are used in prediction problems where the outcome is categorical, and the algorithm generally look for the best variable-based splits to minimize the loss function (in our case the Gini index).
A benefit of a decision tree is its readability and ease of use outside of algorithms. We will use the `rpart` package in R to this end, and tune the model using complexity parameters ranging from 0.0001 to 0.0021. For this model, and for all following ones, we will use a ten-fold cross validation on our training set, with 10% of the observations in each sample.

We get the best accuracy with a complexity parameter of 0.001. Using our testing set, this model is able to predict whether a video has an above-expected number of views with an accuracy of 0.689, 0.18 higher than our random guessing approach.
The resulting decision tree is shown below.



We can notice on this tree the relative importance of the number of votes per comment (votes_p_), number of dislikes per view (dislikes), category, and engagement (engageme) for broad categorizations. We can also see some of the dimensions of our reduced topics matrix appearing, which were discussed earlier.

Random forests

While not as powerful as a human-legible algorithm as a single decision tree, we will try to see if using a random forests algorithm can lead to an improved accuracy. The goal of this type of algorithm is to improve prediction performance and reduce instability by averaging multiple decision trees (a forest of trees constructed with randomness).
To this end, we will train an algorithm using the `Rborist` package in R, tuning it using nodes ranging from 12 to 16, and testing 2 predictors for each given split.

The results show the model with the best accuracy uses 14 nodes. Using our testing set, this model is able to predict whether a video has an above-expected number of views with an accuracy of 0.683.

Random ferns

Another approach we can use on our classification problem are random ferns. While classification tree and random forests use a standard naive Bayes approach, which considers all predictors / features are conditionally independent for a given outcome, random ferns looks into possible groups of interdependent features, with each group instead independent from the other.
While this approach is more useful for a very large number of predictors (e.g. in image-based algorithms), we can still assess if the accuracy of trees and forests can be improved here. We will use depths ranging from 13 to 16 for tuning purposes in the training phase.

The results show the model with the best accuracy has a depth of 15. Using our testing set, this model is able to predict whether a video has an accuracy of 0.682.

Linear discriminant analysis

After removing non-numeric values from our test and train sets, numeric classification approaches can be used. Since our number of predictors remains moderate, we can start by using a linear discriminant analysis model. This however assumes that the variables are multivariate normal, and that the correlation structure is the same for all classes.

After training the model on the training set, we find the model has an accuracy of 0.67 when predicting whether the number of views is above expectation on the test set.

K-nearest neighbors

Next, we will use a k-nearest-neighbors model. Since this is a situation where we expect videos with close characteristics to behave similarly, this can be a powerful and simple approach.
For tuning purposes, we use a number of neighbors ranging from 70 to 140.

Using the test set, we find the model has a maximum accuracy of 0.641, reached when k = 80 neighbors.

Neural network

Lastly, a neural network can be trained on our data set. We will use a single-hidden-layer feed-forward neural network, through the `nnet` package in R. Feed-forward neural networks are neural networks where connections between units are not cyclical: the information flows one way, from the input nodes to the output nodes, after going through the hidden layer. Neural networks are algorithms based on the biological neural networks, that, on a basic level, work by adding weighted inputs in each neuron, to which a *bias* value can be added, and the results then is used through an activation function to determine whether the neuron is activated.
To tune the model, between 10 and 20 units in the hidden layer were used, with weight decays ranging from 0.15 to 0.25.

Using the test set, we find the model has a maximum accuracy of 0.702, reached when using 15 units in the hidden layer, and a weight decay of 0.25. We may point out that the time needed for the training, for a similar accuracy as other simpler approaches may make this one inefficient for our problem if the data set becomes much larger.

Exploring a combination of models

One can notice that most of the models used give very similar accuracies on our test set. Before concluding, we will look if a combination of these models can improve the accuracy of our predictions.
The best combination will be assessed on the test test, using a custom-built algorithm that considers every possible combination of models, where each is used at most once, and identifies the most accurate combination overall.

Using this approach, we find that the best combination on the training set is to combine the following: Classification tree, Random forests, Neural network. Using the test set, this yields an accuracy of 0.704, which is very slightly better than any of the individual models.

Results

Using the models presented in the previous section, we predicted whether the videos in the test set had an above-expected number of views with the following accuracy:

Table 1: Accuracy for each model

model	accuracy
Random guess	0.509
Classification tree	0.689
Random forests	0.683
Random ferns	0.682
LDA	0.670
KNN	0.641
Neural network	0.702
Best combination	0.704

We can conclude that combining these models yields a small improvement over any other model. However, with every model used yielding close accuracies, that remain relatively far from optimal, there is no clear choice at this stage.

Conclusion

As far as our initial problem is concerned, we have determined, using several different algorithmic approaches, that one is able to predict with close to 70% accuracy whether a video will be more or less successful than what would be expected from a video of the same age and by a channel with a similar amount of subscribers. Using our classification tree approach, we can see that, aside from the broad category of the video, metrics related to viewer interaction, such as the number of votes per comment, the number of dislikes per view, and the engagement metric defined earlier, are strong indicators of relative popularity. Video topics also play a role in this, and can be used to refine this binary decision down the line.

Discussion

Usability

With a 70% accuracy, the approach used and end model can be considered as a starting point for future analyses. Nevertheless, using the best combination of models, the test set was separated into two sets depending on the outcome (1 and 0), with significantly different average gaps to the expected log10(views) reported, using the test set:

- -0.481 if the outcome from the model is 0
- 0.329 if the outcome from the model is 1

While this approach has hence a limited usability when predicting an individual video's relative popularity, due to its 70% accuracy, it may be a starting point when used on a larger sample size. Depending on the need, it also seems to provide a slightly higher specificity (0.794) than sensitivity (0.606).

Time as a possible confounder

Earlier in this analysis, we assumed that the equation

$$\log 10(views) = f(time, log10(subscribers)) + \varepsilon$$

could be simplified into

$$\log 10(views) = f_1(time) + f_2(log10(subscribers)) + \varepsilon$$

This notably assumes that time and log10(subscribers) are independent. However, one can assume that the number of subscribers of a channel is also a function of time. While we have tried to mitigate the impact of this in our analysis by focusing on a limited time frame (63 weeks), and log-transforming the number of subscribers, the change in number of subscribers for a channel since the individual video was uploaded may ultimately have had an impact on our modeling.
A way to remove this effect would be to use the number of subscribers of a channel at the time of the video's initial upload, which unfortunately does not seem to be available through the public API used.

Other likely predictors

Through the approach used, a specific set of metrics / variables were iteratively selected to be used as predictors. Nevertheless, several other variables, of possible material importance, were not considered here. A first, likely important one, consists in the thumbnail image for each video. YouTube being by essence a primarily visual social media platform, it can be expected that, together with the title and the channel, the thumbnail of the video would play a significant role in whether or not the video is viewed. To possibly improve our approach by taking this data into consideration, one could train a machine learning algorithm to either classify thumbnail images, or identify specific patterns, and include the results as another potential predictor.
Another item that has not been considered are views originating from interactions external to YouTube. One can advertise videos on websites or other social media, generating additional momentum towards views, which is not captured by our approach. While more complex since the YouTube API cannot be used to gather this kind of information, as a first approach one could focus on a few social media sites and identify links to the video posted around the time of its initial upload.

Future work

To improve on this study, we can consider some follow up steps:

- First step would be to generate a truly random set of videos. This can possibly be achieved by looking into the full list of results (instead of the first 50) from each query into the YouTube search algorithm. However, given the number of videos on the platform, it may be unreasonably long, and the public API limitations would severely increase the time required. Another option would be to randomly guess video IDs. However, there are 4.4579157×10^{66} possible IDs, and though no official data exists on the number of videos available on the platform, one would assume the total amount of IDs in use to be a negligible percentage of this number, hence randomly guessing video IDs would be very inefficient.

- Another step would be to complete our list of predictors by including an image analysis on the available snapshots.

- Finally, and ideally, building on the work presented here and the few steps exposed above, one could work towards building a full prediction model for video views, including the two baseline predictors (time and subscribers), and iteratively completing it with predictors from the previous analyses, starting with the most important ones and going down the list until the loss function is small enough for the model to be usable.

References

Introduction to Data Science, R. A. Irizarry, 2021

rFerns: An Implementation of the Random Ferns Method for General-Purpose Machine Learning, M. B. Kursa, Journal of Statistical Software, 2014

Fast Keypoint Recognition in Ten Lines of Code, M. Özuysal, P. Fua, V. Lepetit, 2007

Feedforward neural network: an introduction, in Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives, S. Haykin, 2001