# What makes a Youtube Video Viral?

## Introduction

The topic of identifying viral videos on YouTube involves analyzing data to understand the factors that contribute to a video's popularity on the platform. With the rise of social media and online video consumption, identifying these factors has become increasingly important for content creators, marketers, and other stakeholders in the industry. The motivation behind this project is to provide insights into the various factors that make a video popular or trending on YouTube. By analyzing a range of variables, such as views, likes, dislikes, comments and categories, we can better understand what makes a video stand out and gain traction.

The report will begin with an introduction to the project, outlining the goals and objectives. It will then provide a brief overview of the methodology used, including data cleaning, processing, and analysis techniques. Finally, the report will conclude with the statistical analysis that is best suited to predict a video's popularity on youtube. We will arrive at this conclusion by delving deep into the study results.

## Objective

The objective of this project is to understand which factors determine the popularity of a video on youtube. Various statistical techniques and models have been analyzed and they have been supplemented with diagnostic metrics to determine the most effective method. For the purpose of the study, the popularity of a video is directly proportional to the number of views it receives.

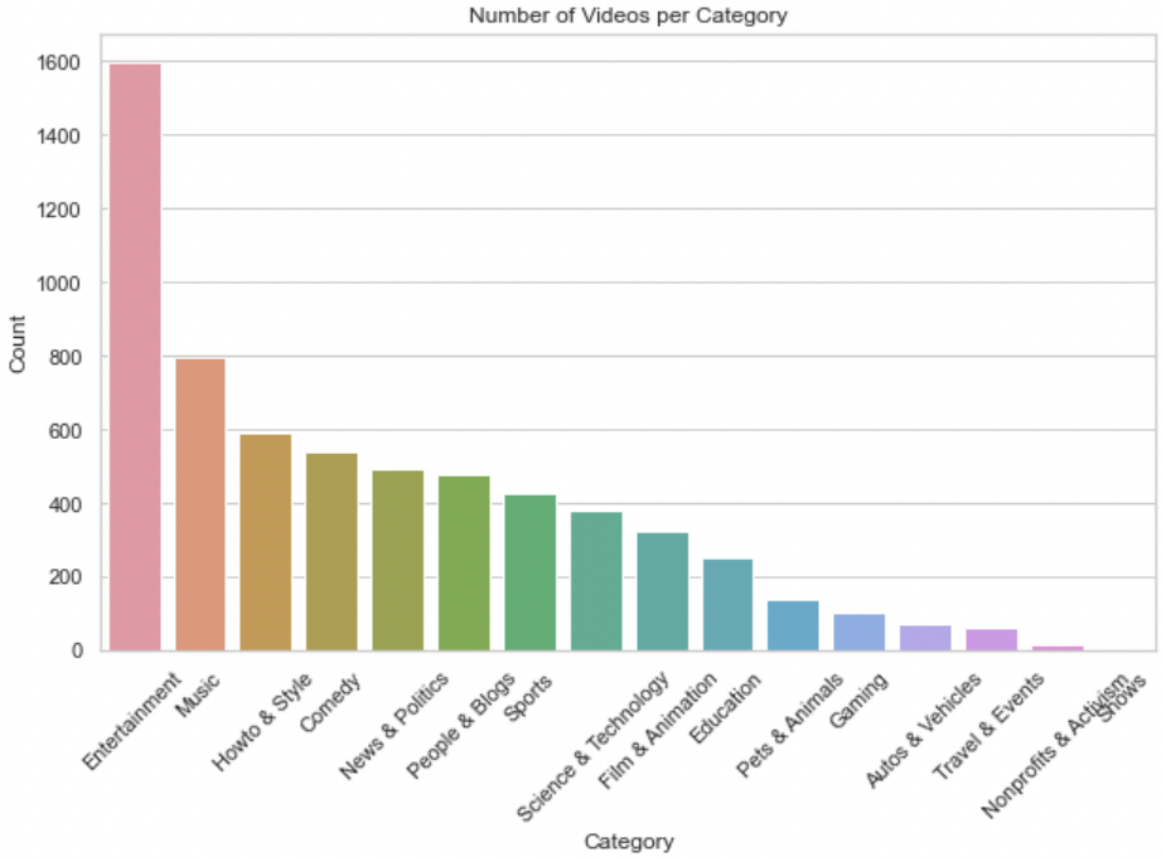
## Dataset

The dataset we used in the report is *YouTube Trending Video Dataset* and it contains information on trending videos on YouTube during the period from November 2017 to June 2018. The dataset includes over 40,000 observations with 18 variables, including video title, category, view count, likes, dislikes, publish time, comment count, and more. The data was collected through the YouTube API and contains information from various countries, including the United States, Canada, and the United Kingdom.

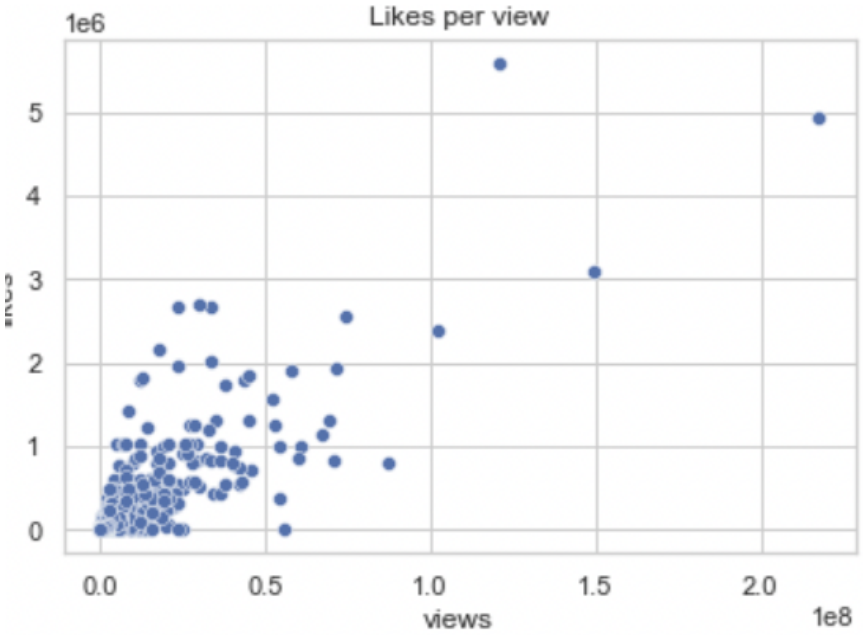
## Exploratory Data Analysis

For the purposes of simplicity, we have only used US data for our analysis. The first step was to perform data cleaning and processing to check for missing data, duplicates, outliers and consistency. The preliminary analysis showed that there were multiple duplicates in the dataset due to the redundant nature of video IDs. We subsetted our analysis to only consider the latest published videos, i.e. `publish_time` to solve this problem. The final cleaned dataset consisted of 6438 rows and 18 variables. Summary statistical analysis was done and it was noted that likes and views have right skewed distributions. Additionally, the entertainment category accounted for the highest percentage of total videos.

# What makes a Youtube Video Viral?




The correlation coefficient between views and likes was 0.832.

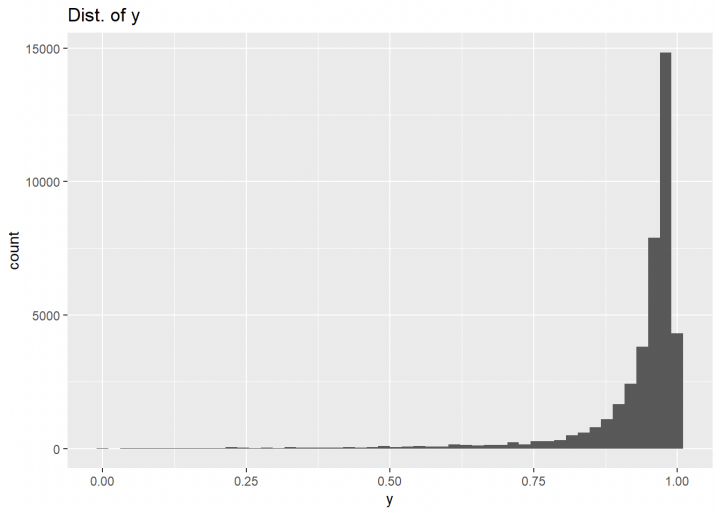


# What makes a Youtube Video Viral?

We chose our response variable as `like_ratio` which is defined as

$$Like\ Ratio\ for\ Video_i = \frac{\#\ Likes}{\#\ Likes + Dislikes}$$

The following univariate histogram of `like_ratio` shows a highly negatively skewed distribution as there was a "long tail" of extreme outliers of heavily disliked videos (compared to the mean).




We explored some data transformations to help correct this feature of the data. If we start with the simplest model: a simple linear regression on y, we have:

$$Y = p(X) = \beta_0 + \beta_1 X \in (-\infty, +\infty)$$

The problem with this model is that the predicted values of Y may fall outside the desired range of $[0, 1]$. To fix this problem, we need to apply a transformation to the Right Hand Side that results in a range of $[0, 1]$. The logistic function is a natural choice, although other functions such as the probit (popular in econometrics) are also used. Also, logistic regression is a special case of a *Generalized Linear Model*, first unified by McCullagh & Nelder (1989) as the exponential family of distributions with the special property that they maximize the information entropy on a given dataset. Thus, the logistic function is a natural choice as it is the *canonical link function* for the Bernoulli distribution (the outcome of a single user clicking either "Like" or "Dislike" for a particular video).

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \in [0, 1]$$

Expressing the linear predictor in terms of the response variable $Y = p(X)$:

$$log\left(\frac{p(X)}{1 - p(X)}\right) = \beta_0 + \beta_1 X \in (-\infty, +\infty)$$

The Left Hand Side is the *logit* function of the probability $p(X)$, also known as the *log-odds* since $\left(\frac{p(X)}{1-p(X)}\right) \in (0, \infty)$ is the *odds ratio* (commonly used in betting markets).

Further, if we substitute in $Y = \frac{\#Likes}{\#Likes+\#Dislikes} = p(X)$ into the equation of our model, we get:

$$log(\#Likes) - log(\#Dislikes) = \beta_0 + \beta_1 X$$

This has the intuitive interpretation as a *differenced log-linear model*. It is reasonable to assume that trends on a global platform such as YouTube result in view counts that naturally increase *exponentially* over particular time periods. By taking logarithms of the count of "Likes" and "Dislikes", we are measuring the *daily rate of increase* in counts. We have attempted to model this relationship in our analysis.

**Statistical Methodologies**

1. **Linear Regression**

Linear regression is a supervised learning algorithm that compares input (X) and output (Y) variables based on labeled data. It's used for finding the relationship between the two variables and predicting future results based on past relationships. Completing a simple linear regression on a set of data results in a line on a plot representing the relationship between the independent variable X and the dependent variable Y. The simple linear regression predicts the value of the dependent variable based on the independent variable. There are certain assumptions to linear regression: Linearity, Independence, Homoscedasticity, Normality. The basic formula for a regression line is Y' = bX + A, where Y' is the predicted score (dependent variable), b is the slope of the line, X is the independent variable, and A is the Y-intercept. For multiple linear regression, this changes to:
Y'= $b_1X_1 + b_2X_2 + A$, where $X_1$ and $X_2$ are independent variables. The correlation coefficient or R-squared value helps in determining if the model is fit properly. The R-squared value ranges from 0 to 1.0, denoting zero correlation at the low end (0) and a 100% correlation at the high end (1.0).

After importing the dataset, we create a new dataset by calculating the like ratio (likes / (likes + dislikes)), and converting the `category_id` into a factor. Following this, we calculate the time it takes for a video to become trending (difference between the trending date and publish time), and extract the hour from the publish time column. The sentiments for each video are also calculated and added to the new dataset using the `sentimentr` package using lexical based sentiment analysis. We also logit transform the like ratio to make the variable more normal for better diagnostics in linear regression, called `log_ratio`. In this sense we are actually doing logistic regression however, since we are predicting the continuous probabilities, we still consider it linear regression.

**Result**

The results obtained from linear regression are interesting and a good precursor for the methodologies shown later in the paper. The predictors used in the model are

`time_to_trend` which is the time it takes the video to start trending from when it was uploaded, `hour` which is the hour of the day the video was uploaded, and `sentiment` which is the extracted sentiment score from the title of the video.

| | Degrees of Freedom | Sum Squared | Mean Squared | F value | p-value |
|---|---|---|---|---|---|
| Time to trend | 1 | 62 | 61.59 | 30.939 | 2.77e-8 |
| hour | 23 | 98 | 4.26 | 2.138 | 0.00122 |
| sentiment | 1 | 60 | 59.81 | 30.048 | 4.38e-8 |
| category | 15 | 1599 | 106.59 | 53.547 | <2e-16 |
| residuals | 6196 | 12334 | 1.99 | | |

Therefore we see that all the predictors are significant which is great since it indicates a robust model. Furthermore, the p-values for all predictors are quite small except for the hour variable. In terms of diagnostics, despite the transformation, normality is still violated however the other assumptions such as homoskedasticity are not violated. Lastly, the R-squared value for this model is 0.1285 which is low, indicating that there are perhaps other factors that explain the log ratio better or that the sentiment of the title could be better extracted which will be later explored in the NLP section.

**2. Random Forest Model**

Random Forest Model builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks.

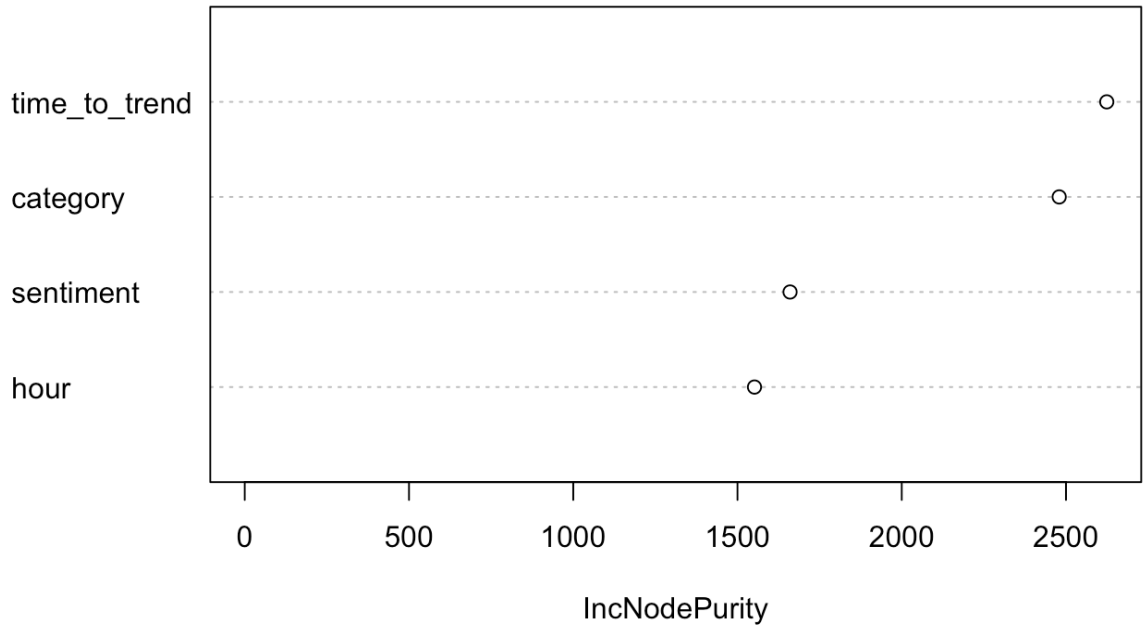
The random forest model can be used to predict the like ratio of a video based on various input features such as time to trend, hour of publish, sentiment of the title, and video category. The model is a non-parametric method that does not make any assumptions about the underlying distribution of the data. Hence, we don't have to worry about the data not being normally distributed. Instead, it builds a large number of decision trees on randomly selected subsets of the data and then averages their predictions to produce a final prediction.

In this analysis, the random forest model was used to predict the like ratio of a video. The input features used in the model were time to trend, hour of publish, sentiment of the title, and video category. The model was trained on a subset of the data and then used to predict the `like_ratio` based on all the variables. The variable importance plot shows that `time_to_trend`, `category`, and `sentiment` are the most important predictors. The

diagnostic metrics used were mean squared error (MSE) and the coefficient of determination (R-squared) values.

**Result**

Overall, this shows that `time_to_trend`, `category`, and `sentiment` are important factors in predicting the like ratio of YouTube videos. However, linear regression is not the correct model due to violation of the normality and homoscedasticity assumptions. This violates the assumptions of regression. Random Forest Models on the other hand have a low coefficient of determination: 0.0985, making it worse than the linear model.




### 3. Natural Language Processing (NLP)

NLP combines the power of linguistics and computer science to study the rules and structure of language, and create intelligent systems (run on machine learning and NLP algorithms) capable of understanding, analyzing, and extracting meaning from text and speech.

We extracted the category names from the json file to perform text analysis. There were a total of 31 unique categories. After merging csv and json files, it was observed that entertainment was the top category with 24% of the total videos.

Text analysis was performed by combining columns such as `title`, `channel_title`, `tags`, `category_name` and `description`. Two sets of analysis were performed using the `title` column and combined texts column. Data was preprocessed by tokenization, removing punctuations, stop words and other special characters. To understand the degree of popularity, the likes/views ratio was analyzed to distribute data in various bins: `not_trending`, `not_so_trending`, `trending`, `super_trending`, `super_duper_trending`. These bins are ordinal with ranks 1-5 respectively. Data was divided into 5 categories based on their

distribution - top 3% in `super_duper_trending`, next 7% in `super_trending` and so on. Likes/Dislikes ratio used in earlier analysis did not produce 5 bins as the distribution was heavy-tailed on the right.

| | index | likes | views | likes_over_views |
|---|---|---|---|---|
| 0 | count | 6237.00 | 6237.00 | 6237.00 |
| 1 | mean | 53390.30 | 1778757.15 | 0.03 |
| 2 | std | 184535.55 | 6072795.32 | 0.03 |
| 3 | min | 0.00 | 559.00 | 0.00 |
| 4 | 1% | 8.00 | 2808.96 | 0.00 |
| 5 | 10% | 455.60 | 39986.00 | 0.01 |
| 6 | 50% | 11894.00 | 501719.00 | 0.02 |
| 7 | 75% | 37777.00 | 1392722.00 | 0.04 |
| 8 | 90% | 115412.40 | 3508741.40 | 0.06 |
| 9 | 95% | 200997.40 | 6107330.80 | 0.08 |
| 10 | 97% | 320634.80 | 9617128.96 | 0.09 |
| 11 | 99% | 776950.96 | 22186355.72 | 0.12 |
| 12 | max | 5595203.00 | 217750076.00 | 0.22 |

In the first set of analysis, visualization of most frequent words was carried out for each category. In this regard, word cloud was generated using the concatenated text and only the “title” column. The below figures show the output of the “title” column as it provided better representation of categories.


***Word Cloud: Not trending videos (on the left) and Super duper trending videos (on the right)***

Overall, this shows that title words including ‘news’, ‘trailer’, ‘interview’, ‘espn’ fall in the no trending category whereas ‘bts’, ‘official’,‘video’ and ‘music’ account for immense popularity of a youtube video.

These findings were reinforced by running the topic modeling on each trending category. Gensim-LDA was used to extract hidden topics from large amounts of text i.e. from the title column. In the least popular category, the hidden topics that showed up are news and sports. Whereas, in the most popular category, “bts” and official music videos were the hidden topics.

The degree of popularity in 5 bin categories was also used as a response variable to conduct multiclass classification analysis. The dataset was split into train (70%) and test (30%) sets using random_state = 42. The combined text column was used for modeling as it resulted in better model performance. CountVectorizer and Tfidf Vectorizer were used to generate text features. GridSearchCV function helped to find best model parameters for Random Forest Classifier, Gradient Boosting Classifier, SGD and SVC Classifier. Several model iterations were performed and a model with better accuracy was chosen for prediction. Below table summarizes model iterations parameters and results on the test data. Due to the imbalanced nature of data, accuracy and weighted average are used as model evaluation criteria.

**Result**

| Model | Features | Strategy | Parameters | Accuracy | Weighted Average |
|---|---|---|---|---|---|
| Random Forest | CountVectorizer | None | {'max_depth': 8, 'n_estimators': 50} | 0.46 | 0.35 |
| SGD Classifier | CountVectorizer | None | default | 0.56 | 0.56 |
| SVC Classifier | CountVectorizer | OVR | default | 0.60 | 0.57 |
| SVC Classifier | CountVectorizer | OVR | { C =10, kernel= 'rbf'} | 0.60 | 0.59 |
| SVC Classifier | TF-IDF | OVR | {'estimator__C': 1, 'estimator__kernel': 'linear'} | 0.61 | 0.60 |
| Gradient Boosting Classifier | TF-IDF | OVR | {'max_depth': 5, 'n_estimators': 100} | 0.60 | 0.58 |
| SVC Classifier | TF-IDF + 2 Independent Vars | OVR | {'estimator__C': 1, 'estimator__kernel': 'linear'} | 0.54 | 0.47 |
| Gradient Boosting Classifier | TF-IDF + 2 Independent Vars | OVR | {'max_depth': 5, 'n_estimators': 100} | 0.61 | 0.59 |

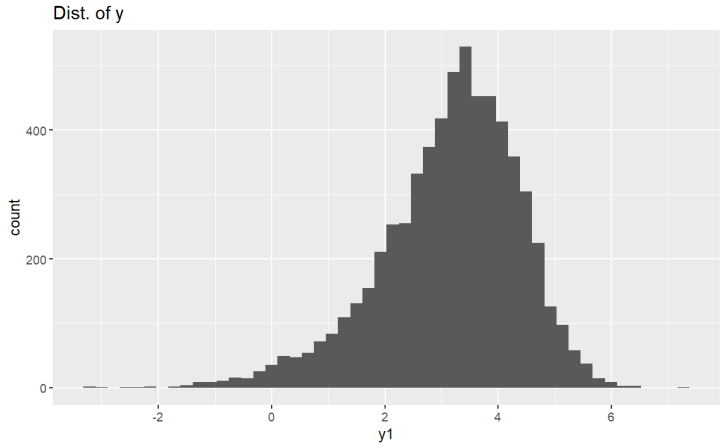
The summary table tells us that Random Forest Classifier performed poorly on the test data. The model with best results is SVC Classifier based on TF-IDF and One-vs-Rest strategy with

parameters C= 1 and kernel = “linear”. Two additional features - likes and dislikes variables were added to the existing feature set while training SVC and Gradient Boosting models. In the case of SVC, model performance deteriorated and GBC performance improved by 1%.

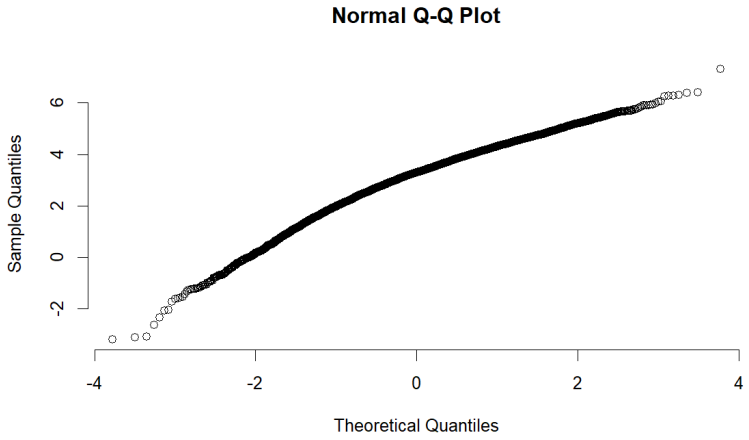
Detailed analysis of the above results can be found in the text analysis code.

### 4. Generalized Additive Model (GAM)

The last model we built was a Generalized Additive Model. The response variable Y (*like ratio*) that we wish to model falls in the range [0,1], which can be interpreted as the *probability* of a single user’s reaction to a video being a “Like” (rather than a “Dislike”).



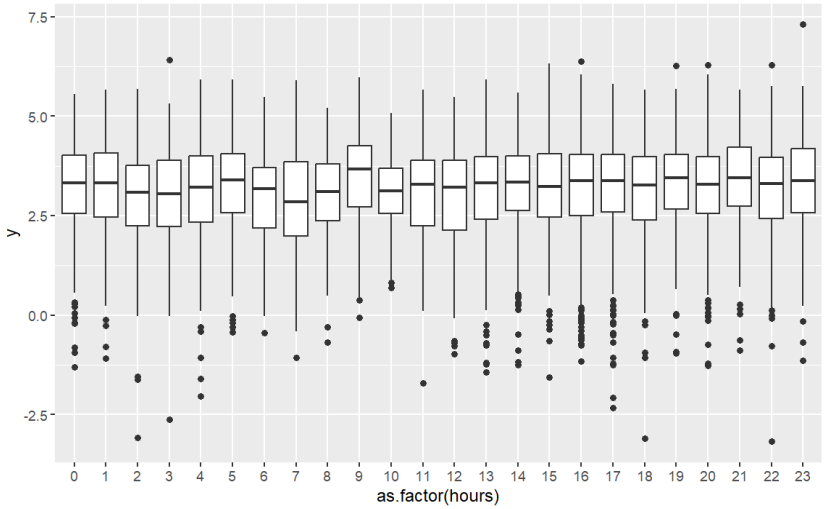

The histogram of $log\left(\frac{Y}{1-Y}\right) = log\left(\frac{\#Likes}{\#Dislikes}\right)$ has largely fixed the negative skew to give a more symmetrical distribution, similar to the Gaussian. Another advantage of the log-transform is that outliers are “*marshaled in the order of their magnitude*” as mentioned by Galton (1889) that results in a Gaussian distribution with the theoretical justification being the sum of a large number of small independent random variables. We don’t need to deliberately exclude them (potentially biasing our model) or arbitrarily apply an upper cap to them. The Q-Q plot below shows broad resemblance to the Gaussian distribution, though the quantiles don’t quite fit on an exactly straight line, which might suggest excess kurtosis (fatter tails).




Adding in Predictors:
We have a number of potentially useful predictors at our disposal, such as:

- Continuous: Time of day of video publication to YouTube platform
- Categorical: Category of video

**Time of day of video publication**




The above box-and-whisker plots of the video publication time shows the mean is roughly constant. However, the effects may be too subtle to see without fitting a model.

Hastie & Tibshirani (1986) introduced Generalized Additive Models where each linear coefficient term is replaced by a more flexible smooth function. Note that the *additivity* of each predictor is maintained (but no longer *linear* in the predictors), hence the name of the class of models. This allows the fitting algorithm to estimate each predictor independently and allows for simple interpretation of the model.

Our logistic regression model is:

# What makes a Youtube Video Viral?

$$log\left(\frac{\#Likes}{\#Dislikes}\right) = \beta_0 + \beta_1 X$$

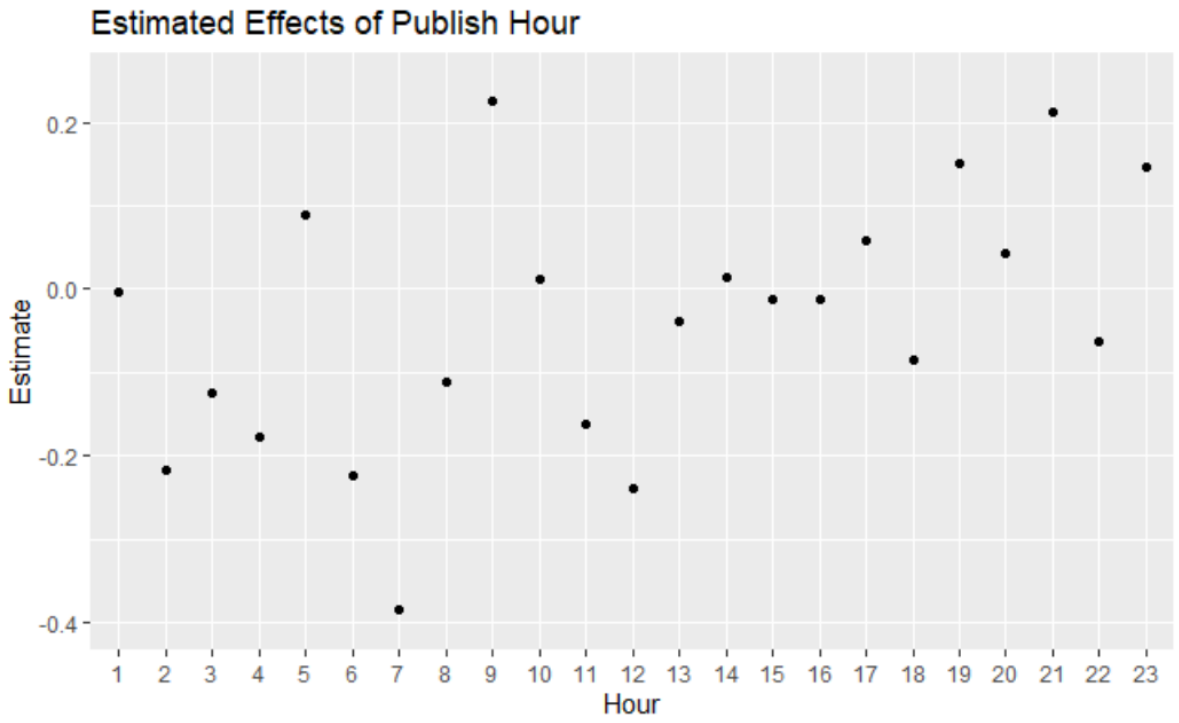
The Generalized Additive Logistic Model is

$$log\left(\frac{\#Likes}{\#Dislikes}\right) = \beta_0 + f_1(X_1)$$

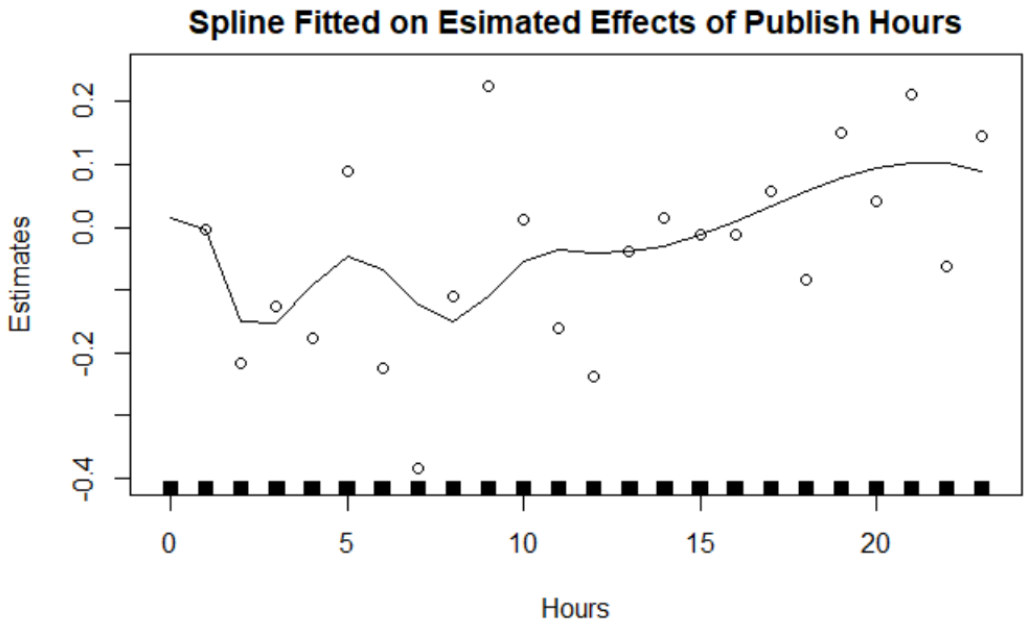
Which can be easily extended to *p* predictors:

$$log\left(\frac{\#Likes}{\#Dislikes}\right) = \beta_0 + f_1(X_1) + \ldots + f_p(X_p)$$

Below is a plot that shows the estimated effects of the hour of day a youtube video was published on our response variable. We can see that there appears to be a few trends going on here: between hour 1-7 and 8-12, there are downward trends; after hour 12, there is a pretty clear upward trend. Intuitively, this makes sense as more people engage with Youtube later on in the day.




The below plot shows the estimated GAM fitted with splines with knots at 1, 5, 8, 10, 12, 13, 15, 17, 19, 21, and 23. We have chosen these knots based on the amount of flexibility needed to capture the non-linear dependence in the data more precisely. This approach of using domain knowledge is adequate for this situation, but perhaps a more rigorous approach of cross-validation would be suitable in large-scale applications.




Turning our attention to our other predictor, the plot below shows the estimated effects on our response variable based on the category of the video.

To simplify our GAM model, we grouped together categories with very similar effects and that would make sense to bundle together contextually. For example, the categories "*Education*" and "*How to & Style*" have similar effects, but also are similar content-wise. The table below gives detail into all the groupings we created.

| buckets | title | category_id |
|---|---|---|
| cat1 | Music | 10 |
| cat1 | Pets & Animals | 15 |
| cat2 | Howto & Style | 26 |
| cat2 | Education | 27 |
| cat3 | People & Blogs | 22 |
| cat3 | Comedy | 23 |
| cat4 | Gaming | 20 |
| cat4 | Science & Technology | 28 |
| cat5 | Film & Animation | 1 |
| cat5 | Shows | 43 |
| cat6 | Travel & Events | 19 |
| cat6 | Entertainment | 24 |
| cat7 | Sports | 17 |
| cat7 | Autos & Vehicles | 2 |
| cat7 | Nonprofits & Activism | 29 |
| cat8 | News & Politics | 25 |

## Result

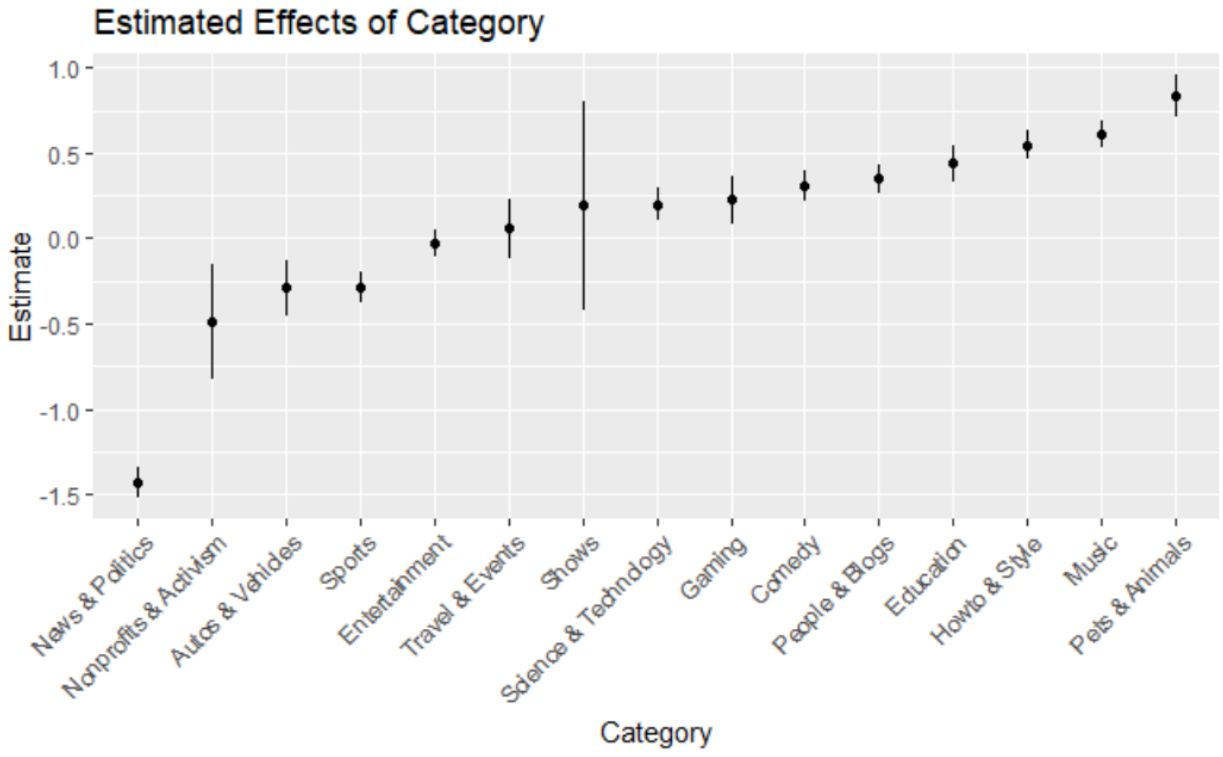
### Final Model

Putting this all together, we have:

$$log\left(\frac{\#Likes}{\#Dislikes}\right) = \beta_0 + f_1\left(Time\ of\ publication_1\right) + \sum_{i=1}^{n} \beta_i I(Video \in Category\ i)$$

### Final Model Estimates

The fitted model estimates for the $\beta_i$ coefficients are presented in the following graph, with the values presented in the following table. The parameter estimates are sorted in increasing order. For example, the *News & Politics* category has the lowest parameter estimate (and lowest *like ratio*), whereas the *Pets & Animals* category has the highest parameter estimate (and highest *like ratio*). Our model has quantified the average differences between categories and can provide some insights on how category affects the *like ratio* of a video.



| Estimate | Std..Error | t.value | Pr...t.. | category | X | category_id | title |
|---|---|---|---|---|---|---|---|
| 0.623395755 | 0.0740253 | 8.4213873 | 4.57E-17 | 10 | 2 | 10 | Music |
| 0.793125137 | 0.11336915 | 6.99595188 | 2.91E-12 | 15 | 3 | 15 | Pets & Animals |
| -0.313737583 | 0.08207995 | -3.82234123 | 1.33E-04 | 17 | 4 | 17 | Sports |
| -0.004608778 | 0.15864365 | -0.02905114 | 9.77E-01 | 19 | 6 | 19 | Travel & Events |
| -0.376073812 | 0.14847128 | -2.53297347 | 1.13E-02 | 2 | 1 | 2 | Autos & Vehicles |
| 0.154854278 | 0.12697803 | 1.21953599 | 2.23E-01 | 20 | 7 | 20 | Gaming |
| 0.326593181 | 0.08053344 | 4.05537352 | 5.07E-05 | 22 | 9 | 22 | People & Blogs |
| 0.312831936 | 0.07869656 | 3.97516683 | 7.11E-05 | 23 | 10 | 23 | Comedy |
| -0.052131012 | 0.06851351 | -0.76088664 | 4.47E-01 | 24 | 11 | 24 | Entertainment |
| -1.484816381 | 0.08005103 | -18.54837435 | 8.41E-75 | 25 | 12 | 25 | News & Politics |
| 0.536403431 | 0.07756244 | 6.91576285 | 5.12E-12 | 26 | 13 | 26 | Howto & Style |
| 0.397874865 | 0.09472997 | 4.20009478 | 2.71E-05 | 27 | 14 | 27 | Education |
| 0.169745851 | 0.08506596 | 1.9954615 | 4.60E-02 | 28 | 15 | 28 | Science & Technology |
| -0.27636834 | 0.31417196 | -0.87967219 | 3.79E-01 | 29 | 16 | 29 | Nonprofits & Activism |
| 0.204747959 | 0.55853294 | 0.36658171 | 7.14E-01 | 43 | 30 | 43 | Shows |

**Conclusion**

The GAM model might be informative for YouTube stakeholders, such as content creators posting new videos, advertisers wishing to sponsor videos, or even YouTube management in understanding user behavior. It allows for both flexibility to fit accurately to data, and interpretability to understand non-linear predictor relationships. It is rare to have this "best of both worlds" in machine learning algorithms, which tend to fall into the extreme ends on the spectrum of predictive accuracy at the expense of interpretability. We hope this modeling approach will prove valuable. In this exercise, application of text classification of YouTube video content also proved to be useful. The trained model could capture the differentiation in YouTube Videos. For future research, application of neural networks could be attempted to gauge model improvement.

**References**


- Linear Regression: https://www.mastersindatascience.org/learning/machine-learning-algorithms/linear-regression/
- Random Forest Model: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
- NLP: https://monkeylearn.com/natural-language-processing/
- GAMs
    - Galton, F. (1889) "*Natural Inheritance"* Macmillan
    - Hastie, T. & Tibshirani, R. (1986) "*Generalized Additive Models"* Statistical Science 1(3): pp. 297-310
    - Hastie, T., Tibshirani, R. & Friedman, J. (2009) "*The Elements of Statistical Learning: Data Mining, Inference and Prediction*" Springer
    - James, G., Witten, D., Hastie, T. & Tibshirani R. (2013) "*An Introduction to Statistical Learning in R*" Springer
    - McCullagh P. & Nelder, J. (1989) "*Generalized Linear Models*" Chapman and Hall

# What makes a Youtube Video Viral?

**Submitted by:**

1. Avery Tamura (akt2153)
2. Spardha Sharma (ss6343)
3. Shriniket Buche (ssb2215)
4. Simran Padam (sdp2158)
5. Milton Lim (mtl2164)