DATA 621 - Business Analytics and Data Mining

Fall 2020 - Group 2 - Final Project Paper

Avraham Adler, Samantha Deokinanan, Amber Ferger, John Kellogg, Bryan Persaud, Jeff Shamp

12/11/2020

Abstract

Using data from YouTube, we attempt to predict the number of views a video will receive using criteria such as location, category, number of likes, number of dislikes, and number of comments. Using forms of linear regression we have covered this semester, we will test various combinations of features for predictive power. YouTubers are advised to allow viewers to interact via ratings and to post content which will engender heated discussion.

Keywords: Youtube, linear regression, elastic net, R

Contents

Introduction

YouTube has changed the future of video entertainment forever [1]. In 2019, the platform was estimated to have between \$16 billion and \$25 billion in revenue [2]. YouTube's model connects a user's creativity with a desire for global recognition [3]. Before YouTube, international fame was not conceivable outside of a standard television or movie studio. Today, creators from all over the world are gaining international prominence using their own equipment and space. "YouTube is central to today's video ecosystem," says Enders Analysis research analyst Jamie McGowan Stuart [4]. The current top channel "Vlad and Nikita" earns around \$312,000 per video [5]. According to a survey from Google, 6 out of 10 people already prefer online video platforms or streaming services over live TV [6]. There are researchers predicting that by 2025—four years now—half of viewers under the age of 32 will not pay TV service [7]. Understanding some of the underpinnings of what generates views is beneficial to anyone entering, or already in, the YouTube world.

The remainder of this paper will cover a literature review, an overview of out methodology, the specifics of our modeling, a discussion of our findings, thoughts for future work, a statistical appendix, a detailed code appendix, and finally our references.

Literature Review

With the popularity of Youtube, this is not a new question. Approaches this question in the greater data science network include those based on more advanced machine learning techniques such as SGD or neural net classifiers [8]. Others leveraged NLP and specially engineered features such as "clickbait" or "NSFW" tags [9]. These attempts often used the same dataset as we are using.

Reviewing more academic literature uncovers research into the use of Support Vector Regression with various basis functions on Youtube and/or Facebook videos [10], [11]. Other attempts included building multi-stage treed regression models where the outcome of a first stage determined which specific second-stage model would be used for final popularity prediction [12].

These approaches usually added a temporal element to their analysis, and used "earlier" values to predict later views. Using more sophisticated algorithms and temporal elements tended to return statistically significant models. The downside of these approaches are their complexity and opaqueness, of course. Our approach will necessarily be simpler, although likely more transparent, being restricted to the family of linear models covered in this course and not regressing over time.

Methodology

Using the famous data set from Kaggle [13], we will explore relationships between a video's views and the number of likes, dislikes, and comments using `R` [14] and the `caret` package [15]. We may also use a video's category as predictors.

We manually scrubbed the data and discovered that the country-specific files really were not! They neither refer to videos created by country nor do they refer to views *specific* to country. Rather they are the total number of views and other predictors for that video on that day as collected by someone within that country. Meaning that aggregation is almost always multiplying. There may be some videos unique to a specific country—one which was not viewed in other countries, but they are many magnitudes smaller than those seen by all. As the United States had the most observations, we decided to analyze its data set.

With the data, we identified both numeric and factor predictors, and engineered features for convenience as well. We will use these features to investigate relationships with actual views using linear regression models.

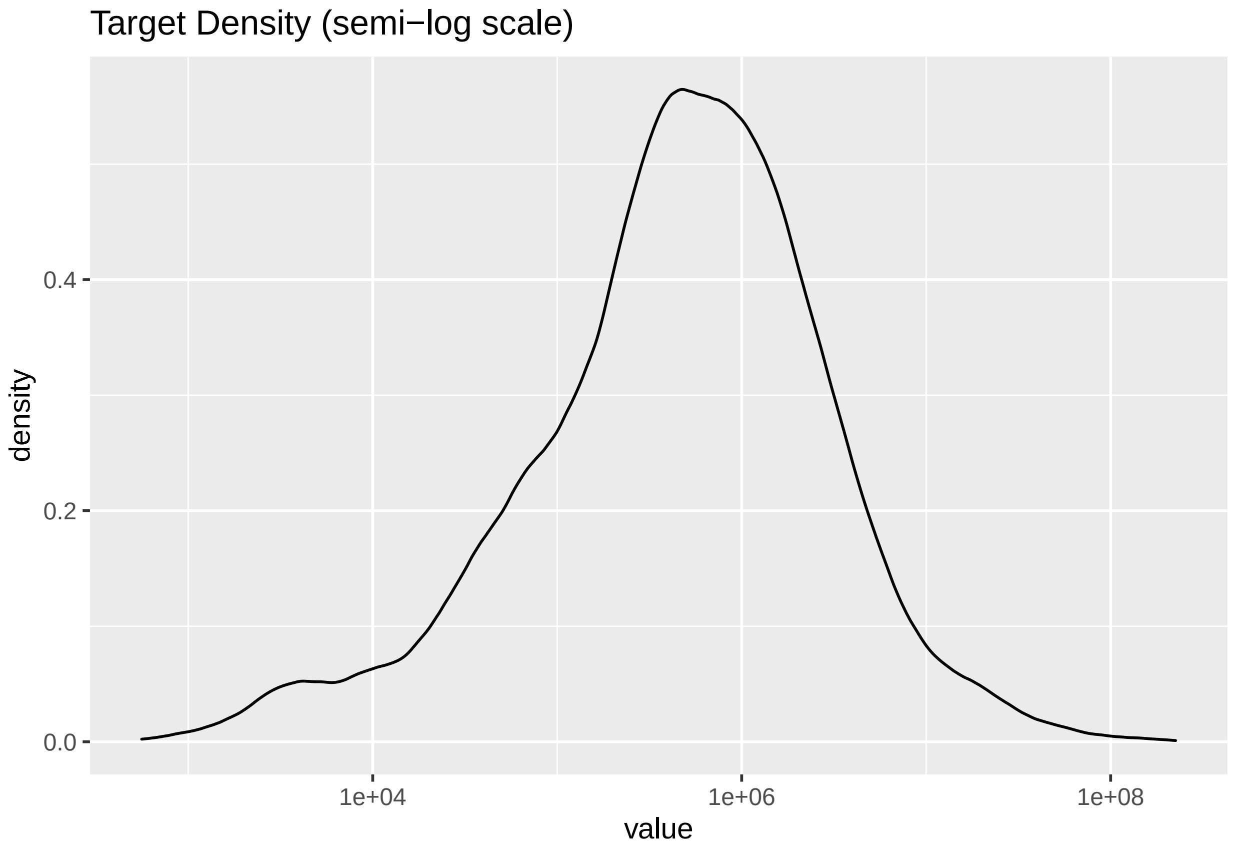
Given the models, we will compare the RMSE, R^2, and MAE on a holdout set and will select the model that performs best as the winner for this paper. We do not expect it to outperform more sophisticated models.

Experimentation & Results

Data Exploration

Since we are not factoring in time, it is incorrect to use all the observations. Therefore, we will extract the latest observation by video by country and use this subset.

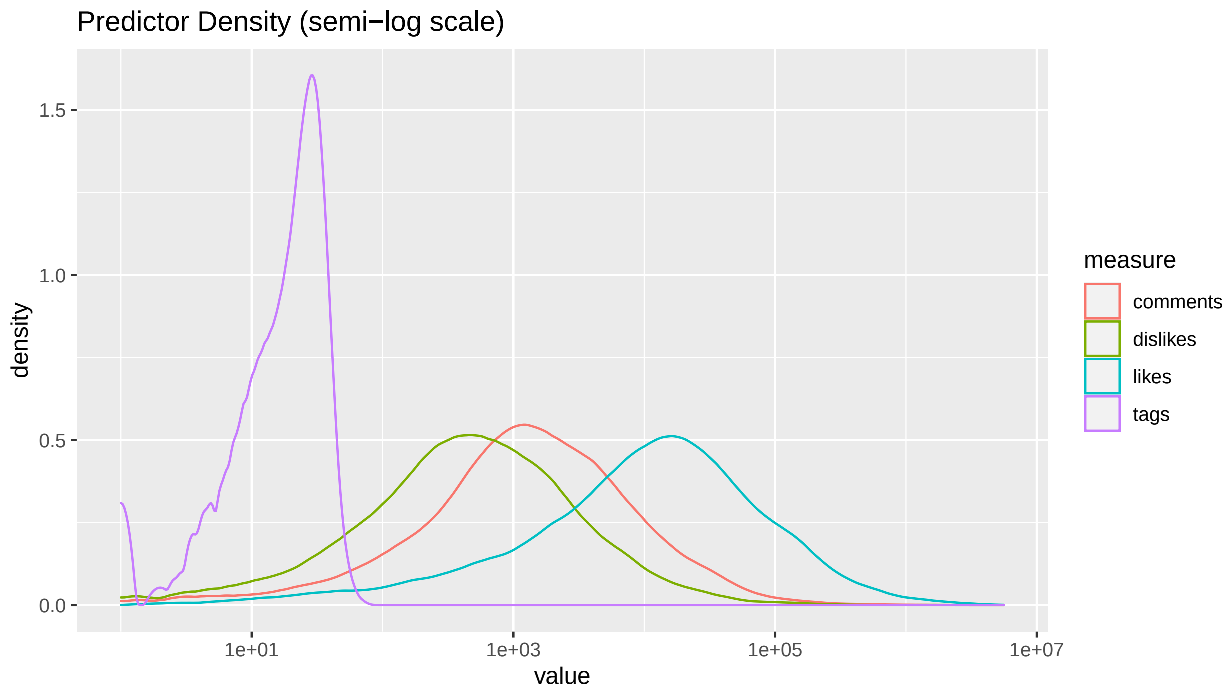
Target Variable



On a log scale, `views` looks rather Gaussian, which implies it has a lognormal distribution.

Numeric Predictors

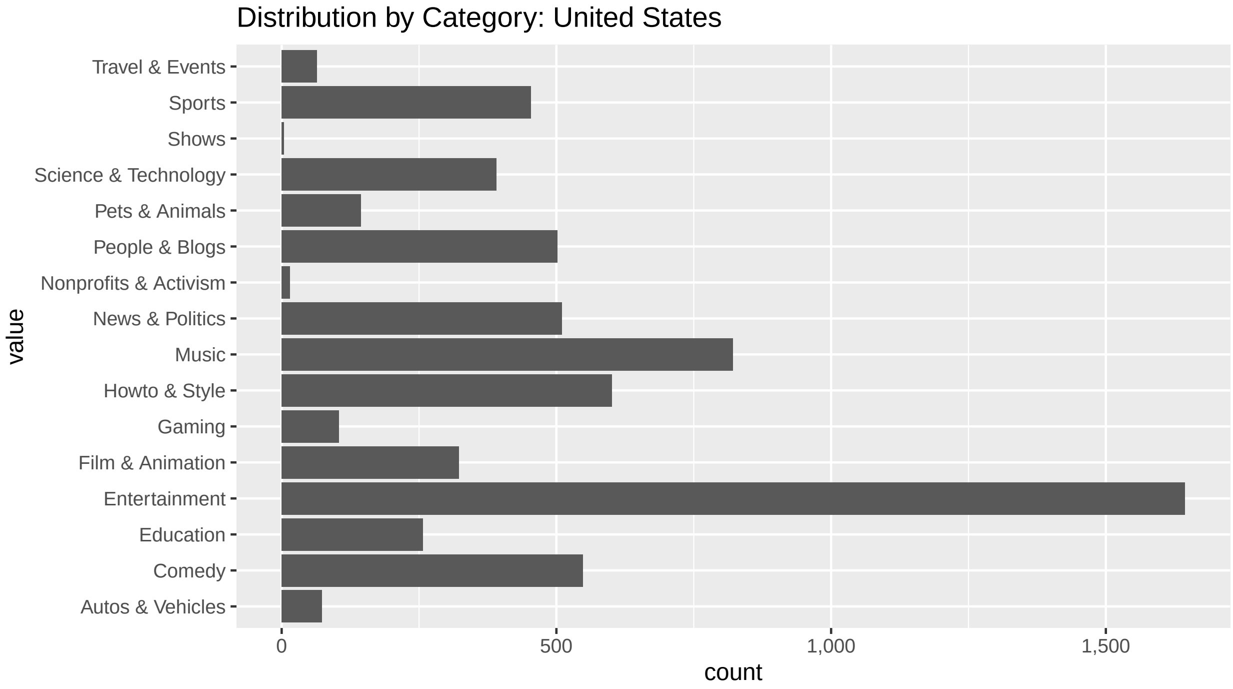
For numeric predictors, we are using the number of `likes`, `dislikes`, `tags`, and `comments`.



As expected, the number of tags is orders of magnitude less than the rating or comment variables. On average there are more likes than comments and more comments then dislikes, but all three exhibit symmetric Gaussian-like behavior.

Categories

While tabular representation of factor predictors is difficult, a distribution of videos by categories may prove informative. It is clear from the graphs below that interests vary by category. `Entertainment` seems to be the most common in the US with `Music` coming in second.



Feature Selection & Engineering

We will consider the relationship between `views` and the numerical predictors of `likes`, `dislikes`, `comments`, and `tags`. We will also consider the `category`, whether or not comments or ratings were disabled and, if there was an error with the video.

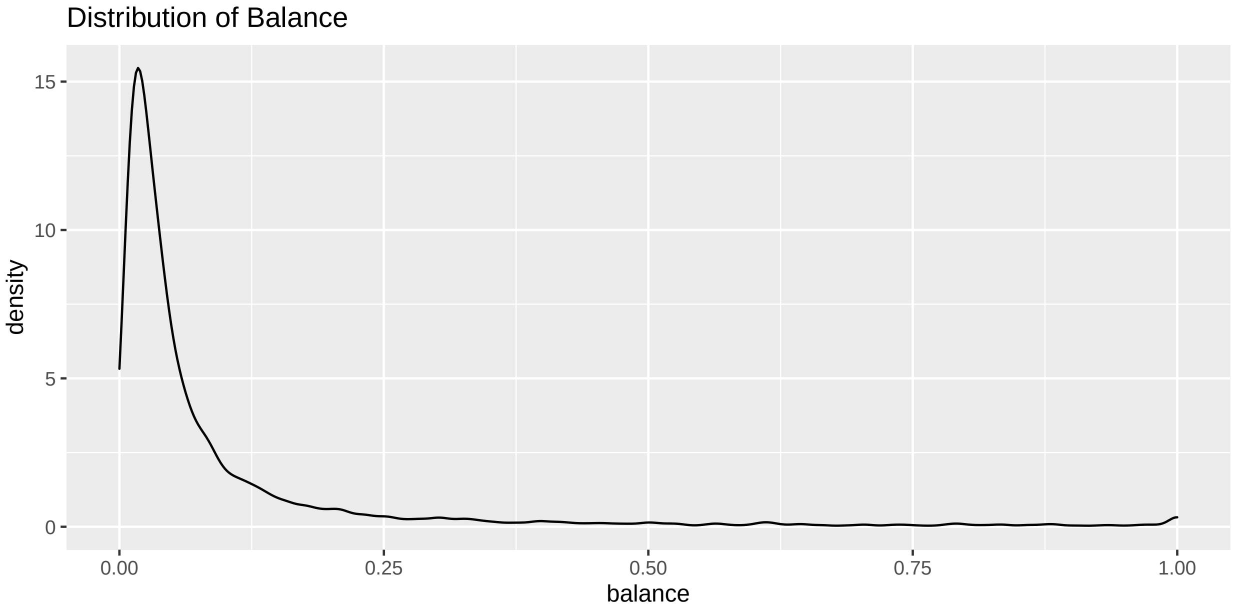
We will add two features. The first is `balance`: the ratio between the likes and dislikes. The hypothesis is that if a video is either universally loved or panned, it will get fewer views than if there is a healthy disagreement about it.

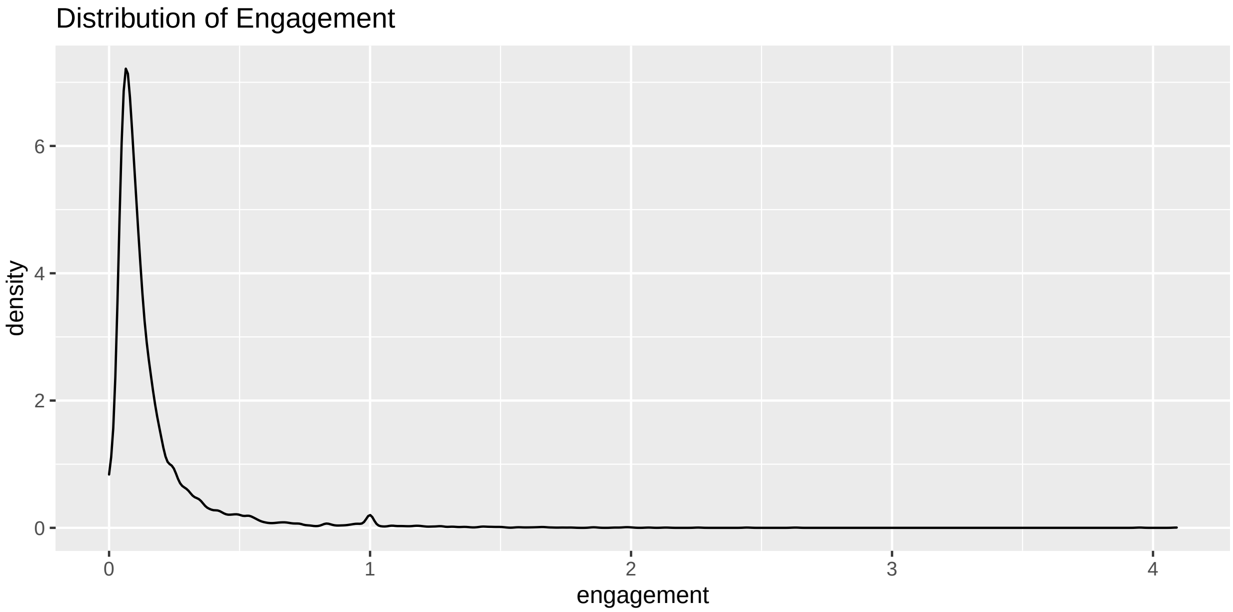
To minimize division by 0 errors, `balance` is defined as follows:

$$balance = \begin{cases} \text{ratings enabled} & \frac{\min(likes, dislikes)}{\max(likes, dislikes)} \\ \text{ratings disabled} & 1 \end{cases}$$

This constrains the ratio to the interval $[0, 1]$ with a maximum of 1 when the two are equal. Now, division by 0 can only occur when both are 0. This usually occurs when ratings are disabled. There is only 1 case out of the 6455 observations where the ratings were not disabled, yet there are neither likes nor dislikes. As this is a distinct incongruity for YouTube, we will remove that one observation from the data. When ratings are disabled, perforce there is no disparity so the ratio will be set to 1.

The second is `engagement`. This will be the ratio of comments to sum of likes and dislikes, which should give us some measure of comments to ratings. We will use a similar approach as `balance` in terms of constraining the interval to $[0, 1]$).





It's pretty clear from the distribution of `balance` that, at least for videos viewed in the US, there is a healthy dose of disagreement, as its mode is much closer to 0 than 1. Similarly, `engagement` exhibits right-skewed behavior, with a mean of 0.17 and a mode at 0.06669, yet a maximum of 4.091.

Model Building & Interpretation

We will first separate 20% of the data as a true holdout set. It is on this data that our models will be compared. We will train models on the remaining 80% of the data.

Simple Linear Regression

Linear regression may be the best known algorithm used when analyzing a continuous numeric outcome. It searches for a linear relationship of the predictors that minimizes the squared error between the "predictor" function and the observations [16].

This model will start with the numeric and logical features and the engineered disparity ratio. It will not use the exceeding mean indicator, as that is generated from the target variable, and it is felt it will distort the prediction to regress `views` on a function of `views`. The algorithm will proceed through feature selection using the AIC as the optimization metric.

Table 1: Model 1 Linear Regression Output

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.764122e+04	1.250939e+05	0.2209638	0.8251294
likes	4.089654e+01	4.074972e-01	100.3602988	0.0000000
dislikes	8.448949e+01	2.197608e+00	38.4461244	0.0000000
comments	-1.177850e+02	3.011448e+00	-39.1124309	0.0000000
tags	5.778433e+03	4.064090e+03	1.4218270	0.1551370
category.Comedy	-6.235216e+05	1.785885e+05	-3.4913861	0.0004846
category.Education	-4.287037e+05	2.456775e+05	-1.7449859	0.0810470
`category.Film & Animation`	5.093116e+05	2.207869e+05	2.3068013	0.0211054
`category.Howto & Style`	-3.073255e+05	1.707480e+05	-1.7998782	0.0719384
category.Music	-4.045870e+05	1.624886e+05	-2.4899407	0.0128077
`category.News & Politics`	-5.492536e+05	2.126931e+05	-2.5823768	0.0098396
`category.Nonprofits & Activism`	-3.112912e+06	1.031773e+06	-3.0170507	0.0025649
`category.People & Blogs`	-3.927638e+05	1.828501e+05	-2.1480105	0.0317594
cmtDisabledTRUE	6.896673e+05	3.931516e+05	1.7542022	0.0794553
rtgDisabledTRUE	1.799149e+06	7.521645e+05	2.3919623	0.0167942

	Estimate	Std. Error	t value	Pr(>|t|)
balance	7.220398e+05	3.680614e+05	1.9617376	0.0498468
engagement	9.569649e+05	2.692064e+05	3.5547635	0.0003817

The signs of these coefficients make sense in the main. Increased `likes` and `dislikes` are correlated with increased views. Of course one usually views a video at least once prior to rating it. Interestingly, increased `comments` are negatively correlated with views.

We set the baseline category to `Entertainment`, considering it was the most popular. Therefore we expected negative coefficients for other categories found significant. We were surprised that `Film & Animation` was an exception. The "worst" category predictor by magnitude is clearly `Nonprofits & Activism`.

The factor predictors tend to have much higher magnitude coefficients than do the numeric ones. This makes sense. The amount of `likes`, `dislikes`, and `comments` are many orders of magnitude greater than 1. Therefore, their coefficients can be much smaller. A Boolean variable is either 1 or 0, therefore its coefficient is much greater even if its actual contribution is lower.

When ratings are disabled, there is a bump to views. This is probably because the intercept is artificially low due to the predictive power of `likes` and `dislikes`. To balance that when there are none needs a big boost.

What may be most interesting is that the disparity and engagement ratios are powerful and significant indicators. Conflict seems to be good for Youtube videos. The more a video is argued over, the more views it seems to get!

Generalized Linear Model: Gaussian and Poisson Errors with Log Link

The generalized linear model (GLM) is an extension of the simple linear model, but the errors can be distributed per any member of the exponential family and the relationship between some function of the predictors—called the link—and the mean needs to be linear, not that the mean itself must be linear in the predictors [17].

The models under consideration here assume either a Poisson or Gaussian distribution of the errors, but a multiplicative relationship between the mean and the predictors. This is expressed by using a log link function. This is **not** the canonical link function for the GLM distributions, but as we are using numerical methods there is no issue.

The first approach *appears* similar to that of the common technique of performing a standard linear regression on the logs of the observations, but it is different. As per [18], one approach *"... log transforms observed values, while the second one log transforms the expected value.... the key difference being the relation between the predicted value and the variance."*

The second approach is the classic Poisson regression.

Table 2: Model 3a (Gaussian) Linear Regression Output

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	14.2861037	0.0872818	163.677883	0.0000000
likes	0.0000008	0.0000000	55.433632	0.0000000
dislikes	0.0000012	0.0000001	15.777013	0.0000000
comments	-0.0000010	0.0000001	-8.972967	0.0000000
tags	0.0219640	0.0017204	12.766911	0.0000000
category.Comedy	-0.4133942	0.1717724	-2.406640	0.0161351
category.Education	-1.2147352	0.6284152	-1.933014	0.0532893
`category.Film & Animation`	0.4222149	0.1249512	3.379037	0.0007328
category.Gaming	0.5844522	0.2129377	2.744710	0.0060775
`category.Howto & Style`	-0.4969620	0.1888352	-2.631724	0.0085206
category.Music	1.3639562	0.0678751	20.095077	0.0000000
`category.News & Politics`	-1.4327263	0.6924289	-2.069131	0.0385836
`category.Nonprofits & Activism`	1.0852467	0.2690092	4.034236	0.0000556
`category.People & Blogs`	-0.2928011	0.1691947	-1.730557	0.0835907

	Estimate	Std. Error	t value	Pr(>|t|)
category.Pets & Animals	-1.2461746	0.7503116	-1.660876	0.0967992
category.Science & Technology	-0.4371828	0.2569025	-1.701746	0.0888635
cmtDisabledTRUE	-0.3688376	0.2812549	-1.311400	0.1897813
rtgDisabledTRUE	3.5045553	0.5423535	6.461755	0.0000000
balance	1.7018128	0.1889399	9.007163	0.0000000
engagement	-4.9969487	0.5297577	-9.432517	0.0000000

Table 3: Model 3b (Poisson) Linear Regression Output

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	14.0330089	0.0000354	396,042.69231	0
likes	0.0000013	0.0000000	72,361.47923	0
dislikes	0.0000017	0.0000000	23,023.69127	0
comments	-0.0000029	0.0000000	-21,935.23055	0
tags	0.0149661	0.0000009	16,421.17353	0
category.Autos & Vehicles	0.0079983	0.0001248	64.10016	0
category.Comedy	-0.1682162	0.0000469	-3,583.50428	0
category.Education	-0.9411878	0.0000941	-10,004.32664	0
category.Film & Animation	0.5236250	0.0000457	11,455.91237	0
category.Gaming	0.6103779	0.0000745	8,191.68221	0
category.Howto & Style	-0.4695823	0.0000517	-9,078.25497	0
category.Music	0.9903774	0.0000313	31,667.29055	0
category.News & Politics	-0.7477750	0.0000811	-9,218.41818	0
category.Nonprofits & Activism	0.4944259	0.0001833	2,697.81022	0
category.People & Blogs	-0.1117497	0.0000483	-2,312.46637	0
category.Pets & Animals	-0.8372529	0.0001152	-7,269.31181	0
category.Science & Technology	-0.3022528	0.0000597	-5,060.16511	0
category.Shows	-0.4930116	0.0005801	-849.80596	0
category.Sports	0.0898222	0.0000484	1,856.22474	0
category.Travel & Events	-0.1482607	0.0001418	-1,045.78526	0
cmtDisabledTRUE	0.2614997	0.0000830	3,150.78212	0
rtgDisabledTRUE	2.0490864	0.0001653	12,396.46970	0
vidErrorTRUE	0.7412456	0.0002632	2,815.77070	0
balance	0.6266857	0.0000967	6,477.54888	0
engagement	-2.0249499	0.0001289	-15,711.33248	0

These models have more predictors than the simple Gaussian (G)LM. Moreover, the signs are different from the simple LM. The very fact that `Nonprofits & Activism` has a positive coefficient should raise concerns. These models are probably poor ones.

ElasticNet: Penalized Regression

Instead of using AIC to select features, one can make use of penalized regression. Using an L_1 penalty is the underpinnings of Lasso regression, which can perform feature selection. Using a squared error term, L_2, is at the heart of ridge regression [19]. Using both methods together is called the elastic net [20], [21]. To tune the hyperparameters, which includes the weighting between Lasso and Ridge, we will use 10-fold cross-validation.

Table 4: Coefficients of "Optimal" Elastic Net Model

	x
likes	37.929
dislikes	71.576
comments	-91.559
tags	5,327.826
category.Autos & Vehicles	452,316.793
category.Comedy	-487,626.540
category.Education	-305,462.954

	x
category.Film & Animation	529,828.004
category.Gaming	70,842.866
category.Howto & Style	-224,454.907
category.Music	-19,637.301
category.News & Politics	-289,416.657
category.Nonprofits & Activism	-3,080,371.336
category.People & Blogs	-288,332.650
category.Pets & Animals	-8,741.556
category.Sports	64,460.848
category.Travel & Events	383,240.133
cmtDisabledTRUE	562,364.076
rtgDisabledTRUE	1,991,830.971
vidErrorTRUE	-303,189.597
balance	783,932.467
engagement	450,854.439

There is no clean table of coefficients with elasticNet. Rather there is a sequence of models built behind the scenes. The coefficients of the selected model can be found through predicting them, but there are no corresponding p- or Z-values. Worse, there is no intercept returned.

Nevertheless, the *relative* magnitude and sign of the parameters are in line with our expectations. Both `likes` and `dislikes` are positively correlated with views and `comments` is negatively correlated. A few more categories join `Film & Animation` as contributing to excess views over the baseline `Entertainment`. But `Nonprofits & Activism` has the largest factor by far, and it is negative, which stands in contrast to the second model. Lastly, having a higher `balance`—ratio of likes to dislikes closer to 1— or a higher `engagement` tends to increase views. Conflict seems to be good for YouTubers!

Model Evaluation

Table 5: Model Performance on Test Set

Model	RMSE	R2	MAE
LM	3,301,299	0.675	1,068,070
GLM: Gauss+Log	5,256,922	0.175	2,186,177
GLM: Poisson+Log	4,952,845	0.267	1,838,099
ElasticNet	3,279,379	0.679	1,036,441

Discussion & Conclusions

The three models contain both intuitive and counter-intuitive results. As expected, videos with more ratings tend to have more views. However, as likes outnumber dislikes, that tends to reduce the number of views. All the models agree that having a video whose likes and dislikes are close in magnitude increases the propensity for views.

The elasticNet model performed best, and it's clear that the log link is inferior to the identity link for this data set. If there is any clear takeaways for prospective YouTubers, it would be:

- Make sure your videos will engender heated discussions. You want as many *dislikes* as likes. That seems to drive attention to your video!
- Keep your ratings enabled. Allowing users to rate your video drives attention to them!
- Perhaps disable comments, though.

- Stay **away** from activism.
- Talk about films or comics. This probably includes the anime and manga phenomena.
- Talking about cars is also good.
- Having more tags is better than having fewer.

The main limitation of this analysis is that the linear model, even with penalization, is probably not the best model for human behavior. There are likely non-linear effects and tipping points which could be much better captured by tree-based models or even SVMs with the appropriate kernel.

Also, this analysis may be valuable for someone planning a video, but it does not help them react once the video is posted, as we did not investigate any changes over time.

Lastly, we encountered problems trying to aggregate disjoint data by country without falling prey to gross overcounting.

Areas of future work would include investigating more sophisticated algorithms, analyzing not only final views but growth patterns, and either finding better sources or methods to increase the available data to include other countries without overcounting.

Appendix

Statistical Appendix

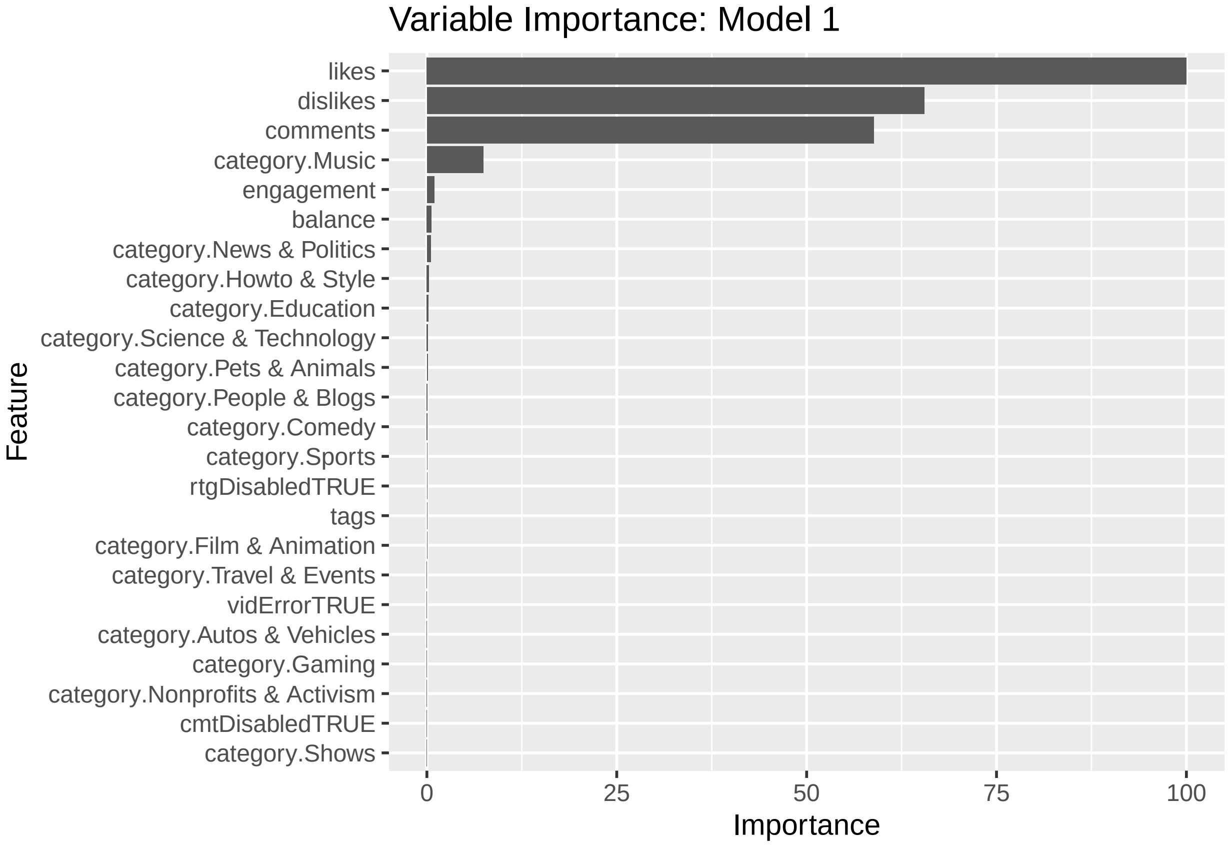
Table of Numeric Predictors

A more detailed analysis of the empirical statistics for the numeric data is found in the table below.

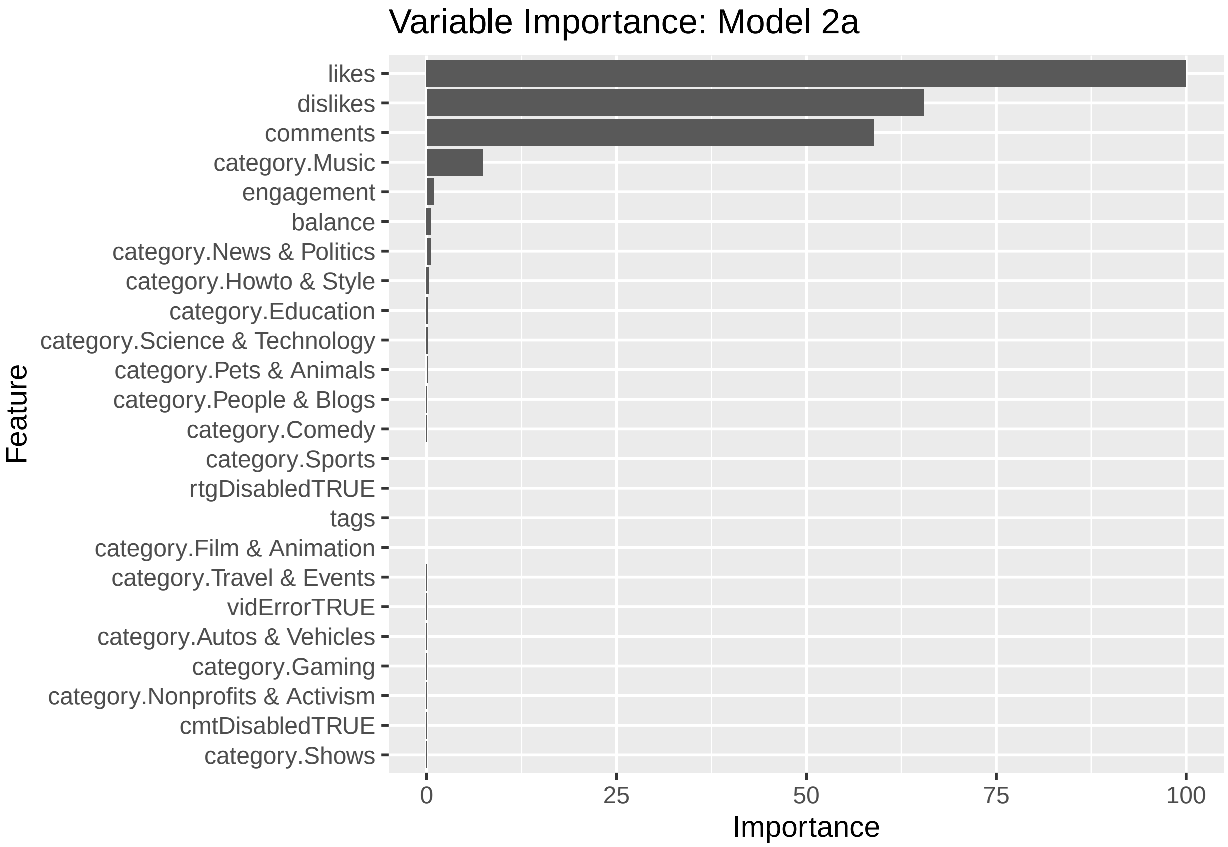
Table 6: Table of Numeric Predictors

measure	Mean	SD	Min	Q1	Median	Q3	Max	IQR
comments	6,400.96	33,565.97	0	370	1,257	4,037.5	1,361,580	3,667.5
dislikes	3,043.35	31,734.32	0	126	443	1,508.0	1,674,420	1,382.0
likes	55,164.91	192,516.35	0	2,776	11,856	38,121.5	5,613,827	35,345.5
tags	19.94	12.14	1	10	19	29.0	69	19.0
views	1,951,292.86	7,014,155.75	559	156,743	514,150	1,467,267.0	225,211,923	1,310,524.0

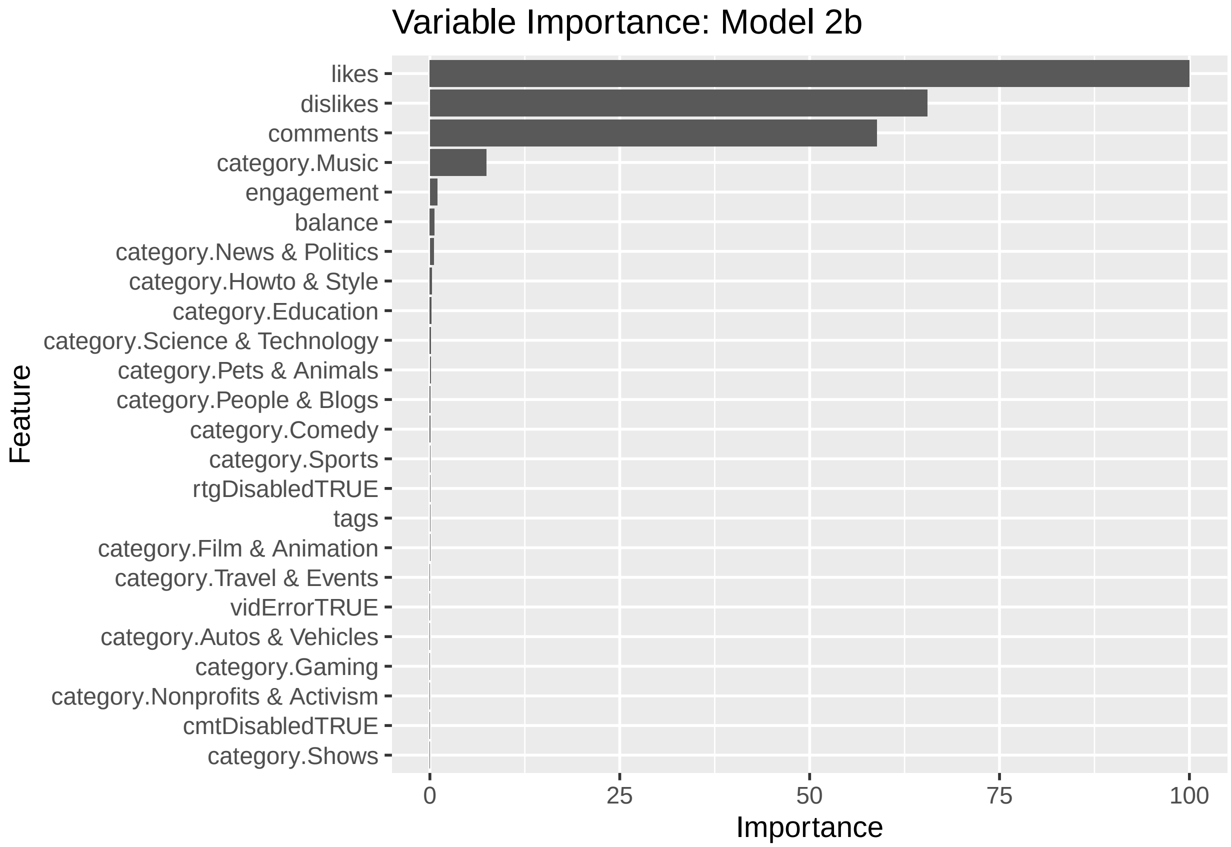
Variable Importance: Model 1



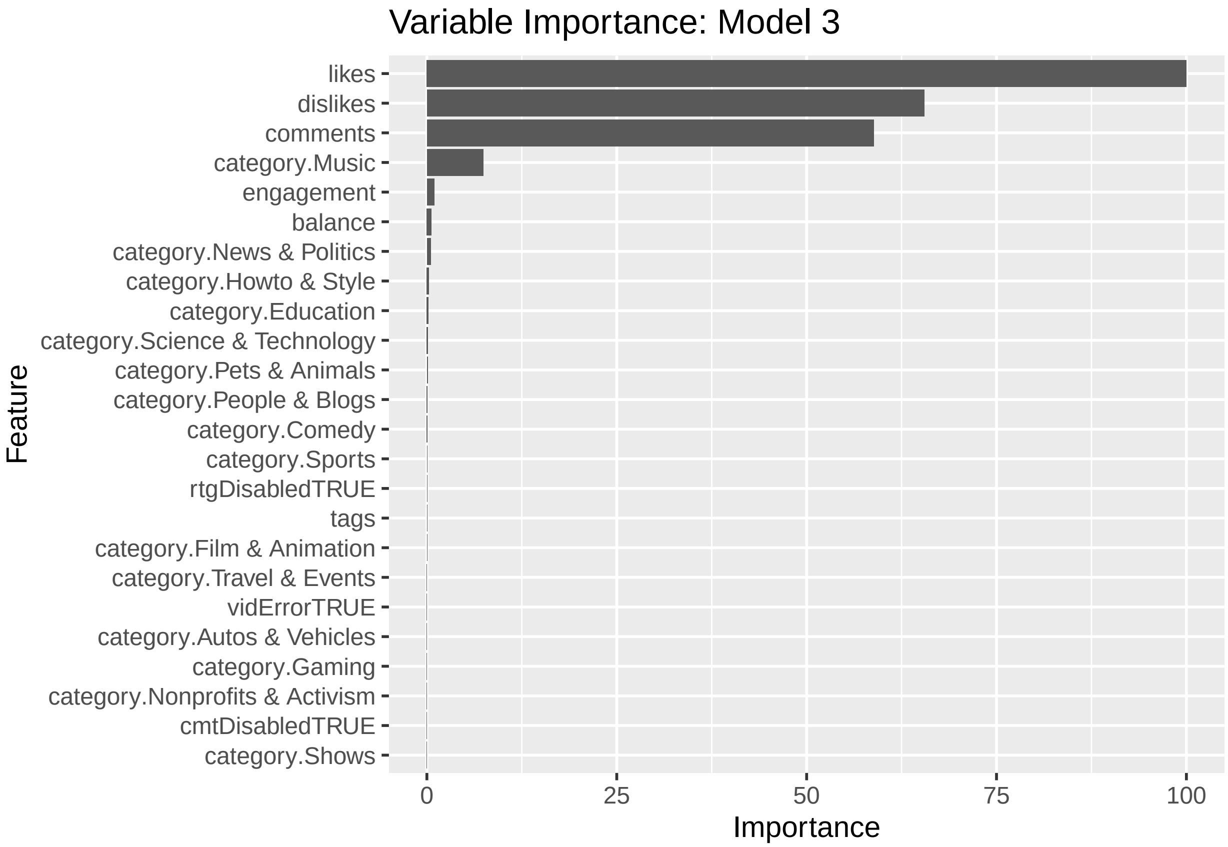
Variable Importance: Model 2a



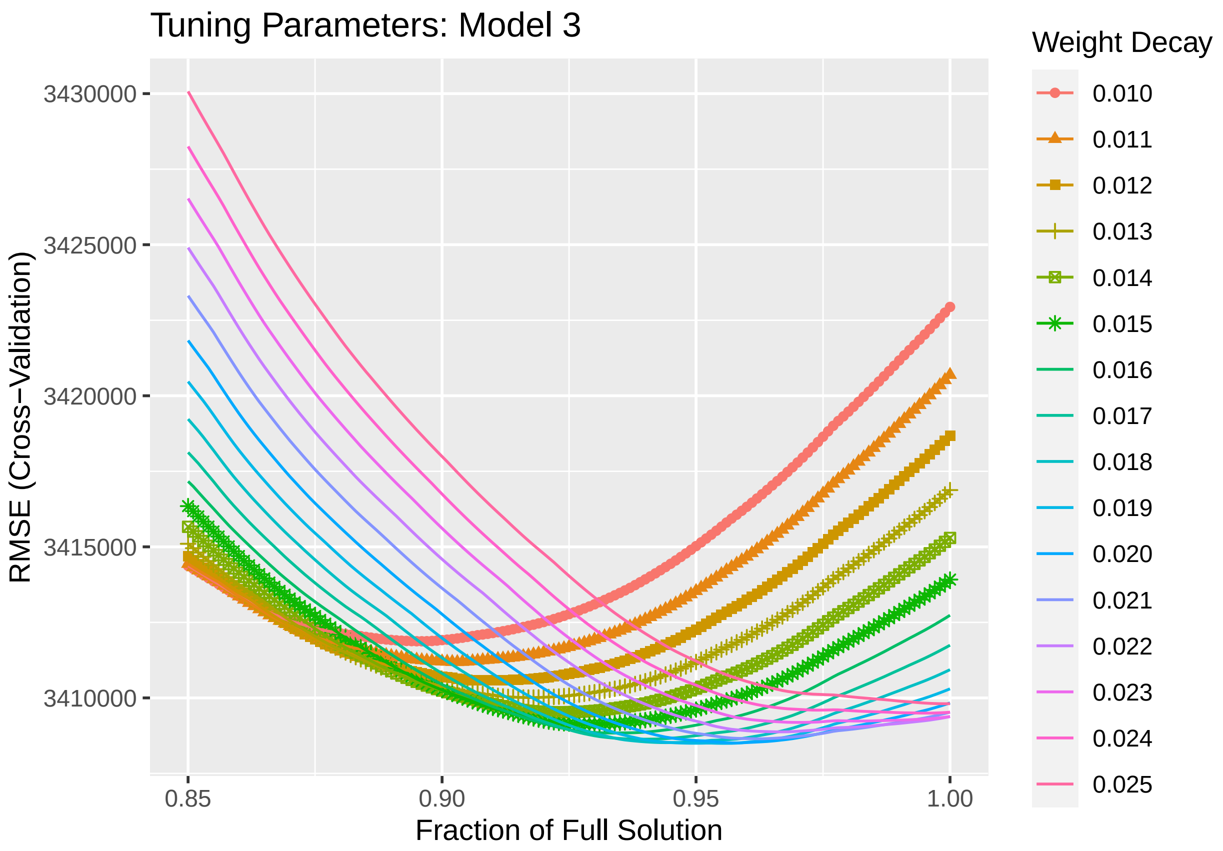
Variable Importance: Model 2b



Variable Importance: Model 3



Hyperparameter Tuning: Model 3



Code Appendix

The code chunks below represent the R code called in order during the analysis. They are reproduced in the appendix for review and comment.

```
library(jsonlite)
library(knitr)
library(stringr)
library(ggplot2)
library(scales)
library(caret)
library(data.table)
```

```
currentPath <- getwd()
dataPath <- "Data2"
us_path <- file.path(currentPath, dataPath, "USvideos.csv")

colClass <- c(rep('character', 4L), 'integer', 'POSIXct', 'character',
              rep('double', 4L), rep('logical', 3L))
us_set <- fread(us_path, encoding = 'UTF-8', colClasses = colClass)
nobsUS <- nrow(us_set)

# Category IDs
us_cat_path <- file.path(currentPath, dataPath, "US_category_id.json")
us_cats <- fromJSON(us_cat_path)
us_cats <- data.table(id = as.integer(us_cats$items$id),
                      category = us_cats$items$snippet[, 2])
setkey(us_cats, id)
```

```
# Select the row numbers of the first entry of the latest trending date by title
# and by country. There are 34 duplicates. Use this as our restricted data set
# corresponding to the most recent view count. The inner set of brackets gets
# the row number (called V1) and the outer set is a simple subset by those row
# numbers.
usExtract <- us_set[
  us_set[, .I[which.max(views)], by = c('title')]$V1
  ]
nobsUSExt <- dim(usExtract)[[1]]

# Substitute category for numeric ID through joining
usExtract <- us_cats[usExtract, on = 'id == category_id']

# Change data types for convenience, count tags and then get rid of actual tags
# and replace category IDs with actual categories
usExtract[, `:=`(trending_date = as.IDate(trending_date, format = "%y.%d.%m"),
                 views = as.double(views),
                 tag_count = 1L + str_count(tags, '[|]'),
                 id = NULL,
                 category = as.factor(category))
          ][, `:=`(tags = NULL,
                   category = relevel(category, 'Entertainment'))]

# Make name shorter for display purposes
setnames(usExtract,
         c('comment_count', 'tag_count', 'comments_disabled',
           'ratings_disabled', 'video_error_or_removed'),
         c('comments', 'tags', 'cmtDisabled', 'rtgDisabled', 'vidError'))

# Categorize the variable names
numVars <- c('views', 'likes', 'dislikes', 'comments', 'tags')

# Melt for numerics
usSetN <- melt(usExtract, measure.vars = numVars, variable.factor = FALSE,
               variable.name = 'measure', value.name = 'value')

# Melt for category
usSetF <- melt(usExtract, measure.vars = 'category', variable.factor = FALSE,
               variable.name = 'measure', value.name = 'value')

# Table of summary statistics
statsN <- usSetN[, .(Mean = mean(value, na.rm = TRUE),
                     SD = sd(value, na.rm = TRUE),
                     Min = min(value, na.rm = TRUE),
                     Q1 = quantile(value, prob = 0.25, na.rm = TRUE),
                     Median = median(value, na.rm = TRUE),
                     Q3 = quantile(value, prob = 0.75, na.rm = TRUE),
                     Max = max(value, na.rm = TRUE),
                     IQR = IQR(value, na.rm = TRUE)),
```

```r
                 keyby = c('measure')]
```

```r
ggplot(usSetN[measure != 'views'], aes(x = value, color = measure)) +
  geom_density(kernel = "epanechnikov") +
  ggtitle("Predictor Density (semi-log scale)") +
  scale_x_log10(label = scientific)
```

```r
ggplot(usSetF[measure == 'category'], aes(y = value)) +
  geom_bar() +
  ggtitle("Distribution by Category: United States") +
  scale_x_continuous(labels = comma)
```

```r
ldOutlier <- usExtract[!rtgDisabled & dislikes == 0 & likes == 0]
ldOn <- dim(ldOutlier)[[1]]
```

```r
# Remove the outlier
usExtract <- usExtract[!(!rtgDisabled & dislikes == 0 & likes == 0)]

# Add the engineered feature
usExtract[, `:=`(balance = ifelse(rtgDisabled, 1,
                               pmin(likes, dislikes) / pmax(likes, dislikes)),
                 engagement = ifelse(rtgDisabled, 1,
                                     comments / (likes + dislikes)))]
engageMode <- prettyNum(density(usExtract$engagement, kernel = 'ep')$x[
  which.max(density(usExtract$engagement, kernel = 'ep')$y)], digits = 4L)
engageMean <- prettyNum(mean(usExtract$engagement), digits = 4L)
engageMax <- prettyNum(max(usExtract$engagement), digits = 4L)
```

```r
ggplot(usExtract[, .(balance)], aes(x = balance)) +
  geom_density() +
  ggtitle("Distribution of Balance")

ggplot(usExtract[, .(engagement)], aes(x = engagement)) +
  geom_density() +
  ggtitle("Distribution of Engagement")
```

```r
# Create seen and hidden sets
set.seed(617)
seenIDX <- createDataPartition(usExtract$views, p = 0.8)$Resample1
seenSet <- usExtract[seenIDX, ]
hideSet <- usExtract[-seenIDX, ]
```

```r
# Create dummy Variables. These will be used for the next three models
modDum <- dummyVars(views ~ likes + dislikes + comments + tags + category +
                   cmtDisabled + rtgDisabled + vidError + balance + engagement,
                   data = seenSet, fullRank = TRUE)
seenX <- predict(modDum, seenSet)
seenY <- seenSet$views
```

```r
# Using stepAIC means no cross-validation. Train on entire dataset.
trC <- trainControl(method = 'none')
set.seed(181)
lm1 <- train(x = seenX, y = seenY, family = gaussian(link = 'identity'),
             method = 'glmStepAIC', direction = 'both', trace = 0,
             trControl = trC)
```

```r
kable(summary(lm1$finalModel)$coefficients,
      caption = "Model 1 Linear Regression Output",
      digts = 3L, format.args = list(big.mark = ','))
```

```r
set.seed(181)
lm2a <- train(x = seenX, y = seenY, family = gaussian(link = 'log'),
             method = 'glmStepAIC', direction = 'both', trace = 0,
             trControl = trC)
```

```r
kable(summary(lm2a$finalModel)$coefficients,
      caption = "Model 3a (Gaussian) Linear Regression Output",
      digts = 3L, format.args = list(big.mark = ','))
kable(summary(lm2b$finalModel)$coefficients,
      caption = "Model 3b (Poisson) Linear Regression Output",
      digts = 3L, format.args = list(big.mark = ','))
```

```r
trC <- trainControl(method = 'cv', number = 10L)
tG <- expand.grid(fraction = seq(0.85, 1, 0.001),
                  lambda = seq(0.01, 0.025, 0.001))
set.seed(181)
lm3 <- train(x = seenX, y = seenY,
             method = 'enet', trControl = trC, tuneGrid = tG)
```

```r
# Process the hidden set
tstDum <- dummyVars(views ~ likes + dislikes + comments + tags + category +
                   cmtDisabled + rtgDisabled + vidError + balance + engagement,
                   data = hideSet, fullRank = TRUE)
hideX <- predict(tstDum, hideSet)
hideY <- hideSet$views

# Predict using the models
lm1P <- predict(lm1, newdata = hideX)
lm2aP <- predict(lm2a, newdata = hideX)
lm2bP <- predict(lm2b, newdata = hideX)
lm3P <- predict(lm3, newdata = hideX)
# Compare the results
compTable <- data.table(Model = c('LM', 'GLM: Gauss+Log', 'GLM: Poisson+Log',
                                  'ElasticNet'),
                        RMSE = c(RMSE(lm1P, hideY), RMSE(lm2aP, hideY),
                                 RMSE(lm2bP, hideY), RMSE(lm3P, hideY)),
                        R2 = c(R2(lm1P, hideY, formula = 'traditional'),
                               R2(lm2aP, hideY, formula = 'traditional'),
                               R2(lm2bP, hideY, formula = 'traditional'),
                               R2(lm3P, hideY, formula = 'traditional')),
                        MAE = c(MAE(lm1P, hideY), MAE(lm2aP, hideY),
                                MAE(lm2bP, hideY), MAE(lm3P, hideY)))
kable(compTable, digits = 3L, format.args = list(big.mark = ','),
      caption = "Model Performance on Test Set")
```

```r
# Empirical stats Table
kable(statsN, digits = 2, caption = "Table of Numeric Predictors",
      format.args = list(big.mark = ","))
```

```r
# Model 1: VarImp
ggplot(varImp(lm1)) + ggtitle('Variable Importance: Model 1')
```

```r
# Model 2: VarImp
ggplot(varImp(lm2a)) + ggtitle('Variable Importance: Model 2a')
```

```r
# Model 2: VarImp
ggplot(varImp(lm2b)) + ggtitle('Variable Importance: Model 2b')
```

```r
# Model 3: VarImp
ggplot(varImp(lm3)) + ggtitle('Variable Importance: Model 3')
```

```r
# Model 3: Tuning paths
ggplot(lm3) + ggtitle('Tuning Parameters: Model 3')
```

References

[1] B. Moylan, “A decade of YouTube has changed the future of television.” [Online]. Available: https://time.com/3828217/youtube-decade/

[2] D. Wakabayashi, “YouTube is a big business. Just how big is anyone’s guess.” *The New York Times*, pp. Section B, Page 1 [Online]. Available: https://www.nytimes.com/2019/07/24/technology/youtube-financial-disclosure-google.html

[3] Google, “Understanding the YouTube ecosystem.” [Online]. Available: https://www.thinkwithgoogle.com/features/youtube-playbook/topic/ecosystem/

[4] A. Foster, “YouTube turns 15: Viral vlogging and altering the state of TV.” [Online]. Available: https://www.ibc.org/trends/analysis-the-youtube-revolution/5796.article

[5] “The youtube league.” [Online]. Available: https://www.cashlady.com/youtube-league/

[6] C. O’Neil-Hart and H. Blumenstein, “The latest video trends: Where your audience is watching.” [Online]. Available: https://www.thinkwithgoogle.com/marketing-strategies/video/video-trends-where-audience-watching/

[7] J. L. McQuivey, “By 2025, 50% of adults under age 32 will not pay for TV.” [Online]. Available: https://go.forrester.com/blogs/15-10-07-by_2025_50_of_adults_under_age_32_will_not_pay_for_tv/

[8] Y. Li, K. Eng, and L. Zhang, “YouTube videos prediction: Will this video be popular?” 2019. [Online]. Available: http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26647615.pdf

[9] A. Srinivasan, “Youtube views predictor.” [Online]. Available: https://towardsdatascience.com/youtube-views-predictor-9ec573090acb

[10] T. Trzciński and P. Rokita, “Predicting popularity of online videos using support vector regression,” *IEEE Transactions on Multimedia*, vol. 19, no. 11, pp. 2561–2570, 2017, doi: 10.1109/TMM.2017.2695439. [Online]. Available: https://arxiv.org/pdf/1510.06223.pdf

[11] H. Pinto, J. M. Almeida, and M. A. Gonçalves, “Using early view patterns to predict the popularity of youtube videos,” in *Proceedings of the sixth ACM international conference on web search and data mining*, 2013, pp. 365–374, doi: 10.1145/2433396.2433443.

[12] S. Ouyang, C. Li, and X. Li, “A peek into the future: Predicting the popularity of online videos,” *IEEE Access*, vol. 4, pp. 3026–3033, 2016, doi: 10.1109/ACCESS.2016.2580911. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7491235

[13] M. Jolly, “Trending YouTube video statistics,” 2019. [Online]. Available: https://www.kaggle.com/datasnaek/youtube-new

[14] R Core Team, *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing, 2020 [Online]. Available: https://www.R-project.org/

[15] M. Kuhn, *Caret: Classification and regression training.* 2020 [Online]. Available: https://CRAN.R-project.org/package=caret

[16] S. J. Sheather, *A modern approach to regression with R.* Springer Science+Business Media, 2009.

[17] J. J. Faraway, *Extending the linear model with R: Generalized linear, mixed effects and nonparametric regression models.* Chapman & Hall/CRC, 2006.

[18] A. Gelman, “Log transformations and generalized linear models.” [Online]. Available: https://statmodeling.stat.columbia.edu/2006/04/10/log_transformat/

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data mining, inference and prediction*, Second. New York: Springer Science+Business Media, Inc., 2009.

[20] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005 [Online]. Available: http://www.jstor.org/stable/3647580

[21] H. Zou and T. Hastie, *Elasticnet: Elastic-net for sparse estimation and sparse PCA*. 2020 [Online]. Available: https://CRAN.R-project.org/package=elasticnet