Multiple Linear Regression for YouTube Videos View Prediction

Xinyi Lu and Daiyan Zhang

California State University East Bay

STAT 632: Linear and Logistic Regression

Dr. Joshua Kerr

May 10, 2022

Multiple Linear Regression for YouTube Videos View Prediction

Introduction

Founded and maintained since 2005, YouTube is one of the internet's biggest platforms. With their number of videos watched per day exceeding 1 billion. YouTube videos are a great way to promote brand awareness and get your product out there. High-ranking videos can also be a great tool for pushing customers through the sales funnel at a lower cost than services like Google Ads.

When people create a YouTube video, they want to make sure they're reaching the best audience for their services and they want it to get as many views as they can because the more views, the more money and influence. The problem is people don't know which factors influence video views most. If video makers are aware of the most important factors, they can spend their time and resources cleverer and more put more efforts on optimizing the top factors. Another problem is that video makers want to estimate their income before they make the video. If video makers can predict the income, it's easier for them to get a sponsor because what the sponsors care most is the reward on investment.

This research goal is to find the significant factors that influence the views of a YouTube video and to find the best model to predict the number of video views.

Data Description

The data is "YouTubers saying things" by PRANESH MUKHOPADHYAY on Kaggle[1]. The dataset contains 2515 unique videos and their subtitles from over 91 different YouTubers, ranging from all different kinds of categories.

No	Column	Dtype	Description
1	`Id`	`str`	Unique ID for the video. (e.g., `dQw4w9WgXcQ`)
2	`Channel`	`str`	Name of the YouTube channel.
3	`Subscribers`	`str`	How many subscribers did the channel have while collecting the dataset
4	`Title`	`str`	Title of the video
5	`CC`	`int`	Did the video have manual subtitles? (Possible values `0` or `1`, where `0` means that the Transcript is auto-generated, and may be less reliable)
6	`URL`	`str`	URL of the video (e.g., `https://www.youtube.com/watch?v=dQw4w9WgXcQ`)
7	`Released`	`str`	When the video was released
8	`Views`	`str`	How many views did the video have during the collection of this dataset
9	`Category`	`str`	Category of the channel (e.g., **Science**, **Comedy**, etc)
10	`Transcript`	`str`	Subtitle for the video
11	`Length`	`str`	Duration of the video

Table 1

There are 11 columns in the dataset. Table 1 shows the variables description. The transcript column in the dataset contains the subtitles for the respective videos and CC is the indicator whether it's auto generated or made manually. The response variable would be Views, while the potential predictors would be Subscribes, CC, Released, Category, Transcript, Length. Let's inspect the data first.

There are some quantity variables in chr type, which is not statistics-friendly, and they are even in different units. For example, the Views variable is in string format and the units are different, like "10K views", "10M views". They are preferred to be in number and in the same unit in order to conduct statistical analysis. The same problems happen to Subscribers, Length and Released as well.

[1] *MUKHOPADHYAY, PRANESH.* (2022, February). *YouTubers saying things.* **https://www.kaggle.com/datasets/praneshmukhopadhyay/youtubers-saying-things**

	URL <chr>	Views <dbl>	Subscribers <dbl>	Released <dbl>	Length <dbl>
1	https://www.youtube.com/watch?v=FozCkl1xj-w	7900	6280	24	14
2	https://www.youtube.com/watch?v=lugcIAAZJ2M	11000	4590	24	8
3	https://www.youtube.com/watch?v=JiEO6F8i0eU	2300	282	36	11
4	https://www.youtube.com/watch?v=1T4XMNN4bNM	21000	17400	108	10
5	https://www.youtube.com/watch?v=0ZWGeidvrJw	8500	1590	84	5
6	https://www.youtube.com/watch?v=YiEj9mrqTN0	14000	7930	24	21
7	https://www.youtube.com/watch?v=PZFLM2DVQHs	502	389	72	34
8	https://www.youtube.com/watch?v=CoDpjqZpAh0	583	216	6	8
9	https://www.youtube.com/watch?v=VT128ElBWkM	2200	1620	48	5
10	https://www.youtube.com/watch?v=AmKX9tCVtUE	5500	5720	36	23

Table 2

Table 2 shows the cleaned data. After cleaning, Views, Subscribers, Released and Length are numbers, and Views, Subscribers are in K and Released and Length are in minute.

The next step, researchers do sentiment analysis. Firstly, researchers tokenize the word in Transcript, anti-join the stop words, and use get_sentiments("afinn") to get the AFINN value. (Note: AFINN is from -5 to 5,) then do tf-idf of words to give a weight of a word's importance in each video transcript. Lastly, researchers name "afinn_score" as the variable of the sum of the product of AFINN value and tf-idf value of Transcript, and use the same way to get Title's AFINN score: "afinn_title_score".

After all the data cleaning and sentiment analysis, the dimension of data set that will be used in this project is 2098 rows and 13 columns.

Next, let's look at the summaries and distributions of our variables.

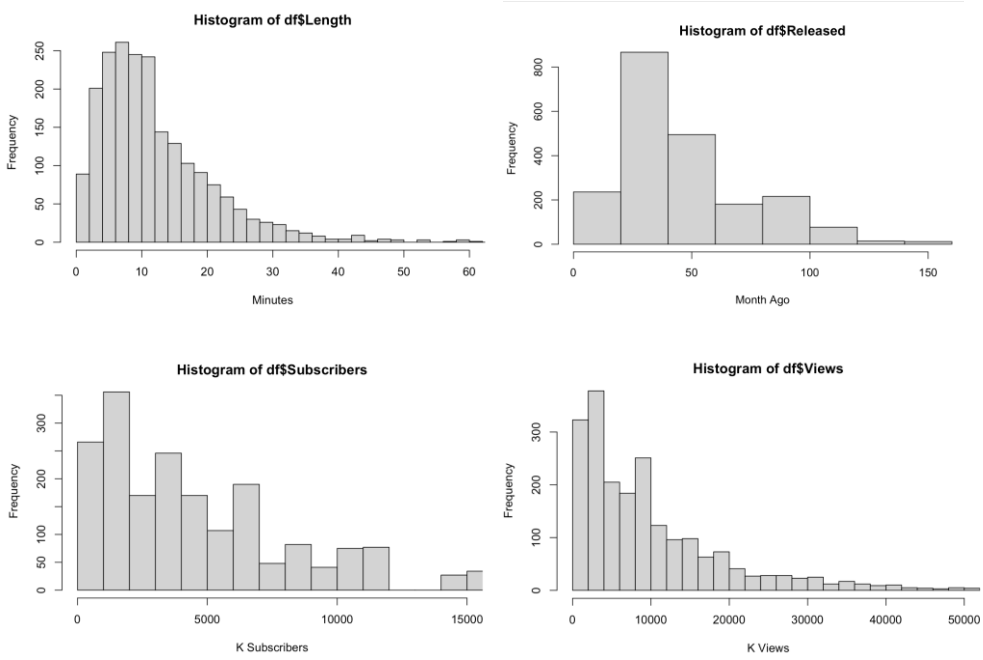

Figure 3

Figure 3 shows the distribution of variable Length, Released, Subscribers and Views are highly right skewed, which means log transformations could be applied.

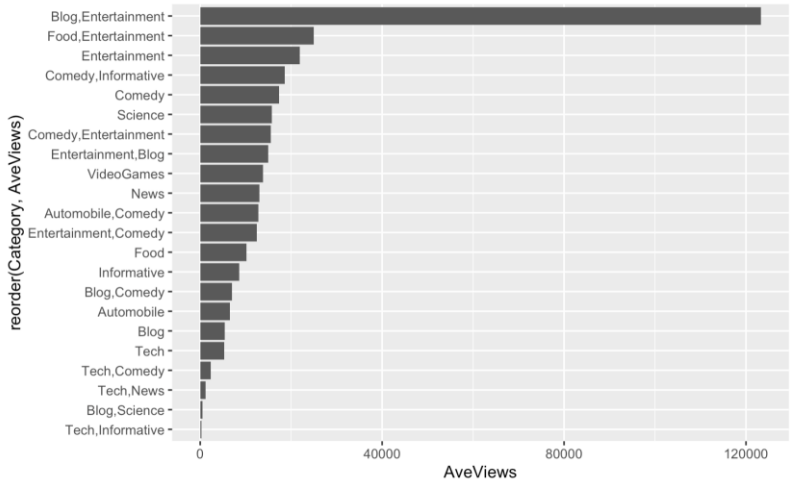

Figure 4

In Figure 4, the most top-view videos are in the Entertainment category, while less view videos are in Tech and Science categories.

Method

Purpose

This project's purpose is to use the "YouTubers saying things" data set to study what factors or variables are highly related to the number of YouTube video view, and how significant the factors affect the View. Because this data set included subtitle of each video, researchers also want to study if the sentiment of video affect the number of views.

Data Analysis Plan

Because of the project purpose, researchers want to find the best model for predicting the View of YouTube video by using the "YouTubers saying things" data set. After cleaning the data set, researchers also do sentiment analysis in Statistical Natural Language Processing by getting the product of AFINN score and value of tf-idf to get the sentiment value and add the value to data set. This project uses three types of method to find the model: Cross-Validation, Regression Tree, and Random Forests. Firstly, researchers use Cross-Validation: randomly splitting the data set in a 70% training and 30% test set to do multiple linear regression models, and use t-test to select predictors for predicting YouTube video's View. Secondly, researchers use Regression Tree find the important predictors and use the Regression Tree model to do 95% confidence interval prediction. Thirdly, researchers use Random Forests model to find the important predictors. Lastly, researchers compare three models' R squared and RMSE to find the best model for predicting the view of video.

Limitations and difficulties

In the original data set, some variables like Views, Subscribers, Released and Length are not in number and they have no standard unit, so for these variables, researcher need to do data wrangling and manipulation for later analysis.

In this project, there are some limitations and difficulties that make the study be not precise. The data set is collected by person at a point of time, so there might have some manual enter error, and data keep changing. These potential problems can affect the precise of modeling and the prediction. Besides, the data set does not include the number of "Like" or the number of "Dislike", which can represent audiences' favor of the video, and with command sense, this factor is a significant predictor. Moreover, the subtitle transcripts in the data set are people speeches or songs' lyric, which means that those subtitle transcripts are different from writing language, so when researchers do sentiment analysis, there's no 100% correct way to find the value or category to represent the sentiment of the video subtitle transcripts. The value for representing the sentiment of each video in this data set still need deeper study to be a more precise variable. Lastly, when checking assumption of multiple linear regression, researchers find that models have more than 80% of the variability in Views, on test set, is explained by predictions form the model, but the models cannot pass the assumption check.

Results

Multiple Linear Regression

After cleaning the data set and add Title and Transcript sentiment value to data set. Researchers randomly split the data into 70% training set and 30% testing set, then fit the training set to multiple linear regression model and use step wise method to select the predictors for predicting response View.

```
Call:
lm(formula = Views ~ CC + Released + Length + Subscribers + Category +
    afinn_score + afinn_title_score, data = df_train)

Residuals:
   Min     1Q Median     3Q    Max
-27366  -4376   -912   1774 297497

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                 -2.465e+02  1.422e+03  -0.173 0.862372
CC1                          9.971e+02  7.798e+02   1.279 0.201194
Released                     8.649e+00  1.310e+01   0.660 0.509263
Length                      -2.090e+01  1.859e+01  -1.125 0.260965
Subscribers                  1.547e+00  5.904e-02  26.199  < 2e-16 ***
CategoryAutomobile,Comedy    6.080e+03  2.209e+03   2.752 0.005995 **
CategoryBlog                -1.060e+03  1.843e+03  -0.575 0.565115
CategoryBlog,Comedy         -1.652e+03  2.639e+03  -0.626 0.531340
CategoryBlog,Entertainment  -1.408e+04  6.788e+03  -2.074 0.038240 *
CategoryBlog,Science         2.246e+02  5.025e+03   0.045 0.964349
CategoryComedy               1.048e+04  3.074e+03   3.409 0.000669 ***
CategoryComedy,Entertainment 1.025e+04  2.060e+03   4.974 7.35e-07 ***
CategoryComedy,Informative   1.091e+04  2.652e+03   4.115 4.09e-05 ***
CategoryEntertainment        2.867e+03  2.430e+03   1.180 0.238347
CategoryEntertainment,Blog   2.185e+03  3.123e+03   0.700 0.484194
CategoryEntertainment,Comedy 5.873e+03  2.103e+03   2.793 0.005295 **
CategoryFood                 2.909e+03  1.601e+03   1.817 0.069475 .
CategoryFood,Entertainment   1.062e+04  3.319e+03   3.200 0.001404 **
CategoryInformative          3.617e+02  1.714e+03   0.211 0.832946
CategoryNews                 1.146e+03  1.810e+03   0.633 0.526938
CategoryScience              1.608e+03  1.598e+03   1.006 0.314587
CategoryTech                -4.594e+03  1.782e+03  -2.578 0.010023 *
CategoryTech,Comedy         -6.606e+02  3.364e+03  -0.196 0.844340
CategoryTech,Informative    -9.800e+01  3.584e+03  -0.027 0.978188
CategoryTech,News           -1.991e+03  3.944e+03  -0.505 0.613668
CategoryVideoGames          -1.544e+03  1.678e+03  -0.920 0.357781
afinn_score                  1.658e+03  1.318e+03   1.259 0.208400
afinn_title_score           -1.980e+02  1.614e+02  -1.227 0.220211
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12800 on 1441 degrees of freedom
Multiple R-squared:  0.5386,    Adjusted R-squared:  0.5299
F-statistic:  62.3 on 27 and 1441 DF,  p-value: < 2.2e-16
```

Table 5: Summary table of lm_full_train

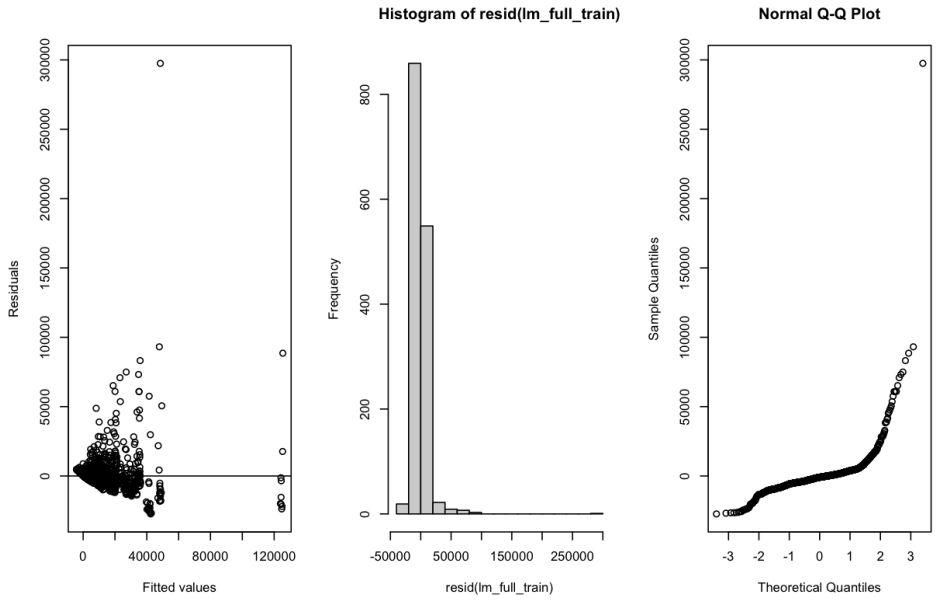

Figure 6: assumption checking for lm_full_train

Researchers first fit the predictors without transformation to training set. As Table 5, most of the predictors are not significant in this model, and the R squared value is 0.5386, which is lower than 0.6. This means that about 54% of the variability in Views, on test set, is explained by predictions form the model. After fitting model, researchers also check the assumption, but as Figure 6 shows that the residual are not constant and residual is not in normality distribution.

Because of the right skewed distribution of predictors, non-significant predictors and low R squared, and not constant and not normal distributed residuals in lm_full_train model, researchers need to do transformation.

```
Call:
lm(formula = log(Views) ~ CC + log(Released) + log(Length) +
    log(Subscribers) + Category + afinn_score + afinn_title_score,
    data = df_train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.24040 -0.35589 -0.07074  0.29884  2.66108

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   1.940526   0.139076  13.953  < 2e-16 ***
CC1                           0.077231   0.033509   2.305 0.021321 *
log(Released)                 0.033397   0.021163   1.578 0.114768
log(Length)                  -0.075007   0.019738  -3.800 0.000151 ***
log(Subscribers)              0.823654   0.014639  56.264  < 2e-16 ***
CategoryAutomobile,Comedy     0.384054   0.093746   4.097 4.42e-05 ***
CategoryBlog                 -0.298031   0.078280  -3.807 0.000146 ***
CategoryBlog,Comedy          -0.141861   0.111884  -1.268 0.205025
CategoryBlog,Entertainment    0.476976   0.198035   2.409 0.016141 *
CategoryBlog,Science         -0.655259   0.214085  -3.061 0.002249 **
CategoryComedy                0.923972   0.130505   7.080 2.25e-12 ***
CategoryComedy,Entertainment  0.867925   0.087353   9.936  < 2e-16 ***
CategoryComedy,Informative    0.743141   0.111948   6.638 4.48e-11 ***
CategoryEntertainment         0.363364   0.102284   3.553 0.000394 ***
CategoryEntertainment,Blog    0.283833   0.132486   2.142 0.032331 *
CategoryEntertainment,Comedy  0.887851   0.089202   9.953  < 2e-16 ***
CategoryFood                  0.322207   0.067960   4.741 2.34e-06 ***
CategoryFood,Entertainment    0.567903   0.141818   4.004 6.53e-05 ***
CategoryInformative           0.048899   0.072590   0.674 0.500650
CategoryNews                  0.189207   0.076760   2.465 0.013820 *
CategoryScience              -0.063591   0.067683  -0.940 0.347614
CategoryTech                 -0.630033   0.075485  -8.346  < 2e-16 ***
CategoryTech,Comedy          -0.461204   0.142387  -3.239 0.001226 **
CategoryTech,Informative     -2.032648   0.154710 -13.138  < 2e-16 ***
CategoryTech,News            -1.035872   0.167710  -6.177 8.51e-10 ***
CategoryVideoGames           -0.253051   0.070052  -3.612 0.000314 ***
afinn_score                  -0.036590   0.055782  -0.656 0.511963
afinn_title_score            -0.009828   0.006826  -1.440 0.150127
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5411 on 1441 degrees of freedom
Multiple R-squared:  0.8168,    Adjusted R-squared:  0.8133
F-statistic: 237.9 on 27 and 1441 DF,  p-value: < 2.2e-16
```

Table 7: Summary table of lm1_train

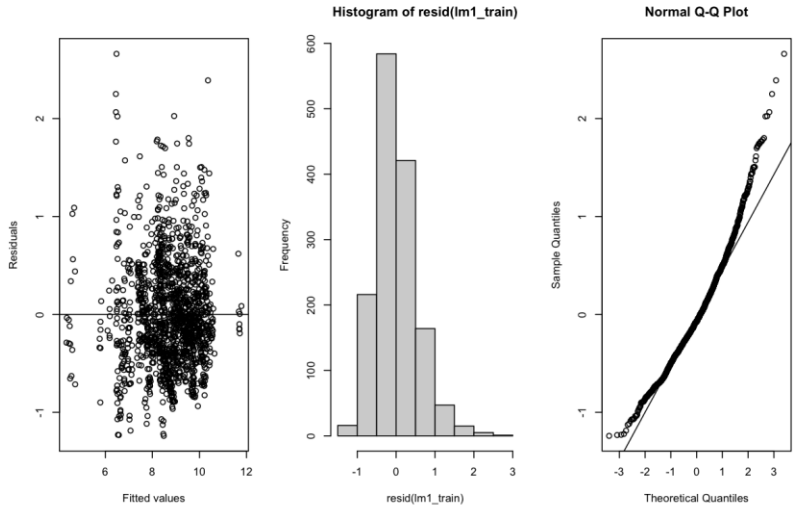

Figure 8: assumption checking for lm1_train

After Log transformation for View, Released, Length and Subscribers, fitting the data in the same training set, the new model preforms better. Except predictors: log(Released), afinn_score and afinn_title_score, other predictors are significant in the model, and there are about 82% of the variability in Views, on test set, is explained by predictions form the model. Since the model performance is good, but some predictors are not significant, researcher decide to use step wise method to select predictors.

```
> lm_step_train <- step(lm1_train)
Start:  AIC=-1776.5
log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) +
    Category + afinn_score + afinn_title_score

                    Df Sum of Sq     RSS      AIC
- afinn_score        1      0.13  422.08 -1778.06
<none>                          421.95 -1776.50
- afinn_title_score  1      0.61  422.56 -1776.39
- log(Released)      1      0.73  422.68 -1775.96
- CC                 1      1.56  423.51 -1773.09
- log(Length)        1      4.23  426.18 -1763.85
- Category          21    292.86  714.82 -1044.14
- log(Subscribers)   1    926.96 1348.91   -71.28

Step:  AIC=-1778.06
log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) +
    Category + afinn_title_score

                    Df Sum of Sq     RSS      AIC
<none>                          422.08 -1778.06
- log(Released)      1      0.72  422.80 -1777.54
- afinn_title_score  1      0.75  422.83 -1777.46
- CC                 1      1.63  423.70 -1774.41
- log(Length)        1      4.35  426.42 -1765.01
- Category          21    293.15  715.23 -1045.29
- log(Subscribers)   1    928.39 1350.47   -71.59
```

```
Call:
lm(formula = log(Views) ~ CC + log(Released) + log(Length) +
    log(Subscribers) + Category + afinn_title_score, data = df_train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.23776 -0.35991 -0.07355  0.29884  2.65881

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     1.938615   0.139018  13.945  < 2e-16 ***
CC1                             0.078774   0.033420   2.357 0.018550 *
log(Released)                   0.033287   0.021158   1.573 0.115876
log(Length)                    -0.075871   0.019691  -3.853 0.000122 ***
log(Subscribers)                0.823934   0.014630  56.318  < 2e-16 ***
CategoryAutomobile,Comedy       0.390449   0.093219   4.189 2.98e-05 ***
CategoryBlog                   -0.292888   0.077871  -3.761 0.000176 ***
CategoryBlog,Comedy            -0.135232   0.111404  -1.214 0.224988
CategoryBlog,Entertainment      0.473538   0.197927   2.392 0.016862 *
CategoryBlog,Science           -0.651934   0.213983  -3.047 0.002356 **
CategoryComedy                  0.924676   0.130475   7.087 2.14e-12 ***
CategoryComedy,Entertainment    0.870786   0.087227   9.983  < 2e-16 ***
CategoryComedy,Informative      0.753271   0.110855   6.795 1.58e-11 ***
CategoryEntertainment           0.364322   0.102253   3.563 0.000379 ***
CategoryEntertainment,Blog      0.285678   0.132430   2.157 0.031154 *
CategoryEntertainment,Comedy    0.892133   0.088945  10.030  < 2e-16 ***
CategoryFood                    0.320361   0.067889   4.719 2.60e-06 ***
CategoryFood,Entertainment      0.572298   0.141632   4.041 5.61e-05 ***
CategoryInformative             0.054002   0.072158   0.748 0.454349
CategoryNews                    0.193073   0.076518   2.523 0.011735 *
CategoryScience                -0.062650   0.067655  -0.926 0.354587
CategoryTech                   -0.630896   0.075459  -8.361  < 2e-16 ***
CategoryTech,Comedy            -0.464749   0.142256  -3.267 0.001113 **
CategoryTech,Informative       -2.034684   0.154648 -13.157  < 2e-16 ***
CategoryTech,News              -1.035821   0.167677  -6.177 8.46e-10 ***
CategoryVideoGames             -0.244047   0.068680  -3.553 0.000393 ***
afinn_title_score              -0.010696   0.006695  -1.598 0.110344
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.541 on 1442 degrees of freedom
Multiple R-squared:  0.8167,    Adjusted R-squared:  0.8134
F-statistic: 247.1 on 26 and 1442 DF,  p-value: < 2.2e-16
```

Table 9:
left: step wise selection; right: summary table of lm_step_train

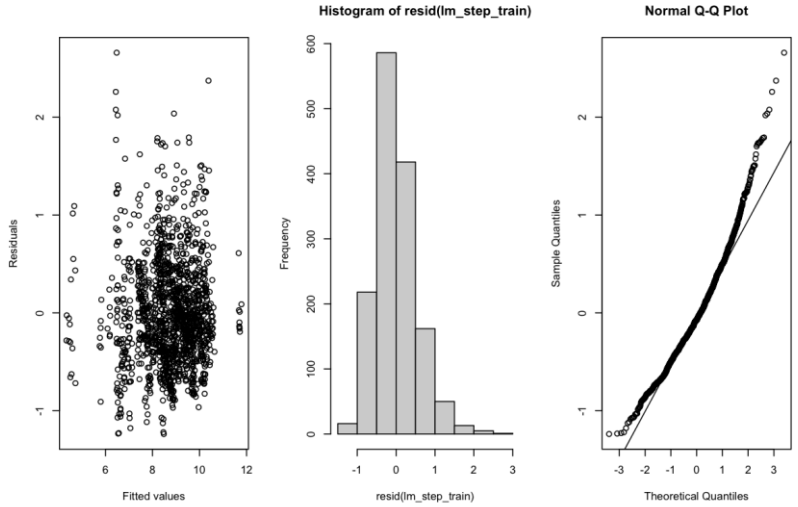

Figure 10: assumption checking for lm_step_train

Researchers do step wise selection on the lm1_tarin model, and get the selection from R. The final model's response is the log(View), and the predictors are CC, log(Released), log(Length), log(Subscribers), Category and afinn_title_score. This model also performs well. There are about 82% of the variability in Views, on test set, is explained by predictions form the model. It has a little bit higher R squared, 0.0001, and adjusted R squared, 0.0001, than the lm1_train, while the model still does not pass the assumption check.

Decision Tree

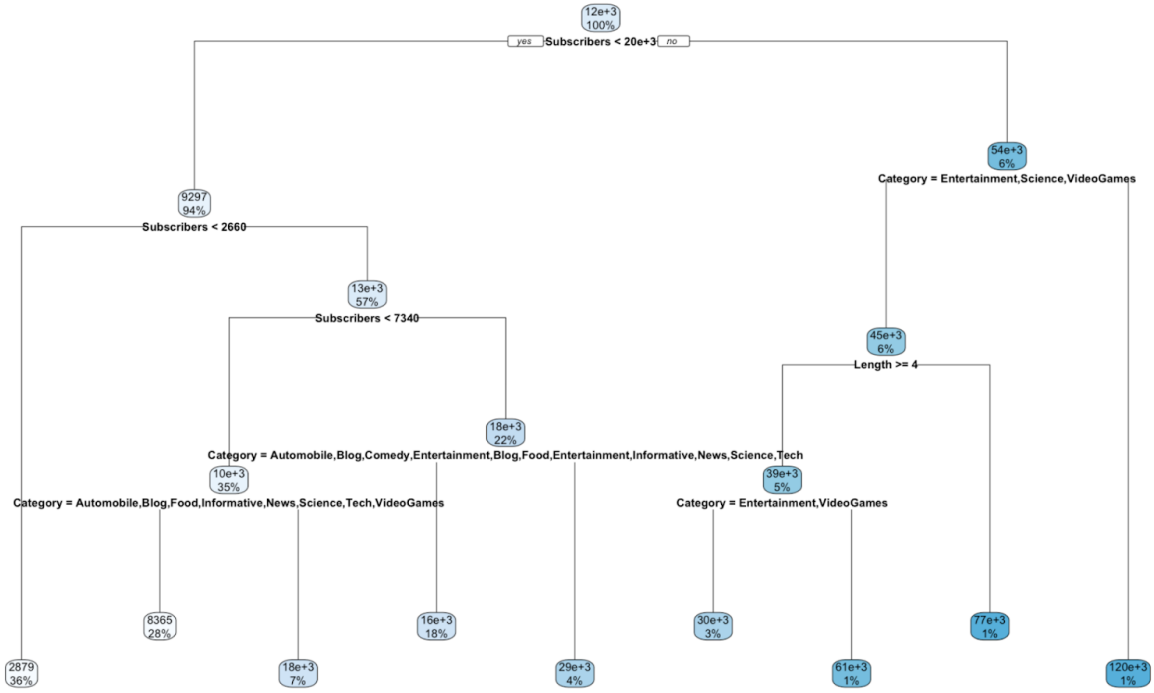

Figure 11

Figure 11 is decision tree that rpart function choses.

```
Call:
rpart(formula = Views ~ CC + Released + Category + Length + Subscribers +
    afinn_score, data = df_train, method = "anova")
  n= 1465

          CP nsplit rel error    xerror      xstd
1 0.34656079      0 1.0000000 1.0023252 0.2420648
2 0.10785793      1 0.6534392 0.6988232 0.1858269
3 0.07015559      2 0.5455813 0.6165450 0.1913220
4 0.03314964      3 0.4754257 0.4946788 0.1889961
5 0.02540655      4 0.4422761 0.4965904 0.1890720
6 0.02514482      5 0.4168695 0.4751621 0.1945831
7 0.01531699      6 0.3917247 0.4456764 0.1941797
8 0.01424796      7 0.3764077 0.4354656 0.1941017
9 0.01000000      8 0.3621597 0.4238468 0.1939854

Variable importance
Subscribers    Category      Length    Released afinn_score
         64          27           4           4           1

             Variables actually used in tree construction:
             [1] Category    Length      Subscribers
```

Table 12

Look at the variable importance table in Table 12, Subscribers, Category and length are the three most important variables, which in total contribute 95% percent of decrease in error. That's why rpart kept them in the model.

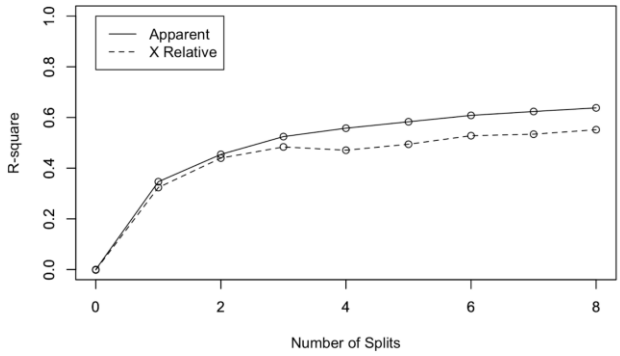

Figure 13

The R-square vs Number of Splits plot in Figure 13 give more information about how it got to the tree in Figure11. It shows how R squared grows when more nodes are added into the tree. When the tree has only the root node, the R squared is below 0.4. When the next two nodes are added, the R squared increases to near 0.6. Then more nodes are added, the R squared grows more slowly until the 8th nodes, where the rpart function decides more nodes won't increase R squared significantly and that's a balance point between model performance and interpretability.

Random Forest

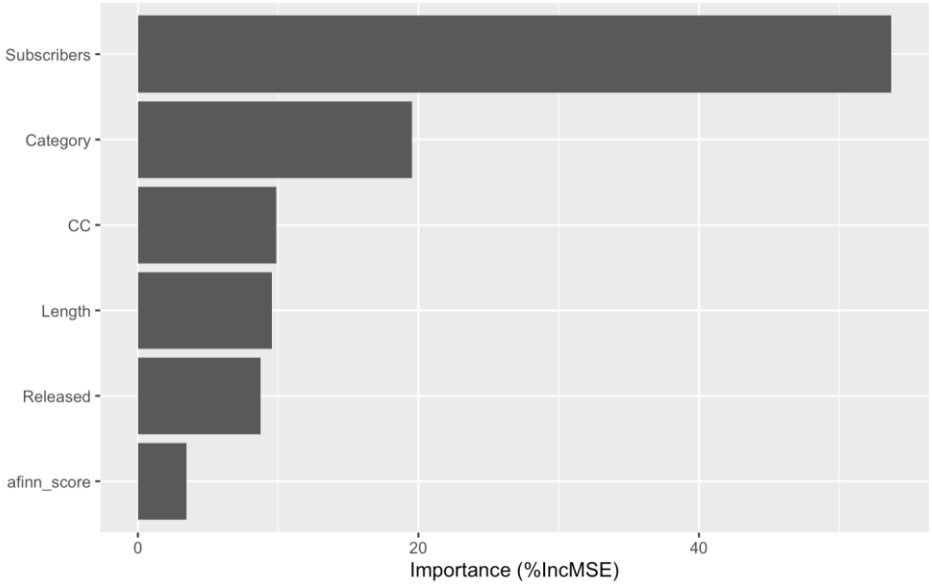

Figure 14

The variable importance plot in Figure 14 indicates that to predict the Views of a YouTube video, the most important predictor is Subscribers of Channel, and then is the Video's Category, and the Length of Video, which is in consist with the results from the decision tree model.

Conclusion

After three types of modeling for the "YouTubers saying things" data set with data wrangling and sentiment analysis for Transcripts and Title, as Table 15, researchers compare five multiple linear regression models with their R squared and adjusted R squared. Researchers conclude that the lm_step_train model preforms the best, and use it to compare with Regression model and Random Forests model.

	Model	R_squared	adj_R_squared	formula
1	lm_full_train	0.5385894	0.5299439	lm(Views ~ CC + Released + Length + Subscribers + Category + afinn_score+ afinn_title_score, data=df_train)
2	full model without transformation	0.5998046	0.5945846	lm(Views ~ CC + Released + Length + Subscribers + Category + afinn_score+ afinn_title_score, data=df1)
3	lm1_train	0.8167780	0.8133450	lm(log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_score+ afinn_title_score, data=df_train)
4	lm1	0.8154339	0.8130265	lm(log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_score+ afinn_title_score, data=df1)
5	lm_step_train	0.8167233	0.8134187	lm(log(Views) ~ CC + log(Released) + log(Length) + log(Subscribers) + Category + afinn_title_score, data = df_train)

Table 15

Table 16 shows that Random Forests model preforms the best in three models with highest R squared and lowest RMSE. In the Random Forests model, about 84% of the variability in Views, on test set, is explained by predictions form the model. At the same time, when applied to withheld data, the prediction of Views is, on average, about 6660.282 thousand off form the actually View. That is, the average error in predicting View is about 6660.282 thousand.

	Model	R_squared	RMSE
1	linear regression	0.8167233	7618.171
2	regression tree	0.7801641	7977.730
3	random forest	0.8367520	6660.282

Table 16

Code Appendix

All the code behind this project and paper is listed on GitHub

https://github.com/YenniniLu/632_Project/blob/main/YouTube.R

References

Julia Selge, David Robinson, Text Mining with R, A Tidy Approach, O'Reilly, 2019. https://www.tidytextmining.com/

Sheather. (2009). A modern approach to regression with R. Springer. https://doi.org/10.1007/978-0-387-09608-7

James, Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning : with applications in R. Springer.

MUKHOPADHYAY, PRANESH. (2022, February). YouTubers saying things. https://www.kaggle.com/datasets/praneshmukhopadhyay/youtubers-saying-things