Estimating Views

Contents

Introduction & Data

YouTube is a video-sharing website owned by Google. It allow users to upload, view, like, dislike, comment, and report videos. The videos are categorized and also have tags that are related to the video's content. There are billions of videos uploaded and many have up to billions of views. Therefore, in this problem, we will focus only on data from users in the US and **we would like to understand the factors that influence the amount of views per video**.

To derive insights and answer these questions, we take a look at a dataset containing the top trending YouTube videos in the US throughout several months. Our data has a total of 15 columns and 40003 observations, split across a training set and a test set. Each observation corresponds to a different video.

Training data: youtube_train.csv Test data: youtube_test.csv

Here is a detailed description of the variables:

- **video_id**: A number that uniquely identifies the video
- **title**: The video's title
- **views**: The amount of views the video has
- **likes**: The amount of likes the video has
- **dislikes**: The amount of dislikes the video has
- **comment_count**: The amount of comments the video has
- **logviews**: The natural logarithm of the views variable.
- **loglikes**: The natural logarithm of the likes variable.
- **logdislikes**: The natural logarithm of the dislikes variable.
- **logcomments**: The natural logarithm of the comment_count variable.
- **category_id**: A number that uniquely identifies the video's category
- **category**: The title of the video's category
- **tags**: The amount of tags the video has
- **publish_month**: The month that the video was published (1-12)
- **trending_month**: The month that the video trended (1-12)

Exercices

Problem 1 : Exploratory Data Analysis

Load youtube_train.csv into a data frame called train.

```
## 'data.frame':    30002 obs. of  15 variables:
##  $ video_id      : int  11504 31534 16360 35321 37618 1823 21123 35694 22055 18262 ...
##  $ title         : chr  "PSG 1-2 Real Madrid | RONALDO & HIS TEAMMATES IN THE DRESSING ROOM: Celebra
##  $ views         : int  1387482 507166 149879 823431 421038 1246841 1515846 606446 141285 522888 ...
##  $ likes         : int  25571 2770 19909 35691 10701 22504 34793 31776 4998 21911 ...
```

```
##  $ dislikes      : int  731 445 127 572 983 9078 1553 1407 222 856 ...
##  $ comment_count : int  1392 675 1315 4098 1551 2060 3128 5650 858 1558 ...
##  $ logviews      : num  14.1 13.1 11.9 13.6 13 ...
##  $ loglikes      : num  10.15 7.93 9.9 10.48 9.28 ...
##  $ logdislikes   : num  6.59 6.1 4.84 6.35 6.89 ...
##  $ logcomments   : num  7.24 6.51 7.18 8.32 7.35 ...
##  $ category      : chr  "Sports" "News & Politics" "People & Blogs" "Howto & Style" ...
##  $ category_id   : int  17 25 22 26 28 1 24 26 24 23 ...
##  $ tags          : int  160 51 324 382 184 299 394 164 469 363 ...
##  $ publish_month : int  3 5 12 4 5 4 11 4 2 1 ...
##  $ trending_month: int  3 6 12 5 5 4 12 4 3 1 ...

## 'data.frame':    10001 obs. of  15 variables:
##  $ video_id      : int  10 13 18 23 24 33 37 39 50 51 ...
##  $ title         : chr  "Everything Wrong With Birdman In 13 Minutes Or Less" "Stop Motion CHALLENGE
##  $ views         : int  945670 162827 48179 148191 3886748 16381551 1898712 89311 353484 527933 ...
##  $ likes         : int  21927 6862 4603 8911 44165 171508 83140 599 24968 3624 ...
##  $ dislikes      : int  832 157 92 267 3070 5428 2717 26 344 134 ...
##  $ comment_count : int  2131 813 446 326 4451 12736 8001 52 2271 494 ...
##  $ logviews      : num  13.8 12 10.8 11.9 15.2 ...
##  $ loglikes      : num  10 8.83 8.43 9.1 10.7 ...
##  $ logdislikes   : num  6.72 5.06 4.52 5.59 8.03 ...
##  $ logcomments   : num  7.66 6.7 6.1 5.79 8.4 ...
##  $ category      : chr  "Film & Animation" "Film & Animation" "Film & Animation" "Film & Animation"
##  $ category_id   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ tags          : int  103 368 83 91 359 409 412 184 438 7 ...
##  $ publish_month : int  2 3 11 2 1 5 5 11 1 4 ...
##  $ trending_month: int  3 3 11 2 1 6 5 12 1 5 ...
```

1.1 How many rows are in the training dataset?

```
## [1] 30002
```

1.2 What is the average amount of likes per video in the training dataset?

```
## [1] 75461.28
```

1.3 What is the category of the video with most views in the training set?

1.4 Which category has the least amount of dislikes in the training set?

1.5 In the training set, out of the videos with at least 1 million likes, how many have at least 100,000 comments?

Problem 2 : Simple Linear Regression

2.1

For the rest of this problem, we will be working with *log(views)*, *log(likes)*, *log(dislikes)*, and *log(comment_count)*, which helps us manage the outliers with excessively large amounts of views, likes, dislikes, and comments. The values of log(views), log(likes), log(dislikes), and log(comment_count) are found in the columns logviews, loglikes, logdislikes and logcomments, respectively.

```
##
## Call:
## lm(formula = logviews ~ loglikes + logdislikes + logcomments,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3446 -0.4088 -0.0453  0.3395  4.7223
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.484528   0.018977  341.71   <2e-16 ***
## loglikes     0.490925   0.004431  110.79   <2e-16 ***
## logdislikes  0.443719   0.004132  107.39   <2e-16 ***
## logcomments -0.096344   0.005719  -16.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6392 on 29998 degrees of freedom
## Multiple R-squared:  0.8522, Adjusted R-squared:  0.8522
## F-statistic: 5.765e+04 on 3 and 29998 DF,  p-value: < 2.2e-16
```

2.1.1 : What is the value of log(views) that our baseline model predicts?

```
## [1] 13.37152
```

2.1.2 : What is the correlation between log(views) and log(dislikes) in the training set?

```
## [1] 0.8722925
```

2.1.3 : Choose the most reasonable answer from the following statements:

1. Higher log of dislikes are associeted with higher log of views, likely because the popular videos often have many dislikes.
2. High log of dislikes are associated with less log of views.
3. There is no association between log of dislikes and log of views.

2.2

Create a linear model that predicts log(views) using log(dislikes).

2.2.1 : What is the coefficient of log(dislikes)?

```
##
## Call:
## lm(formula = log(views) ~ log(dislikes), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1050 -0.4542  0.0600  0.5057  4.6775
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.305149   0.017056   486.9   <2e-16 ***
## log(dislikes) 0.786930   0.002547   309.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8129 on 30000 degrees of freedom
## Multiple R-squared:  0.7609, Adjusted R-squared:  0.7609
## F-statistic: 9.547e+04 on 1 and 30000 DF,  p-value: < 2.2e-16
```

2.2.2 : What is the R2 on the test set?

```
## [1] 0.7519569
```

Problem 3 : Adding More Variables

3.1

As good practice, it is always helpful to first check for multicollinearity before running larger models.

Examine the correlation between the following variables: *logdislikes*, *loglikes*, *logcomments*, *tags*, *publish_month*, and *trending_month*

```
##                logdislikes    loglikes logcomments        tags publish_month
## logdislikes     1.00000000  0.81557883  0.87229943  0.09075122   -0.09935314
## loglikes        0.81557883  1.00000000  0.90305086  0.06599080   -0.09816579
## logcomments     0.87229943  0.90305086  1.00000000  0.07269055   -0.09905996
## tags            0.09075122  0.06599080  0.07269055  1.00000000   -0.03003043
## publish_month  -0.09935314 -0.09816579 -0.09905996 -0.03003043    1.00000000
## trending_month -0.07815380 -0.06608009 -0.06582680 -0.02377686    0.89936110
##                trending_month
## logdislikes       -0.07815380
## loglikes          -0.06608009
## logcomments       -0.06582680
## tags              -0.02377686
## publish_month      0.89936110
## trending_month     1.00000000
```

Which of the following pairs of variables have correlation with magnitude above 0.8? Select all that apply.

1. **logdislikes, loglikes**
2. **logcomments, logdislikes**
3. tags, logcomments
4. trending_month, tags
5. **publish_month, trending_month**
6. **logcomments, loglikes**

3.2

Create a linear model that predicts log(views) using the following variables: *logdislikes*, *tags*, and *trending_month*.

We have excluded *loglikes*, *logcomments*, and *publish_month* due to concerns about multicollinearity.

3.2.1 : What is the value of the intercept?

```
##
## Call:
## lm(formula = logviews ~ logdislikes + tags + trending_month,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0483 -0.4588  0.0560  0.5049  4.7016
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.361e+00  1.972e-02  424.00  < 2e-16 ***
## logdislikes     7.834e-01  2.558e-03  306.25  < 2e-16 ***
## tags            1.655e-04  3.177e-05    5.21  1.9e-07 ***
## trending_month -1.409e-02  1.232e-03  -11.44  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8108 on 29998 degrees of freedom
## Multiple R-squared:  0.7622, Adjusted R-squared:  0.7621
## F-statistic: 3.204e+04 on 3 and 29998 DF,  p-value: < 2.2e-16
```

3.2.2 : What is the R2 on the test set?

```
## [1] 0.7530081
```

Problem 4 : Interpreting Linear Regression

4.1

Using the model from Problem 3, **which of the following variables are significant at a level of 0.001 (p-value below 0.001)?** Select all that apply.

```
##
## Call:
## lm(formula = logviews ~ logdislikes + tags + trending_month,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0483 -0.4588  0.0560  0.5049  4.7016
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     8.361e+00  1.972e-02  424.00  < 2e-16 ***
## logdislikes     7.834e-01  2.558e-03  306.25  < 2e-16 ***
## tags            1.655e-04  3.177e-05    5.21  1.9e-07 ***
## trending_month -1.409e-02  1.232e-03  -11.44  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8108 on 29998 degrees of freedom
## Multiple R-squared:  0.7622, Adjusted R-squared:  0.7621
## F-statistic: 3.204e+04 on 3 and 29998 DF,  p-value: < 2.2e-16
```

1. logdislikes
2. loglikes
3. tags
4. logviews
5. trending_month

4.2

Using the model from Problem 3, **how would you interpret the coefficient of tags?**

1. All else being equal, an increase in tags is associated with a 1.655e-04 increase in log(views).
2. All else being equal, an increase in tags is associated with a 1.655e-04 decrease in log(views).

4.3

Using the simple model from Problem 2, **if the amount of dislikes is 1000, how many views does the model predict the video has?**

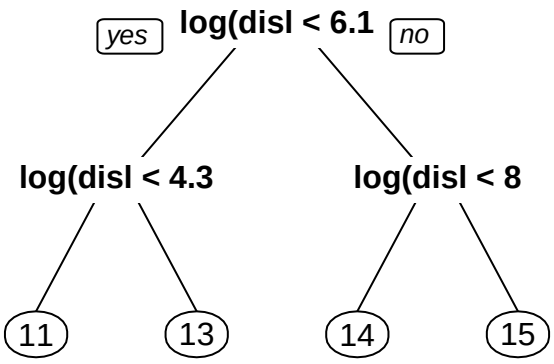
```
## (Intercept)
##    928263.8
```

Problem 5 : CART and Random Forest

In addition to the logistic regression model, we can also train a regression tree. Use the same variable as used in the simple model, logdislikes. Train a regression tree with cp = 0.05.

5.1

Looking at the plot of the tree, **how many different predicted values are there?**



5.2

What is the R2 of this model on the test set?

```
## [1] 0.65701
```

Answer : 0.65701

5.3

The out-of-sample R2 does not appear to be very good under regression trees, compared to a linear regression model. We could potentially improve it via cross validation.

Set seed to 100, run a 10-fold cross-validated cart model, with cp ranging from 0.0001 to 0.005 in increments of 0.0001.

```
## CART
##
## 30002 samples
##     1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 27002, 27002, 27002, 27001, 27002, 27002, ...
```

```
## Resampling results across tuning parameters:
##
##   cp      RMSE       Rsquared   MAE
##   0.0001  0.8105708  0.7622867  0.6107108
##   0.0002  0.8115225  0.7617314  0.6118365
##   0.0003  0.8122356  0.7613073  0.6130578
##   0.0004  0.8148067  0.7598046  0.6149624
##   0.0005  0.8167012  0.7586914  0.6162766
##   0.0006  0.8169936  0.7585151  0.6165615
##   0.0007  0.8174226  0.7582550  0.6170436
##   0.0008  0.8177167  0.7580799  0.6171976
##   0.0009  0.8179288  0.7579512  0.6173344
##   0.0010  0.8187215  0.7574786  0.6180985
##   0.0011  0.8192845  0.7571465  0.6186947
##   0.0012  0.8195414  0.7569911  0.6188644
##   0.0013  0.8195414  0.7569911  0.6188644
##   0.0014  0.8195414  0.7569911  0.6188644
##   0.0015  0.8200332  0.7567003  0.6193313
##   0.0016  0.8203186  0.7565251  0.6196412
##   0.0017  0.8207593  0.7562591  0.6199965
##   0.0018  0.8235695  0.7545884  0.6223759
##   0.0019  0.8257693  0.7532876  0.6242662
##   0.0020  0.8307960  0.7503067  0.6289760
##   0.0021  0.8329095  0.7490437  0.6310261
##   0.0022  0.8336555  0.7485815  0.6320607
##   0.0023  0.8343689  0.7481584  0.6327231
##   0.0024  0.8352155  0.7476368  0.6335798
##   0.0025  0.8360002  0.7471631  0.6345142
##   0.0026  0.8367342  0.7467241  0.6350261
##   0.0027  0.8382352  0.7458069  0.6365821
##   0.0028  0.8388897  0.7453979  0.6371083
##   0.0029  0.8395915  0.7449764  0.6377920
##   0.0030  0.8395915  0.7449764  0.6377920
##   0.0031  0.8395915  0.7449764  0.6377920
##   0.0032  0.8395915  0.7449764  0.6377920
##   0.0033  0.8401534  0.7446404  0.6382681
##   0.0034  0.8401534  0.7446404  0.6382681
##   0.0035  0.8413662  0.7438867  0.6392543
##   0.0036  0.8413662  0.7438867  0.6392543
##   0.0037  0.8421986  0.7433839  0.6398488
##   0.0038  0.8435343  0.7425522  0.6407939
##   0.0039  0.8435343  0.7425522  0.6407939
##   0.0040  0.8440340  0.7422581  0.6414045
##   0.0041  0.8440340  0.7422581  0.6414045
##   0.0042  0.8454258  0.7414175  0.6422986
##   0.0043  0.8458691  0.7411523  0.6425183
##   0.0044  0.8482296  0.7397360  0.6442051
##   0.0045  0.8482296  0.7397360  0.6442051
##   0.0046  0.8491464  0.7391784  0.6447994
##   0.0047  0.8498505  0.7387448  0.6453106
##   0.0048  0.8504174  0.7383988  0.6455231
##   0.0049  0.8512301  0.7379149  0.6460399
##   0.0050  0.8512301  0.7379149  0.6460399
##
```

```
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 1e-04.
```

What is the optimal cp value on this grid?

Answer : 1e-04

5.4

What is the R2 of this new model on the test set?

```
## [1] 0.7528402
```

Answer : 0.7528402

5.5

Create a random forest model that predicts log(views) using the same variable as the CART model, *with nodesize = 200 and ntree = 50. Set the random seed to 100.*

What is the R2 of this new model on the test set?

```
## [1] 0.7513246
```

Answer : 0.7510334

Conclusion

```
## [1] 0.7519569
```

```
## [1] 0.7530081
```

```
## [1] 0.65701
```

```
## [1] 0.7528402
```

```
## [1] 0.7513246
```