Predicting Tik-Tok User Data Based on Video Data

GGteam: Will Chen, Katelyn Cai, Hannah Choi, Weston Slayton

2023-12-01

Introduction and data

With over 1 billion users globally, TikTok is one of the fastest growing social platforms in the world. Understanding ubiquitous algorithm, which is said to generally account for account factors (likes and comments) and video information (captions, sounds, hashtags), is critical to understanding the app's many critiques, from declining youth mental health outcomes and its addictive nature of its explore page. To better understand TikTok's social impact, we decided to explore TikTok's data and how follower count (a huge driver of engagement) is impacted by other aspects of a user's account, like average number of videos, average number of likes, and average number of comments.

The dataset comes from the 'top_users_vids.csv' file (under folder 'Trending Videos Data Collection') of the Github repository found at: https://github.com/ivantran96/TikTok_famous/tree/main. The data was originally collected as part of the DataResolutions's Data Blog project exploring Tiktok's demographics and trending video analytics.

The original data curators collected the data using David Teather's open-source Unofficial Tiktok API (found at https://github.com/davidteather/TikTok-Api), which uses Python to scrape Tiktok data and fetch the most trending videos, specific user information, and much more. Using the list of top Tiktokers, the curators expanded the list of users by collecting suggested users with the API's getSuggestedUsersbyIDCrawler method. They then collected video data of the 25 most recent posts of each user using the byUsername method. They also used the bySound method to collect videos using some of the most famous songs on TikTok to get an idea of how the choice of music can impact the potential of a video to start "trending."

EDA

We begin our EDA process by first examining the dataset.

Currently, our dataset tiktok has 13 columns and 12,559 observations. Each row is a video. The columns cover attributes of each video such as video length, hashtags used, songs/sounds used, and statistics (number of likes, shares, comments, plays, followers, and total number of likes and videos across the account). Variables id, create_time, video_length, n_likes, n_shares, n_comments, n_plays, n_followers, n_total_likes, and n_total_vids are numerical while the others are categorical.

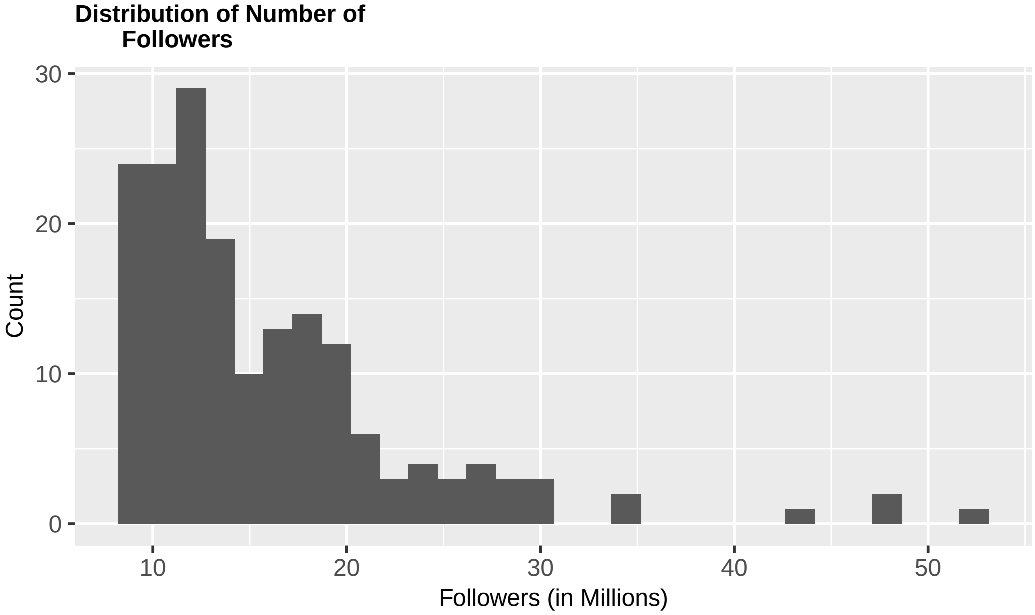
However, it's clear that some of the columns won't be useful for predicting number of followers. It is also apparent that we must address the potential issue user_name might have with the other columns. There's a potential for severe multicollinearity if we choose to just drop user_name, since the number of plays or likes a video would have a strong relationship with the user who posted it. Therefore, any analysis without user and its related features would have to consider the user's account as confounding variables. In addition, we'll be forced to drop valuable features directly related to a user such as user followers, user total likes and user total videos (n_followers, n_total_likes, n_total_vids).

The less relevant variables are create time and video ID. In addition, hashtags and songs might not be useful. Most videos don't include a hashtag and there are too many unique instances of them for it to be valuable in our analysis. We could consider binning hashtag into none and at least 1 hashtag(s), however that wouldn't be useful for our analysis since its rare for tiktok followers to mind the number of hashtags. The same is true for songs; one could consider grouping original songs into one bin and the rest into others. However, from our domain knowledge, its wouldn't be useful to categorize all original songs as similar since most of them could just be user-edited snippets of actual songs.

To address the issues mentioned above, we grouped the data by users and summarized relevant predictor variables by taking their mean. Our modified dataset has 8 columns and 254 observations, with each row being a user.

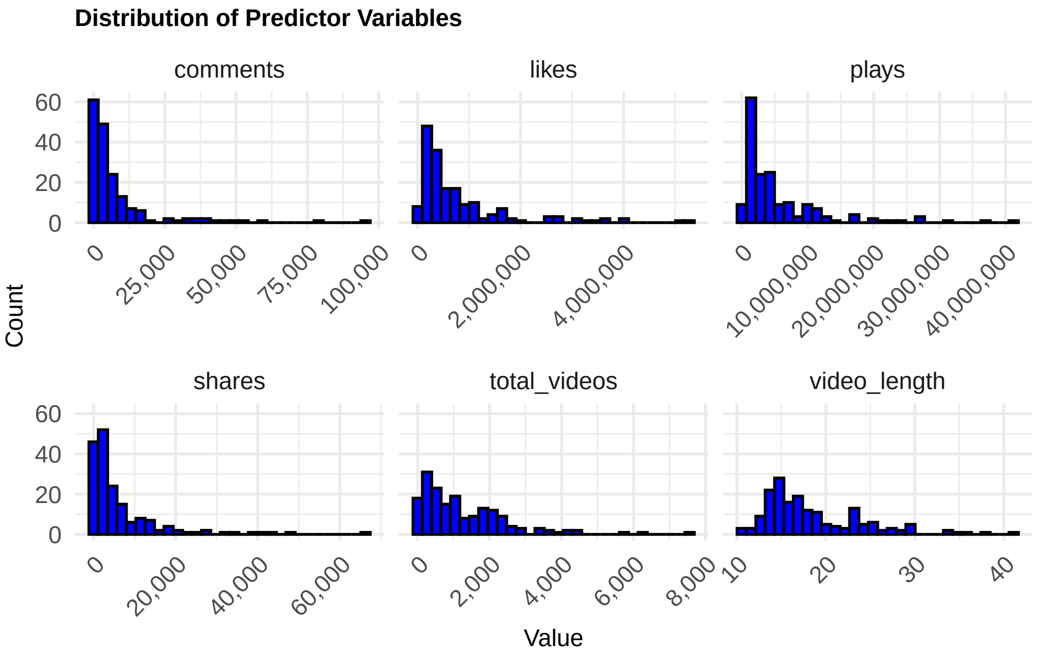
Note that no data leakage is introduced in this process since we are just summarizing by the means of the predictor variable per user. When we split, it'll split based on observations, which are users.

Here's a distribution of our response variable, user followers, from our training set.

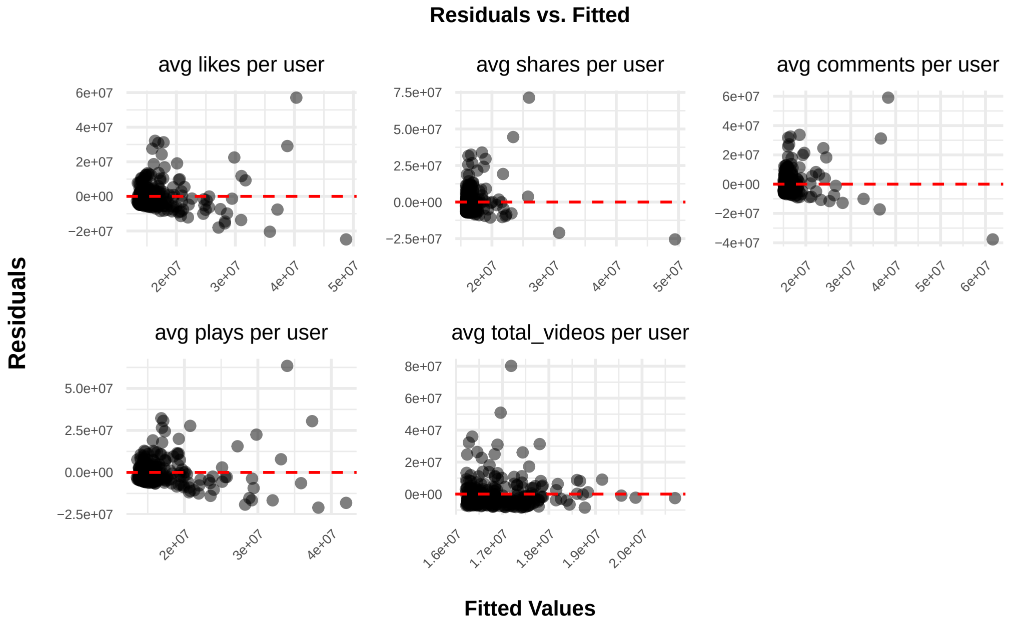


The distribution of our response variable follows, is unimodal and heavily right skewed. The mean is 16,220,526.3 and the standard deviation is 7,710,869.8. The minimum is 8,900,000 and the maximum is 52,300,000. Based on our standard deviation, there seems to be a lot of variation in our dataset; and from our plot, we can see major outliers.

Here are the distributions for the predictor variables we are interested in:

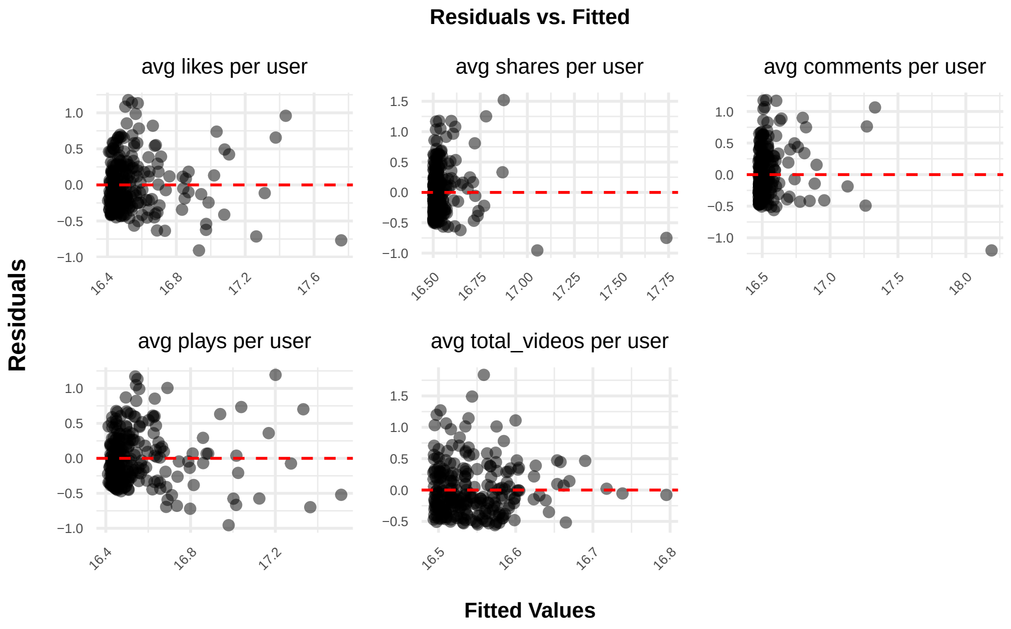


Judging by the number of outliers in our dataset, we are interested in knowing how this might influence our model conditions, specifically constant variance. Constant variance could be violated since we aren't working with a response that's looks normally distributed. Hence, we have the following residual plots for each of the predictor variables we are interested in. The following plots are generated from creating individual linear regression models with just the predictor variable and average follower response.



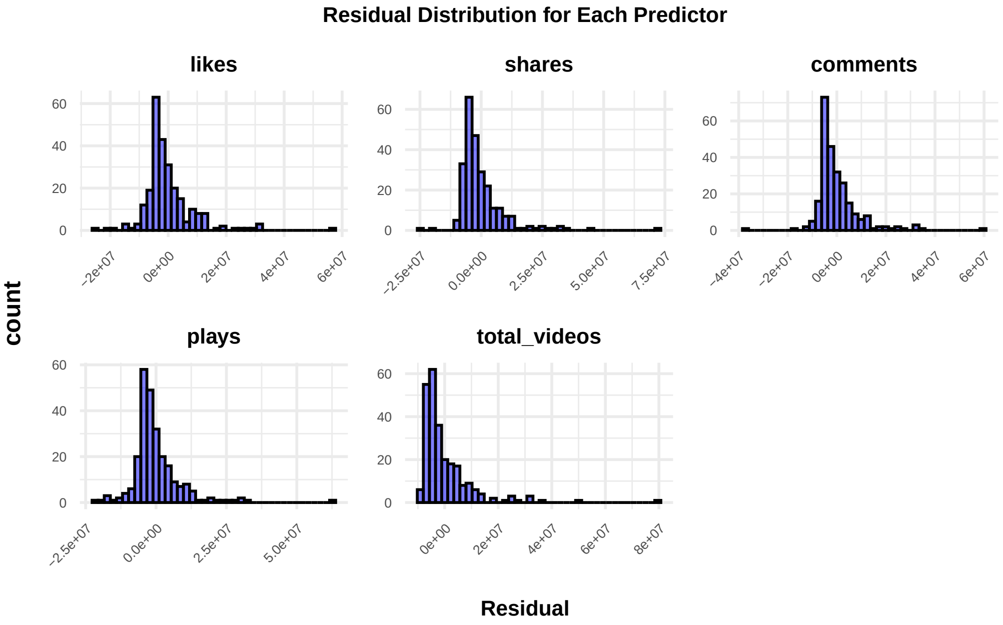
All our predictors variables violate constant variance, because there's a clear outward fan spread for likes, shares, comments, plays and video_length, and an inward spread for total_videos. As a result, we log transformed our response variable of followers.

Because Tiktok videos are commonly divided into 15-second, 1 minute, or 3 minute videos, it also makes sense to process average bin video length into levels, corresponding to "short", "medium" and "long." Now, we can search for interactions effects video_length might have with other predictors such as likes. Therefore, we'll add step_discretize() into the recipe for video_length. We also mean-centered all our numerical variables to make our intercept meaningful and we scaled comments to hundreds, likes to millions, plays to tens of millions, shares to hundreds, and total_videos to units, in order to make these predictors more interpretable.



Fanning is a lot less noticeable now. The residual plots don't seem to suggest any underlying patterns. As such, we conclude the predictors satisfy linearity and constant variance. We can also assume independence is met. Each of the videos are by individual creators, therefore the videos are independent of each other.

We continue with normality.



Normality seems to be satisfied for each of the predictors except total_videos and possibly shares. However, even though total_videos and shares do not have completely normal distributions, because we have more than 30 observations in the dataset, we can conclude that normality is satisfied regardless of the distribution.

We use the transformed dataset to split our data into testing and training. The transformed dataset also includes our log response, log 'followers'. We also mean-centered all our predictor variables to make our intercept meaningful.

Methodology

In honing our final model, we used the transformed dataset to reduce multicollinearity and conduct a cross-validation test on tiktok_train.

Detecting Multicollinearity & Model Comparison

Upon conducting a VIF test, we found that likes and plays had the highest vif values (11.614 and 9.82 respectively).

```
    likes       shares     comments        plays total_videos
11.595466     3.507545     2.598273     9.798934     1.168067
```

Therefore, we constructed two linear regression model fitting variable ‘followers’ with predictor variables ‘likes’, ‘shares’, ‘comments’, ‘plays’, ‘video_length_bin’, and ‘total_videos.’ Our first model m1 had all the previously indicated predictor variables excluding variable ‘likes’ while our second model m2 had those predictor variables excluding variable ‘plays.’ We compared these two models because meaning they have the highest likelihood for multicollinearity.

term	estimate	std.error	statistic	p.value
(Intercept)	16.492	0.043	385.562	0.000
shares	-0.012	0.005	-2.582	0.011
comments	0.008	0.003	2.982	0.003
plays	0.032	0.007	4.885	0.000
video_lengthbin2	0.002	0.061	0.027	0.979
video_lengthbin3	0.059	0.061	0.973	0.332
total_videos	0.092	0.022	4.243	0.000

term	estimate	std.error	statistic	p.value
(Intercept)	16.496	0.043	384.488	0.000
likes	0.247	0.052	4.753	0.000
shares	-0.010	0.005	-2.247	0.026
comments	0.004	0.003	1.533	0.127
video_lengthbin2	-0.003	0.061	-0.043	0.966
video_lengthbin3	0.050	0.061	0.831	0.407
total_videos	0.090	0.022	4.142	0.000

Model Comparison with 5-fold CV

To determine which between m1 and m2, we performed 5-fold cross validation and extract the resulting BIC and AIC scores along with additional evaluations.

Model 1: (without likes):

RMSE:

```
# A tibble: 1 x 4
  mean_rmse mean_adj_rsq mean_aic mean_bic
      <dbl>        <dbl>    <dbl>    <dbl>
1     0.340        0.266     90.7     114.
```

Model 2 (without plays):

RMSE:

```
# A tibble: 1 x 4
  mean_rmse mean_adj_rsq mean_aic mean_bic
      <dbl>        <dbl>    <dbl>    <dbl>
1     0.333        0.259     92.1     116.
```

The difference between the model's evaluations aren't large. Model 1 has a slightly higher RMSE, but it has a lower AIC and BIC, and a higher adjusted r-squared. In this case, we would consider model 1 (the model without likes) to be a better model since it's able to explain more of the variance while also maintaining lower AIC and BIC scores. Therefore, we choose to remove likes and leave plays in the model .

Determining whether video_length_bin are necessary

We saw from our tidy table that the p-values associated with video length bins are high, indicating that the variables aren't significant. Because of this, we can do once again do cross validation to test to see how a model without video_length and likes performs against m1 (our chosen model without likes but contains video length).

We can see from our previous tidy table that the p-values associated with the video length bins are high, indicating that the variables aren't significant. Because of this, we can do a cross validation test to see how a model without video_length performs.

Model 3 (without video length and likes):

```
# A tibble: 1 x 4
  mean_rmse mean_adj_rsq mean_aic mean_bic
      <dbl>        <dbl>    <dbl>    <dbl>
1     0.339        0.269     88.2     106.
```

We can see that when we remove video_length, it makes sense that AIC and BIC both decrease. We also see that adjusted r-squared only slightly increased by approximately 0.002. Therefore, we prefer the parsimonious model without video length bins.

Determining whether interaction terms are needed

We used domain knowledge to hypothesize several potential interaction variables. Our first idea was connecting average plays and likes (or other engagement metrics) to detect bot activities. Our second idea was connecting likes and total number of videos to evaluate the relationship between user engagement and activity. Our third idea was connecting average likes and shares to evaluate the viewer's reason for shares (to detect potential ridiculing sentiment assuming that they would not like the video they feel negatively towards). All interaction terms had low p-values, but the first and third interaction terms showed signs of multicollinearity by increasing the p-value of other variables in the model.

This is a table with the second interaction term:

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	16.562	0.029	570.025	0.000	16.505	16.620
likes	0.200	0.088	2.280	0.024	0.027	0.373
plays	0.022	0.011	2.033	0.044	0.001	0.044
shares	-0.010	0.005	-2.218	0.028	-0.020	-0.001
comments	0.005	0.003	1.707	0.090	-0.001	0.010
total_videos	0.139	0.026	5.318	0.000	0.088	0.191
likes:total_videos	0.116	0.039	2.996	0.003	0.040	0.193

We can see from the p-values in the table that all variables seem to be statistically significant with the interaction term. Therefore, those terms should certainly be in our model. The new p-value for comments is not less than our significance level of 0.05, but it is still low. We can perform CV on a model with likes:total_videos.

Model 4 (with likes:total_videos interaction terms):

```
# A tibble: 1 x 4
  mean_rmse mean_adj_rsq mean_aic mean_bic
      <dbl>        <dbl>    <dbl>    <dbl>
1     0.328        0.316     81.8     108.
```

The mean RMSE (0.3282339) is barely smaller than that of model3, one without likes and video_length_bin (0.3389576). However given we are have included more terms, our adj r-square shows a clear increase (0.26883 to 0.3157319), BIC shows a slight increase (105.9608 to 108.3907), and AIC shows a slight decrease (88.243 to 81.8137). Because the increase in our adj r-square is very significant, AIC decreases, and BIC only increases by a small amount, we choose the new model, model 4.

Results

Model 4 performance on test:

```
# A tibble: 2 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rmse    standard       0.398
2 rsq     standard       0.385
```

Note that we log transformed our response variable. In order to evaluate the meaning of our RMSE of 0.4067, we take exp(0.4067) ~ 1.502. This value is the multiplicative square difference. For example, if have log followers of 16.04552, our model will be more or less off by $16.04552 \pm .4067^2 \implies \exp(16.04552 \pm 0.1654) \implies 7882219 < 9,299,954 < 10,972,689$. This means our model does a fairly poor at predicting a tiktok user's followers. We also have an RSQ of 0.3615, indicating only 36.2% of the variability in followers can be explained by our predictor variables.

There are several terms that are significant when determining the number of followers a tik tok user has. The number of total videos, comments, and plays seems to have a clear positive relationship with follower count. This also would align with our expectations, as the more videos you make, the more engagement your profile is likely to have and more followers you may gain. However, shares have a negative relationship with follower count, which initially seemed counter-intuitive. While it is impossible for the model to determine causality or explain why exactly a relationship exists, we hypothesize that users may share a video because they dislike it, resulting in them not following the user.

When observing the video length bin variable, the middle video length bin (2) has statistically significant difference from the other two video length bins, as well as a statistically significant interaction term with total videos. This shows that not only do medium length videos generate the most followers, but medium length videos combined with a higher number of total videos significantly increase follower count as well. This is certainly an interesting finding from our analysis, as it isn't the most expected result.

Conclusion

We originally decided to look at TikTok's data and how follower count (a huge driver of engagement) is impacted by other aspects of a user's account. We learned that it is extremely difficult to correctly predict follower count, given our model only captures 36.2% of the variability in the dataset. More complex models seemed to only worsen performance, and we chose to prioritize parsimony for this reason - however, even the simple models did not predict well.

Our dataset was extremely difficult to work with, given that it did not meet the conditions for linear regression (linearity and constant variance), and contained multicollinearity. The variables were also extremely large, and needed to be scaled down to have meaningful coefficients - which made late interpretation significantly more difficult. A more complex model was likely needed, that was beyond the scope of our knowledge, given how poorly our model performed at the end. There also may be underlying relationships between follower count, and other portions of the TikTok algorithm that are not contained in the dataset, which our model might have also failed to capture; in the real world, users have reported that TikTok enforces policies differently from user-to-user, and uses different algorithms from region to region.

In order to improve our analysis, it would be helpful to comb TikTok for a dataset that potentially contains more variables. Three potential options we considered included: finding a meaningful way to capture hashtags (which may require manually looking at TikTok videos), finding a meaningful way to capture whether a user typically utilizes trending music, and using the demographic statistics for users (to account for human decisionmaking).

Appendix

First 5 data points before transformation

```
            id create_time     user_name hashtags                         song
1 6.892505e+18  1604786417 charlidamelio       [] Adderall (Corvette Corvette)
2 6.892162e+18  1604706644 charlidamelio       []               original sound
3 6.892157e+18  1604705486 charlidamelio       []               original sound
4 6.891688e+18  1604596107 charlidamelio       []               original sound
5 6.891016e+18  1604439653 charlidamelio       []               original sound
6 6.890973e+18  1604429723 charlidamelio       []      Lemonade Internet Money
  video_length n_likes n_shares n_comments  n_plays n_followers n_total_likes
1           15  480800     9256      51300  1900000    97400000       7.6e+09
2            9 3100000    17200     105700 13300000    97400000       7.6e+09
3            4 2400000    17800      69200 10100000    97400000       7.6e+09
4           15 3200000    12700      64100 14600000    97400000       7.6e+09
5           13 7500000    31100     290300 34700000    97400000       7.6e+09
6            7 7100000    43000      82000 33300000    97400000       7.6e+09
  n_total_vids
1         1642
2         1642
3         1642
4         1642
5         1642
6         1642
```

After transformation

```
# A tibble: 5 x 8
  user_name     likes shares comments  plays followers video_length total_videos
  <chr>         <dbl>  <dbl>    <dbl>  <dbl>     <dbl>        <dbl>        <dbl>
1 .kunno       335020  1067.    4521. 1.99e6  15300000         19.5         3442
2 _arishfakha~ 425868  5269.    2700. 3.96e6  28600000         14.5         2026
3 _saloniyaapa 166544  6122.     728. 1.85e6  12900000         14.6         2005
4 aashikabhat~ 194280  1335.    1053. 2.17e6  16000000         15.0         2720
5 abbyrartist~ 965586  5292.    7005. 3.97e6  13500000         16.5          811
```

Split results:

<a name=“appendix”></a>

```
<Training/Testing/Total>
<177/77/254>
```

> ❗ Important
>
> Before you submit, make sure your code chunks are turned off with `echo: false` and there are no warnings or messages with `warning: false` and `message: false` in the YAML.