Stats101C Final Project

Team MPGA: Crystal Li, Chongxuan Bi, Wenru Shi

I. Introduction

YouTube is a large scale video-sharing platform owned by Google since late 2006. It has become the biggest video platform for people all around the world. While billions of people publish and watch videos on this platform, views for individual videos gradually come on the spotlight, as both a measurement of video popularity and also a source of income for popular youtubers.

In this project, we aim to predict the percentage change in views on a video between 2 to 6 hours since its publishing based on provided variables. Our research goal is to find what variables affect the growth rate between 2 to 6 hours after publishing most.

II. Methodology

A. Preprocessing

We employed 4 methods in the preprocessing step: combine columns into factors, generate new variables, and remove no variation variables and high-correlated variables.

1) : Gather binary columns into factors

We gathered 12 binary columns in to 3 factor columns because they are related. Columns avg_growth_low, avg_growth_low_mid, avg_growth_mid_high are combined into a factor column named avg_growth with 3 levels. The same transformation also applies to columns Num_Views_base, Num_Subscribers_Base, and count_vids. By doing this, we effectively combined many correlated columns into fewer independent ones, which made later analysis valid and clear.

2) : Split the PublishedDate column into factors

We split the PublishedDate column into 3 factor columns. We surmised that date and time would be important factors influencing the growth rate. As each observation in PublishedDate column differed from one another, the original data was not informative and had too much noise. Thus, we split into followings. The date part was used to produce factor columns month and dow (day of week), and the time part was used to generate column qod (quarter of day). Therefore we added a new column indicating different months, because we believed the pandemic had a heavy impact as people spent more time home watching videos. Week of day was also important because on weekends people had more spare time for watching videos than on weekdays. Different time of the day was also crucial, since people tend to watch less before dawn and more in the evening after work. Therefore we made use of published time by splitting it into 6 time intervals.

3) : Remove unrelated/no variation variables

We deleted unrelated variables and variables with 0 standard deviation. We removed variables of max/min of colors because as long as there is a white/black spot on the thumbnail image, the max values for all color channels will be 255/0. So these variables were not very informative. Columns of identical values were also not insightful because they could not differentiate observations, as shown in Fig.1.

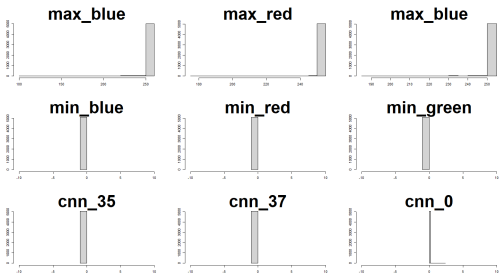

Fig. 1. Histogram of Various Predictors

4) : Combine columns of various punctuations

We combined all punctuation counts into a single column because we thought the number of punctuation had limited effect and the range of values of original columns were too small to be noticed.

5) : Remove Highly-correlated variables

As some variables are highly correlated, as shown in Fig.2, We removed highly correlated variables to address the issue of multicollinearity. The cutoff is correlation of 0.8.

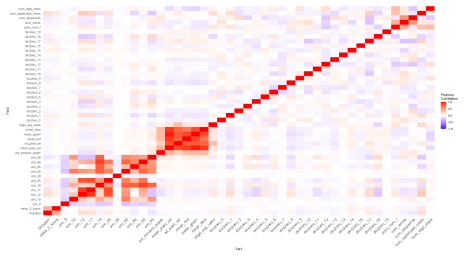

Fig. 2. Correlation between Predictors

As both 'hog' columns and 'cnn' columns give high-level features of the thumbnail image, we believed they were correlated. We observed prediction performance after we deleted columns of either type and decided to remove 'hog' variables after comparison.

After preprocessing variables through different dimensions, 48 most significant variables were chosen in the end to apply into our model selections.

B. Statistical Model

We used 70% of provided training data as our own training, while the rest 30% as our validation data to test for validation dataset prediction errors in RMSE.

1) Model Trial 1: Backward Selection Model

We first applied Backward Selection Model on all 48 variables, as it starts with all predictors in the full model, iteratively removing the least contributing predictors, and stops when removing any extra predictors increases RMSE. In this model, the minimum RMSE occurred when there are 37 variables, with RMSE = 1.672446.

2) Model Trial 2: Regularization

The second trial we used are regularization methods – Ridge Regression and LASSO.These two regularization methods are forms of regression, which shrinks the coefficient estimates towards zero, avoiding the risk of overfitting.

By applying Ridge and LASSO in R, the minimum RMSE we get from LASSO is 1.82, Ridge Regression as 2.17.

3) Model Trial 3: Principal Component Analysis

Based on principal component analysis (PCA), we used Principal Component Regression (PCR) for estimating the unknown regression coefficients in this regression model. We chose to try for PCR as this method can overcome the multicollinearity problem when two or more of the explanatory variables may be collinear.The minimum RMSE we got for PCR is 1.848.

4) Model Trial 4: Partial Least Squares

We then further applied Partial Least Squares (PLS), which also overcomes the multicollinearity problem. Compared to PCR, predicted variables and observable variables in the model are projected to a new space. The minimum RMSE we got for PLS is 1.680653.

5) Model Trial 5: Bagging

After trying out previous models, we haven't seen a huge improvement in the reduction of RMSE. Thus, we progressed to use decision trees methods.

We first used bagging to construct deep trees with low bias, then combat variance. Bagging can reduce variance and avoid overfitting, resulting in lower prediction errors. The minimum RSME for bagging is 1.464276, which is a significant improvement compared to previous models.

6) Model Trial 6: Random Forest

We then tried random forest, a decision tree method like bagging but only differs in going through all possible splits on a random sample of a small number of variables m, where m p. Random forest can also avoid overfitting to the training data and can reduce variability further.

In choosing the value of mtry (variables considered at each split), we used the normal approach when using random forest in regression issues – to divide the number of variable by 3, with variables considered at each split equals to 16.

The minimum RMSE for random forest is 1.492614, also a huge improvement compared to other non-decision tree methods.

7) Model Trial 7: Boosting

The last decision tree method we used is gradient boosting method. It constructs small trees with high variance, and then it combats bias.The most important feature of gradient boosting is its ability to improve weaker learners to make better predictions, which further reducing the prediction error. The minimum RMSE for Gradient Boosting is 1.497578, similar to random forest method.

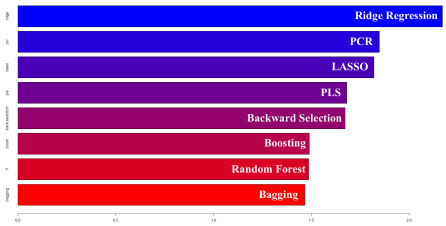

Fig. 3. Validation Dataset RMSEs of Different Models

After applying 7 different models, as Fig.3 shows, bagging has the smallest RMSE on the validation dataset, followed by random forest, boosting, backward selection, PLS, LASSO, PCR, and ridge. Thus, based on these trials, we chose random forest as our final model as it has the second smallest validation RMSE and reduces variability further compared to bagging method.

III. Results

A. Tuning of the Model

After selecting the random forest as the final model based on RMSE, we start to further tune the model by trying different mtry and ntree choices. As shown in the left part of Fig. 4, we have the lowest RMSE at mtry =27 when ntree = 600. Then, we test RMSE for different ntree selections at mtry=27 in the next step. As shown in the right part of Fig.4, ntree 600 and 750 have the lowest RMSE, and we chose ntree= 600, the comparatively smaller ntree choice, to avoid the potential over fitting and long running-time.

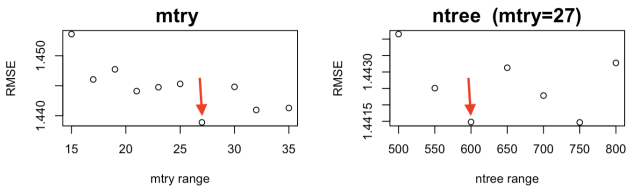

Fig. 4. mtry tuning(left) and ntree tuning(right)

Similarly, we also tried mtry = 33, 35 with different ntree choices and found the lowest RMSE in each combination. Through comparison in 3 mtry-ntree combinations, we found out the lowest RMSE happened when mtry =27 and ntree=600 as shown in Fig 5. Thus, we get 1.4415 RMSE with 48 variables, 27 mtry and 600 ntree in our own splitted test dataset. Moreover, we fit the model on the whole training dataset to predict the growth_2_6 in the test dataset. Finally, we got 1.392 in the public leader board.

We used random forest's built-in 'importance' function to verify our selected variables. As Fig.6 shows, when 'IncMSE'(increase of the Mean Squared Error when

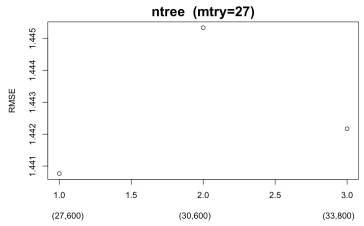

Fig. 5. Combinations for mtry equals to 27, 33, and 35

given variable is randomly permuted) becomes higher, the more important the variable is. Our previously selected variables were all verified as important in our current random forest. As shown in Fig. 7, 10 most important variables (largest percent increase in MSE without split of given predictor) include avg_growth, cnn_25, cnn_10, Num_Subscribers_Base, cnn_86, cnn_89, Num_Views_Base, view_2_hours, count_vids, and cnn_88. Also, some other important variables include duration, qod (published time in one day) and num_uppercase_chars.

	%IncMSE	IncNodePurity
Duration	27.6185504	837.3865
views_2_hours	39.9546184	1698.4802
cnn_10	73.9925751	2845.9662
cnn_25	89.2358434	3121.2800
cnn_68	30.8572490	683.1944
cnn_86	63.3065799	1470.0432
cnn_88	28.9080001	789.4754
cnn_89	55.7365028	2464.4970
pct_nonzero_pixels	5.8290584	401.3064
sd_pixel_val	15.0134993	495.9164
num_uppercase_chars	23.4401140	407.4422
avg_growth	157.4739117	15515.7997
Num_Views_Base	42.8000833	5186.7967
Num_Subscribers_Base	66.3845082	2198.8829

Fig. 6. Random Forest Variables Importance

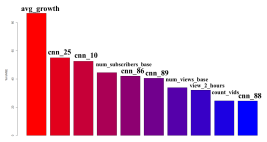

Fig. 7. 10 Most Important Predictors

B. Findings

From the result we can find that the growth rate from 2 to 6 hours of one YouTube video largely depends on its channel. If the channel has a high average growth for all its videos, has more subscribers, has more total views of all its videos in the channel, the video will have a high growth rate between 2 to 6 hours after publish.

In addition, the thumbnail image also plays an important role in increasing the growth rate in videos' views between the 2nd and 6th hour of its publishing. Even though we do not know exact images or patterns in high-click rate videos, the result shows that some measurements show statistical significance in improving the growth rate between 2 to 6 hours after videos are published.

Finally, duration of the video, what time the video posts, and the number of uppercase letters in video title have an impact on growth rate between 2 to 6 hours after videos are published. However, more research needs to be conducted on these factors to examine the correlation between these factors and the growth rate of the video.

IV. Conclusion

A. Summary

Attracted by determining factors in video views growth from 2 to 6 hours after publish date, we prepossessed variables based on correlation, variation and creativity on merging and splitting variables. Reducing the number of variables to 48, we employed 7 statistical models with detailed tuning on hyper parameters and compared RMSE to decide the final model with best performance. Based on selected random forest as the final model, we also figure out the ranked importance of current variables. While evaluating advantages of random forest, we acknowledge certain drawbacks hidden in our model.

B. Evaluations

1) Drawbacks: Since we chose random forest model at the end, we face the loss of model interpretability. Meanwhile, random forest model will take up a lot of memory for very large data sets, especially in our case a 7242*48 data set. Because of its high computational complexity, the running time is relatively high. Furthermore, random forest takes us a long training time to tune those hyper parameters to determine the final mtry and ntree.

2) Advantages: For the overall model, we have tried a lot of models based on the current selected variables. Comparing RMSE of all the models, we select the random forest with the smallest RMSE in mtry =27 and ntree=600. That is because the inborn advantages of random forest: It creates many trees on the subset of the data and combines the output of all the trees. Thus, it reduces the overfitting problem in decision trees and the variance, therefore improving the accuracy. Meanwhile, there is no feature scaling required, such as standardization and normalization, since Random forest rule based on approach instead of distance calculation. Moreover, Random forest has the power of handle large data sets with higher dimensionality 7242*48. It can handle thousands of input variables and identity most significant variables so it is considered as one of the dimensionality reduction method. Further, the model outputs importance of variable, which is always essential for reporting and presentation.

3) Next Step: First, go into detail of how factors such as duration of videos, what time videos post, and number of uppercase letters in video title will impact on growth rate.

Additionally, there might exist some interactions between variables such as doc2vec and publish date and high-level image features (hog and cnn) and doc2vec etc.

After testing the potential interactions, we should dive further into hog and cnn part. Since both of them represents image features from two different extraction methods. There could be some overlapping between them, which could be more complex than solely deleting all the hog features.

C. Retrospection

After this competition ends this Friday, the scores on our public and private leader board are respectively 1.39232 and 1.39941, differed by 0.007. Since the difference is negligible to be considered as over fitting, we consider the difference mainly come from the randomness generated from the 40 percent extraction policy in the Kaggle evaluation.

V. Statement of Contribution

A. Kaggle

Wenru Shi worked on selecting and testing for various models.

Chongxuan Bi worked on preprocessing part of model fitting.

Mingze Li worked on variables selection and the model tuning.

B. Report

Wenru Shi worked on B.Statistical Model in II.Methodology, and B.Findings in III.Results.

Chongxuan Bi worked on A.Preprocessing in II.Methodology.

Mingze Li worked on IV.Conclusion and A.Tuning of model in III.Results.

C. Presentation

Wenru Shi worked on Findings & Suggestion part & next steps.

Chongxuan Bi worked on Introduction and Preprocessing.

Mingze Li worked on final evaluations & model evaluations.