Final Project - The Gould-en Rule

Stats 101C Lecture 3

Andy Shen, Ethan Allavarpu

Fall 2020

Introduction

The purpose of this regression analysis was to predict the growth percentage of a newly uploaded YouTube video during the 2nd through 6th hour of its publication (`growth_2_6`), which may potentially indicate the virality of a YouTube video (i.e., how quickly a video can go viral and "break the Internet." In this analysis, we employ a variety of regression techniques to a variety of attributes that make up a YouTube video.

Methodology

Preprocessing

Data Cleaning We cleaned the data by plotting each predictor variable against `growth_2_6` and examined each univariate plot for possible associations between predictor variables and the growth percentage. We noticed that there existed many outliers or stray points that did not belong in the plot. We systematically removed these points, basing it off personal judgment as to whether the stray points would not assist in prediction. However, we do remove highly correlated variables as indicated by a correlation matrix heat map. For predictor pairs with a correlation coefficient over 0.9, we removed one of the two predictors to avoid issues with correlation.

We also transformed multiple variables and added another variable prior to performing predictor selection, as we saw these as potentially useful for data on YouTube growth. The first adjustmeent we performed was on the `PublishedDate` variable. To this variable, we performed a transformation and added another variable. Initially, the dataset had `PublishedDate` as a character variable; we wrote a function to transform this variable into a numeric variable in terms of minutes (since January 1^{st}, 2020 00:00). This would allow us to see if there was any relationship between the date and `growth_2_6` because we made the variable continuous to match the general perception of time. Not only did we do this with respect to general time since January 1^{st}, 2020 00:00, but we also *added* a variable (`TimeDay`) that has the time of day (in minutes from 00:00) during which a video was uploaded to YouTube. This could be useful because people may be more likely to watch YouTube videos in the evening (after school or work), so posting nearer to those times allows for greater growth between the second and sixth hours of a video's upload.

The second transformation we made was regarding the binary variables near the end of the dataset (Num_Subscribers_Base_low, Num_Subscribers_Base_low_mid, Num_Subscribers_Base_mid_high, Num_Views_Base_low, Num_Views_Base_low_mid, Num_Views_Base_mid_high, avg_growth_low, avg_growth_low_mid, avg_growth_mid_high, count_vids_low, count_vids_low_mid, count_vids_mid_high). When looking at the feature descriptions, wee noticed that the `low`, `low_mid`, and `mid_high` versions of the four variables were natural ordinal variables (with `high` corresponds to values of 0 for the other three options). For these variables, we combined them into a single factor variable with four levels (0, 1, 2, 3) which correspond with `low`, `low_mid`, `mid_high`, and `high`. We did this transformation because bagging and random forest can split along multiple levels of a factor at a node (i.e. 0 and 1 to the left, 2 and 3 to the right), but if we left it as three separate binary variables, the bagging and random forest models could only split between one of the four levels and the three others. Thus, by combining these variables into a factor with four levels, we allow the random forest and bagging models greater ability to distinguish between

the four levels (`low`, `low_mid`, `mid_high`, `high`) of each of these statistics (`Num_Subscribers`, `Num_Views`, `avg_growth`, `count_views`).

We also remove "zeroes" from the data set - ETHAN WRITE ABOUT THIS MORE AS I DONT REALLY KNOW HOW TO EXPLAIN THIS IF IT'S EVEN NECESSARY. TECHNICALLY HAVEN'T DONE IT YET (NOT IN MODEL 15)

Predictor Selection In order to refine our subset of predictors, we use the LASSO to select significant predictors. We first fit a LASSO model for a sequence of candidate λ values. From there, we select our optimal value of λ as the one that is one standard deviation above the λ value that resulted in the lowest test MSE. From these parameters, we extract the predicted coefficients in this LASSO model as our predictors for the candidate model.

We use LASSO to refine our predictors because this technique shrinks coefficient estimates to 0 and keeps the most important ones. As such, we are only interested in the predictors with nonzero coefficients. LASSO is unrelated to the our candidate model of random forest, and it was only used as a technique to refine the large number of predictors in the data set.

Statistical Model

A vast majority of our models were fit using either bagging or random forest. Our first model used LASSO to select predictors, but we did very little pre-processing and did not remove outliers. This did not result in a very successful model which is why we decided that pre-processing was necessary. We then implemented a mixture of bagging and random forest, adjusting certain parameters at a time, such as the number of trees, their depth, as well as our λ values.

The reason we chose bagging and random forest models over methods such as LASSO for final predictions and boosting are because bagging and random forest models produced lower RMSE values. INSERT FIGURE/HIST COMPARING RMSE OF BAGGING, RF, LASSO AVG, and BOOSTING

Our best candidate model, `Model 15`, utilized the random forest approach. After the preprocessing to select the variables that seemed important, we used our smaller dataset to determine the best value of m. This m corresponds to the number of variables the model randomly considers in each node of each decision tree produced. To pick the optimal value of m for our final model that would predict the `growth_2_6` values for the test data, we utilized 5-fold cross-validation. We created our own cross-validation function to get a better estimate of the test RMSE and tried several potential values for m, ranging from 1 to p, to determine which m produced the lowest RMSE. In the case of `Model 15`, we chose the best m based on the minimum *median* RMSE (of the five folds). The reason we chose median and not mean is because with only five folds, a single extremely high or extremely low RMSE could significantly impact the average RMSE for a given m; the median is more resistant to extreme values. This resistance provides reason for us to choose the median RMSE rather than mean RMSE in cross-validation to determine the best m.

Once determining the best m, we fit another random forest–this time with more trees (500 as opposed to 50) on 80% of the training data (20% was set aside as final validation data). We did this to ensure there weren't any final "surprises" when fitting a model with regards to RMSE. After fitting the random forest for the given m and checking the validation RMSE, we predicted values of `growth_2_6` for the test dataset to submit to Kaggle.

Results

Our best evaluation metric with our primary model (`Model 15`) on the Kaggle public leaderboard was an RMSE of 1.39594. Our secondary model (`Model 15b`) had a Kaggle public leaderboard RMSE of 1.40285.

Conclusions

We believe that our model performed well because it works as an ensemble method, meaning that it combines multiple individual models to get more accurate responses. By using cross-validation for our selection of

our best value of m, we limited the potential effect of a random seed showing us an inaccurately good or inaccurately bad RMSE.

Code Appendix