

Final Presentation - The Gould-en Rule

Stats 101C Lecture 3

Andy Shen, Ethan Allavarpu

Fall 2020

Section 1

Introduction

Introduction

- With the rise in popularity of YouTube, many people are now making a living off creating YouTube videos
- The more views gained by the video, the more likely it is for that channel to profit
- We are interested in predicting the growth rate in video views between the **second** and **sixth** hour that a YouTube video is published

Section 2

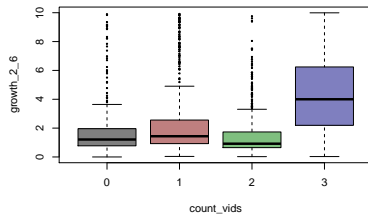
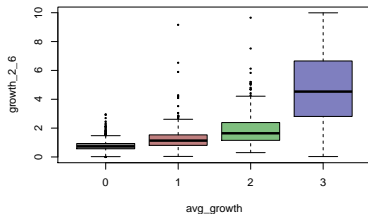
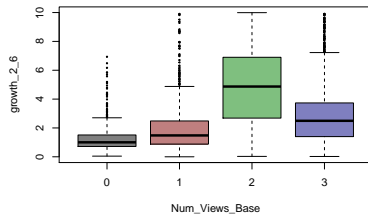
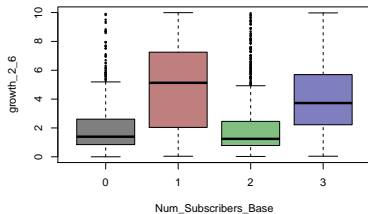
Methodology

Subsection 1

Preprocessing

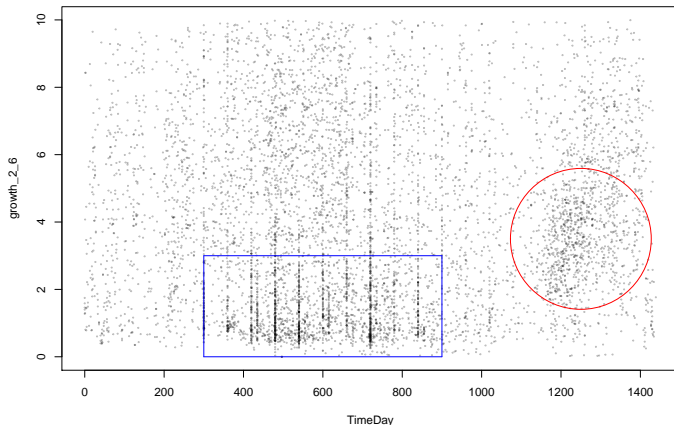
Feature Transformation

- Combined binary variables into a single factor with four levels



Feature Expansion

- TimeDay: What time in the day was a video published?
 - ▶ Created continuous variable



- Small upward trend toward cluster around 1200 (lower clusters earlier in the day)

Outliers

- Examined a univariate plot to look for stray points and removed them systematically
 - ▶ Based off personal judgment and inference on the effect of the stray points
- Remove variables with standard deviation of 0
- We also remove highly correlated variables as indicated by a heat map
 - ▶ To avoid overfitting based on having too many predictors

Predictor Selection

- We use LASSO to select significant predictors
 - ▶ LASSO pushes the coefficients of non-significant predictors to zero
 - ▶ Keeps the most significant ones
- Fit a LASSO model and select our optimal value of λ as the one that resulted in the **lowest cross-validation MSE** (10^{-2})
- Extract the predictors with nonzero coefficients in the LASSO model as our predictors for the candidate model

Subsection 2

Statistical Model

- Most of our models were fit using bagging or random forest
 - ▶ Only adjusted certain parameters at a time (number of trees and m)
 - ▶ Performed better than LASSO, boosting
- Preliminary least-squares model
 - ▶ Kaggle score of ~ 1.65

Candidate Model: Random Forest

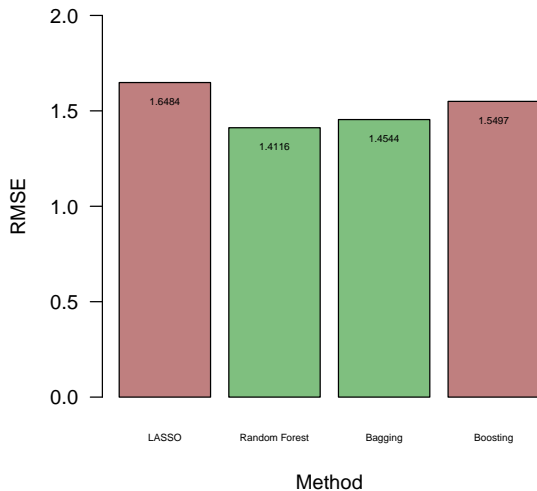
- Use subset to choose m for random forest
 - ▶ Find optimal m with 5-fold cross-validation: select m corresponding to the lowest *median* RMSE of the 5 folds
 - ★ Median is more preferable than mean due to the mean's sensitivity to extreme points
- Once optimal m is selected, fit random forest model to 80% of the processed training data
 - ▶ Extra model and validation RMSE ensures consistent performance

Candidate Model: Bagging

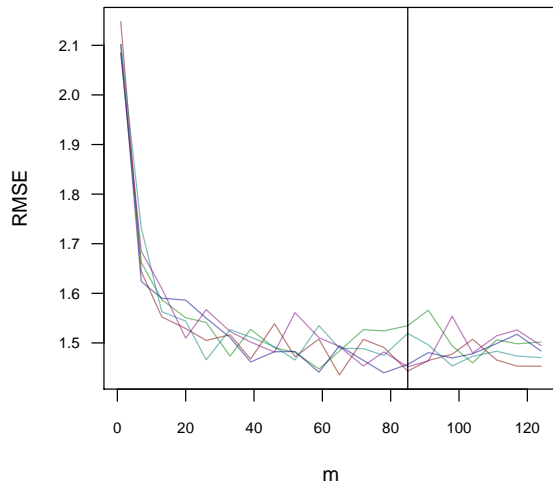
- Considered a bagging approach ($m = p$) as a secondary model to random forest (as opposed to m equaling the value with the lowest median RMSE)
 - ▶ p : Number of predictors

RMSEs

RMSEs for Various Approches



Best m for Random Forest



Section 3

Results

Table 1: Kaggle Scores

Model	Public	Private
Primary	1.39753	1.41019
Secondary	1.40285	1.41321

- Both models beat all four thresholds on the public and private leaderboards

Section 4

Conclusion

Conclusion

- We believed our model performed well due to the fact that it works as an ensemble method
 - ▶ Combines multiple individual models to get more accurate responses (averages predictions)
- By using cross-validation for our selection of m , we limit the potential effect of a random seed showing us an inaccurately good or bad RMSE
- Feature transformation and expansion useful

Variable Importance

