STAT 716 Final Paper

Group 5

Introduction

The purpose of this paper and project was to figure out how the parameters given in the data set can predict the number of shares of online news articles. The dataset was used in the 2015, Fernandes *et al*., paper where the group uses this data to try and offer optimization advice on how to improve shares of online news articles on the online site Mashable. This dataset was scraped from the news site from January 7, 2013 to January 7, 2015. Special articles that did not follow the usual html template were omitted from the data. The Fernandes et al. created and tested models using classification techniques but in doing so, they lose the ability to allow the end user, the news site, to decide on how many shares is considered popular. In this paper, we try different continuous regression models so that the news site does not lose the ability to choose what is considered popular or not.

Materials/Description of Data

In their original study, Fernandes *et al*. compiled a data set consisting of 39,000 observations including a single target variable for prediction and 47 predictor variables. Each observation corresponded to a unique article published by the popular online news website Mashable during a two-year period spanning January 7, 2013 to January 7, 2015. In our treatment of this data, we split the set into a training set consisting of 27,750 of the 39,000 observations, and a test set containing the remainder.

The variable to be predicted, shares, gauges the popularity of a given article via the number of times it is shared by Mashable users. The 47 predictor variables describe a variety of pre-publication attributes of the articles and are placed into six broader categories by Fernandes *et al.*: “words”, describing the usage of language in the article (such as the number of words in the title, number of words in the article, average word length, etc.); “Links”, containing variables describing the number of other Mashable articles linked to that in question, and the respective numbers of shares of those articles; “Digital media”, counting the number of images and videos contained; “Time”, indicating the day of week of publishing; “Keywords”, counting the number of keywords, and the number of shares of previous articles sharing the same keywords; And “natural language processing”, containing variables which use NLP methods to derive such measures as the positivity, negativity, subjectivity, and polarity of words appearing in the article, as well as the similarity of the topic of the article in question to those of the five most popular Mashable articles prior to publication.

While the division of variables described above is not one which is explicitly taken into consideration when conducting analyses, it is worth briefly discussing another division which is. Quantitative variables make up the bulk of our data – including the dependent variable, and 45 of the 47 independent variables. Quantitative variables may be further divided into those described by unbounded integers, and those given by a ratio (bounded within zero and one). Number of words, number of links, and number of videos in the article are examples of the former, whereas the rate of unique words, the rate of positive words, and the rate of negative words in the article are examples of the latter. The two categorical predictors are the day of week of publishing, and article category (world, business, entertainment, etc.), each of which comes pre-coded with dummy variables in the source data set. We also note that no variable contained missing values, so that no imputation was necessary to treat the data. Data cleaning involved removing URL columns which contained no information relevant to our analyses.

Our target variable, shares, is one which is particularly challenging to deal with in the task of modeling predictions and is so due to a combination of several factors. Firstly, shares is heavily right-skewed, with large gaps (discontinuities) between its extreme points. This is best illustrated by the five number summary of shares (min, Q1, med, Q3, and max are respectively given by 4, 942, 1400, 3379, 2800, 843300). This problem is further illustrated by a histogram of shares (Figure 1). A second issue is that shares apparently holds no linear relationship with any of the predictors (and in many cases, no discernible relationship at all). This is illustrated by both a correlation matrix, which reveals the highest magnitude of correlation between shares and any single predictor to be 0.11, and plots of shares related to each predictor (see examples in Figures 2,3,4). Consequently, we expect linear models to perform particularly poorly, and to be outperformed by non-parametric models. We also suspect that, due to the nature of the skew in shares, some of the necessary conditions of linear regression will likely be violated in our models - particularly the normality and zero mean of error terms. This would necessitate log-transforming shares (and possibly right-skewed predictors as well) in order to satisfy the conditions of linear regression. Lastly we note that, because of the characteristics of shares described above, we realistically expect high values of MSE for our predictions regardless of the model chosen.

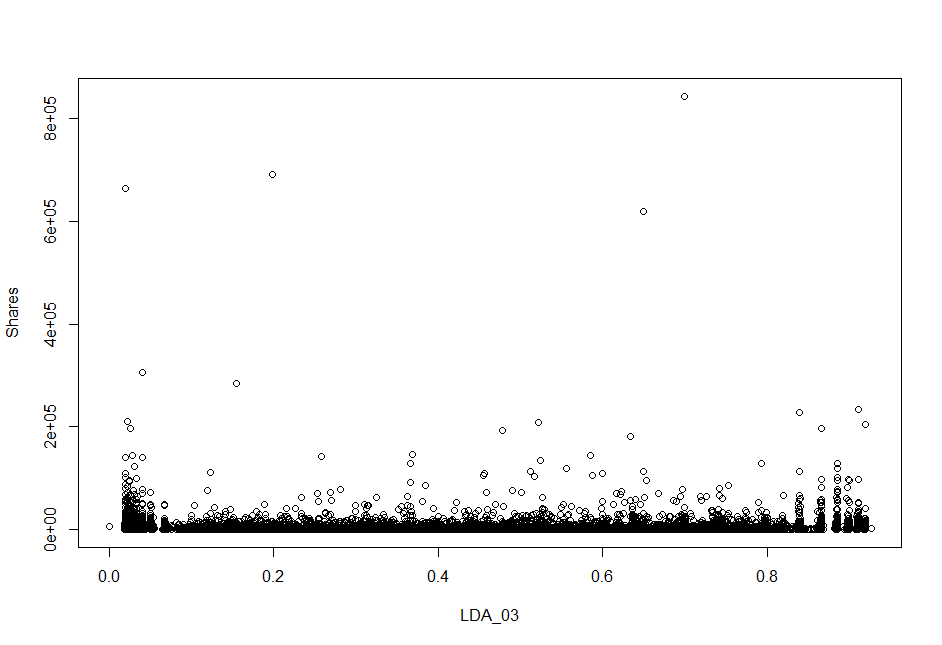
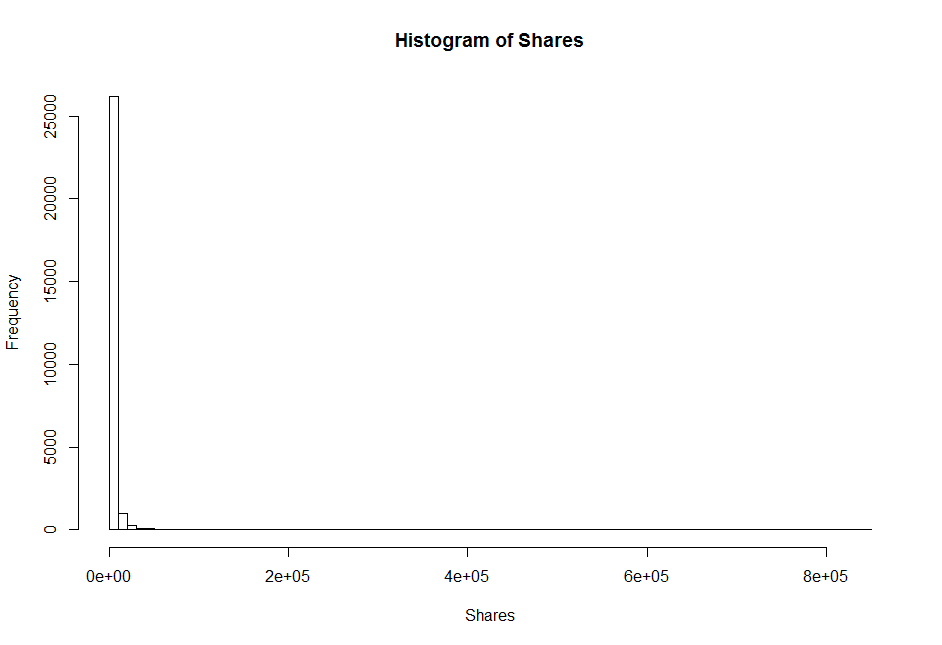


Figure 1. A histogram of shares, our variable for prediction. Figure 2. The relationship between shares and closeness to

the third most popular Mashable topic prior to publishing.

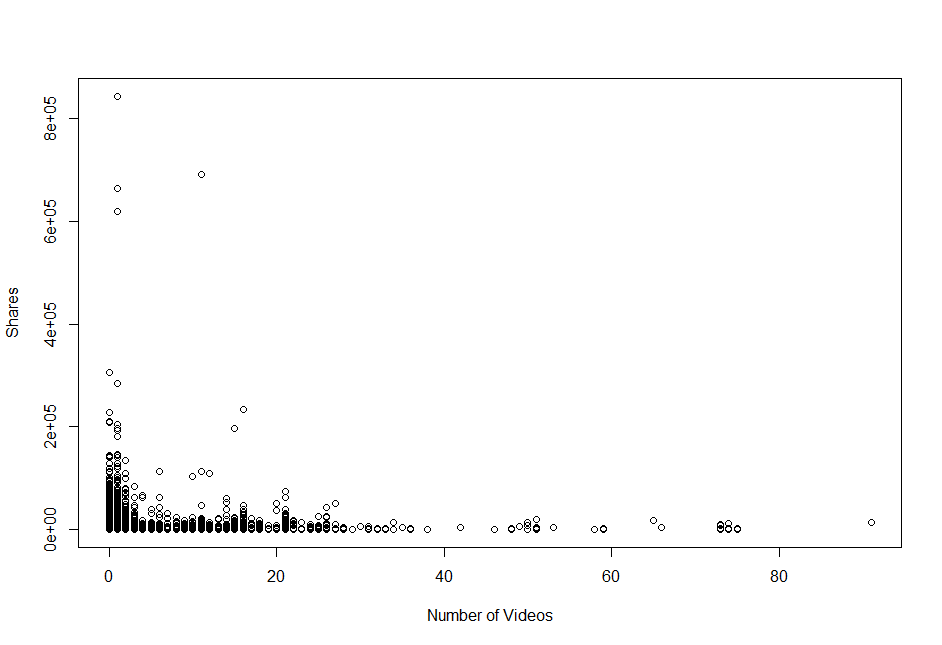
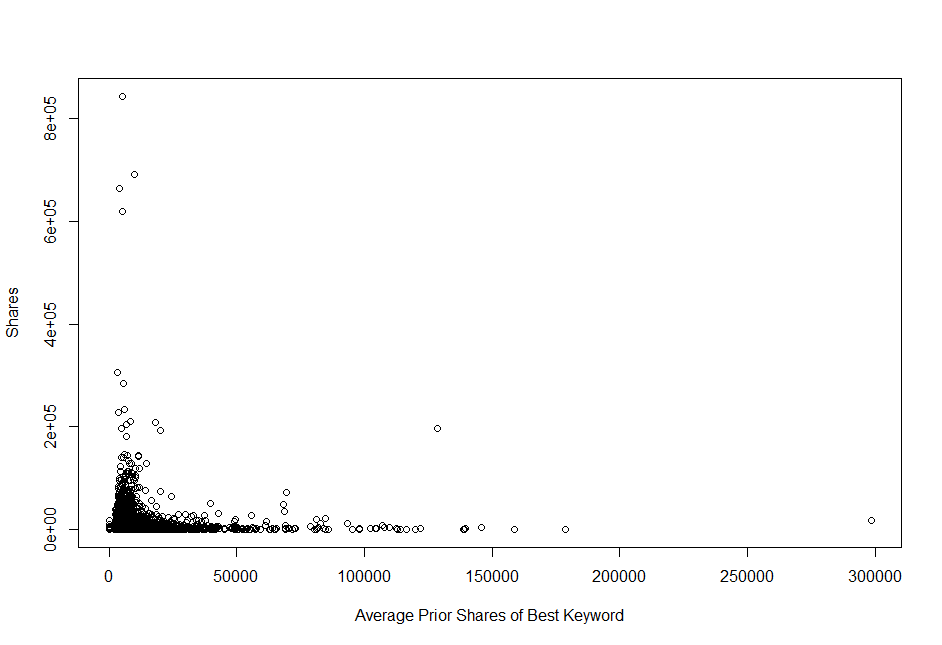


Figure 3. The relationship between shares and prior shares of Figure 4. The relationship between shares and the number

an article’s most popular keyword. of videos contained in the article.

Methods and Results

Lasso Regression

For our regression models, we partitioned our data into training and test sets with 70% and

30% of observations being allocated to each, respectively. As a preliminary model, (without log transformation) we ran lasso regression on shares as a function of all predictors using the glmnet package in R. We selected our tuning parameter, λ using a grid search for 100 values within the interval of 10 and . After choosing the value of λ which minimized cross-validated error, we evaluated the model against our test set. Due to concerns of the regression conditions possibly being violated, we turned to a residual plot, which confirmed our suspicion of skewed residuals (Figure 5).

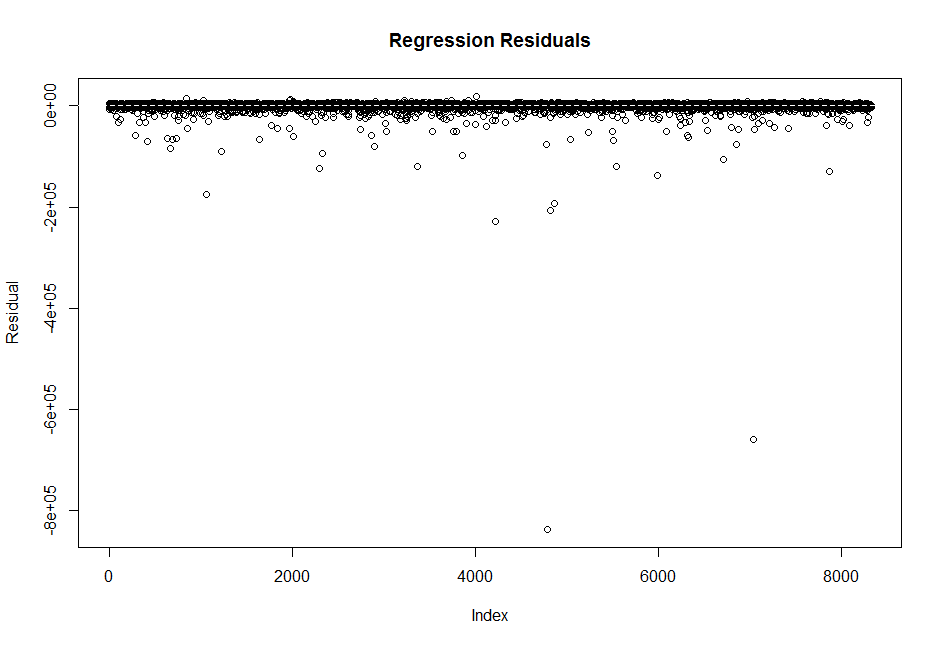
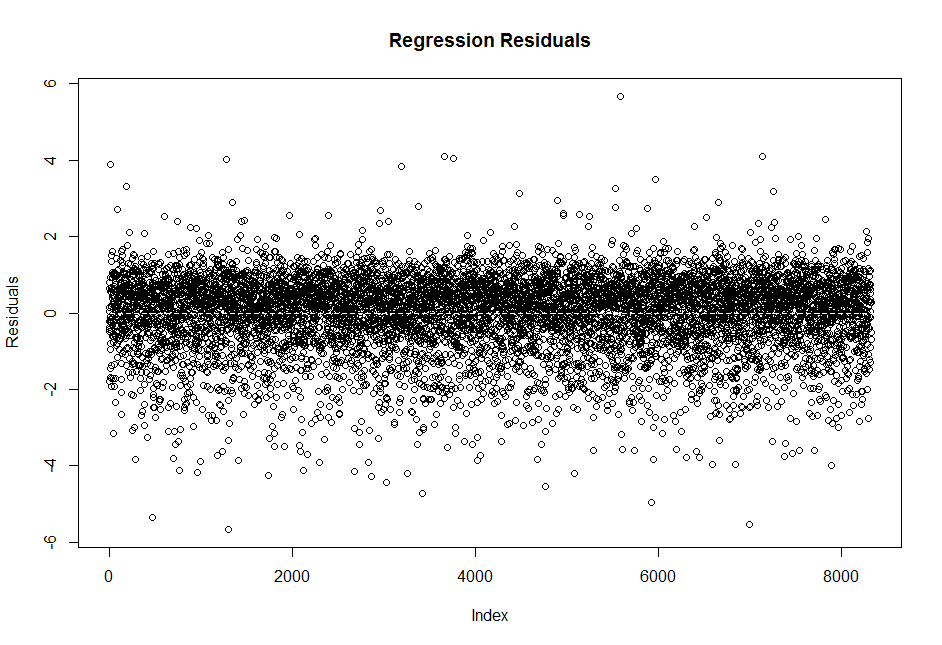
 

Figure 5. Lasso residuals of un-transformed data. Figure 6. Lasso residual of log-transformed data.

We then log-transformed our dependent and unbounded predictor variables in order to re-attempt the lasso model, and did so after confirming the normality of our new Y variable and the resulting residuals (Figures 6). All subsequent regression models used these transformed variables.

When attempting to re-create the lasso model after transformation, we carried out a new grid search for the optimal value of λ (Figure 7), using a plot of cross-validated error to decide to use the boundaries and . Our model keeps only 12 of the 56 original variables (including dummies) after shrinkage and variable selection. This approach yields a λ of approximately 0.043 and a test MSE of 194,446,334.

Ridge Regression and Ordinary Least Squares

When carrying out ridge regression, we repeated the procedure used in constructing the lasso model. Doing so yielded λ of (Figure 8) and a test MSE of 193,574,888. Similarly, for ordinary least squares, we set λ to 0 and obtained a test MSE of 193,575,325.

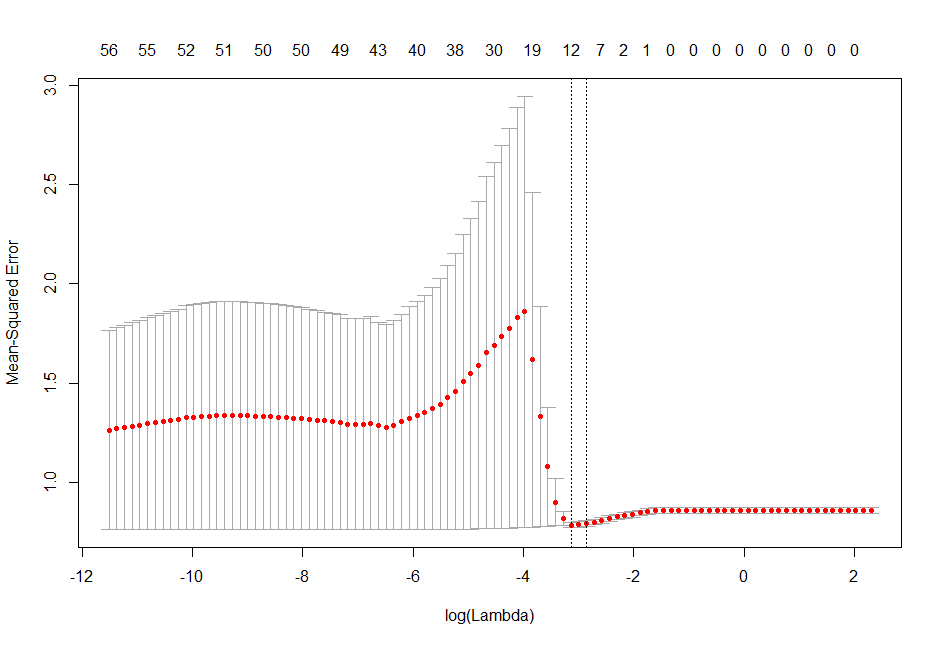
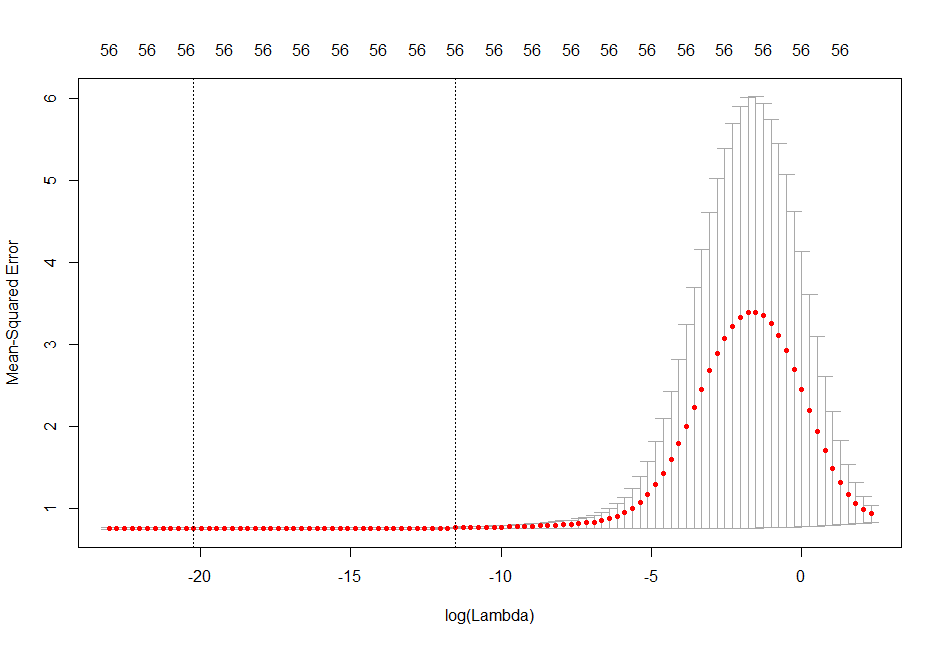
 

Figure 7. Finding the optimal λ for the lasso. Figure 8. Finding the optimal λ for ridge regression.

Stepwise Selection

We constructed both forward selection and backward selection models. The forward selection model used 36 of 56 predictors and produced a test MSE of 193,581,187. The backward selection model used 42 of 56 predictors and produced a test MSE of 193,576,305.

Principal Component Regression and Partial Least Squares

Each of our PCR and PLS models performed optimally when all, or nearly all components were included in the model (in the cases of PCR, and PLS, respectively) and performed similarly to the other regression methods previously discussed. Therefore, we decided not to use either and do not discuss them further.

Tree Based Methods

For tree-based methods, the data was split in half, where one half was used for testing and the other half was used for training. As with all the models, we tried to get a MSE value so that we can compare the model.

Random Forest Model

Using the randomForest package in R, we tried to build the best possible model by getting the optimal mtry values and the number of trees. We did not perform cross validation because variable selection is random when building the tree, so there is no tendency to overfit. The mtry parameter is the number of parameters that the model will look at to build each tree.

To start, we looked at the values 1-10, 15, 25, and 30 for mtry and 250 trees.

|  |
| --- |
|  |
| Figure a. Random Forest model test MSE with mtry values 1 – 5 and 250 trees. |

|  |
| --- |
|  |
| Figure b. Random Forest model test MSE with mtry values 5 - 9 and 250 trees. |

|  |
| --- |
|  |
| Figure c. Random Forest model test MSE with mtry values 5, 10, 15, 25, 30 and 250 trees. |

Figure a, b, and c show that mtry of 2 has the lowest test MSE. We extended the number of trees from 250 to 500 to see if the MSE value will go any lower as the number of trees increase.

|  |
| --- |
|  |
| Figure d. Random Forest model test MSE with mtry values 1 - 5 and 500 trees. |

We chose mtry = 2 and 400 trees as our model. It provided a test MSE of 174472413.

|  |
| --- |
|  |
| Figure e. Variable importance from random forest |

Boosting, Gradient Boosting Model

Using the gbm package in R, we tried to get the best possible model by tuning shrinkage, tree depth, number of trees, minimum observation in terminal nodes of the trees, and bag.fraction, the fraction of training observation randomly selected to build the next tree.

We looked at shrinkage values .01, .05, .1, .2, .3, .4, .5; interaction.depth (tree depth) of 1, 3, 5; build up to 500 trees, a minimum observation in terminal nodes of 5, 6, 7, 8; and a bag.fraction of .5, .75, 1. The following models gave the best cv.error. We used cv.err to compare different gbm models with different parameters even though we did not set a cv.folds value as the object returns it and no further computation is needed. Using the best parameters, we then looked at shrinkage value .2, .3, .4, .5; interaction.depth 1, 3, 5; a minimum observation in terminal nodes of 6, 7, 8; and bag.fraction of .5, .6, .7.

|  |
| --- |
|  |
| Figure f. Best gbm model parameters and the associated c.v. error |

|  |
| --- |
|  |
| Figure g. Best gbm model parameters and associated c.v. error |

We can see that the best model has the shrinkage of .5, interaction depth of 1, number of trees at 35, minimum observation in terminal nodes of 7, bag.fraction of .6. Using these values, we got a test MSE of 122852790.

|  |
| --- |
|  |
| Figure h, i. Top ten parameters according to gbm |

Discussion

As expected, regression models were outperformed by both non-parametric techniques that were attempted. A comparison of the test MSE values obtained from our regression models and tree-based models is given below (Figure 9).

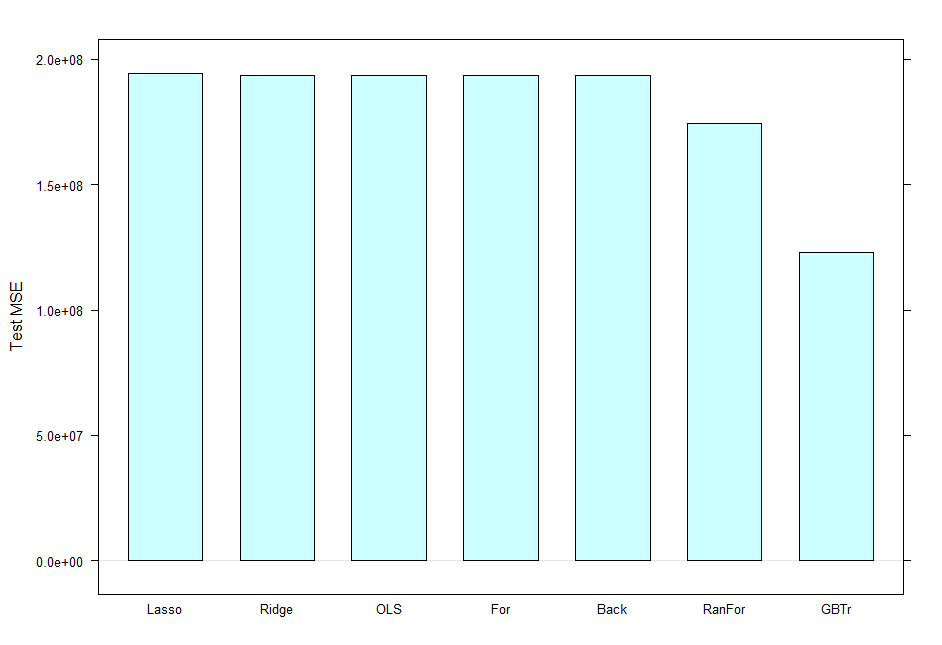


Figure 9.

We chose the GBM boosting model with shrinkage of .5, interaction depth of 1, number of trees at 35, minimum observation in terminal nodes of 7, bag.fraction of .6. We chose this as the final model because it had the lowest relative MSE to the rest of the models that we tried. Boosting gradient model had a test MSE of 122852790 when using the best parameters, while random forest had a best test MSE of 174472413.

From the random forest and the boosting methods, we saw several of the variables consistently in the top ten for importance to the data. The variables kw\_avg\_avg, kw\_max\_avg, kw\_max\_min, kw\_min\_avg, self\_reference\_min\_shares, and kw\_min\_avg appeared at least twice in the list of variables from gbm and on the %IncMSE and IncNodePurity scale from random forest. This shows that keywords and the shares of mentioned articles in articles are important to how often the article is shared. This could be because people are arriving to the article through keyword searches in search engine. As the number of people who read the article increases, the greater the chance that people will share the article. Other than the keyword variables, it seems like the self\_reference\_min\_shares variable is important. This variable describes the minimum shares of referenced articles. Related articles are usually linked in this article and if this article was shared, it can be inferred that the related article has a good chance of being shared because of the related topic and similar readers. Surprisingly, the time of publication is not important. This could be because online news is available all the time, unlike traditional print media such as the Sunday newspaper.

The variables that were important for the nonparametric models had some similarities to the tree based methods. The variables self\_reference\_min\_shares and kw\_avg\_avg were important in ridge regression, lasso regression, gbm, and random forest. This means that the most consistently important variables were self\_reference\_min\_shares and kw\_avg\_avg. This makes sense as readers of articles could increase with more keywords optimized for search engines and readers are more likely to look at related articles of the same topic. However, it appears that the ridge and lasso regression model also considered the number of images, the links, the data channel and if the article was published on a weekend.

One way we can reduce the mean square error would be to gather more samples. This is because the mean square error is a measure of the variance of the data and as the amount of data increases, the variance decreases. Gathering more samples would not take much more work as the script for data scraping was already written. It should be noted that as the number of data points increase, the longer it will take for the computer to generate the model for the nonparametric methods. This is due to the computational complexity of tree building and memory usage.

The most obvious area for improvement in future studies, however, would be the identification of variables with higher explanatory power. Of those included in the present study, the predictor which held the strongest linear correlation with shares had a correlation of only 0.11, which leads us to believe that the quality of our predictors was likely the greatest shortcoming of our models. While the search and identification of such variables is a task with no set procedure, and so would require greater creativity on the part of those carrying out the study, we believe that prioritizing this task would be the most worthwhile course of action for future researchers. Possible ideas for new predictors include: a measure of novelty, as the novelty of a topic is undoubtedly a factor in piquing the interest of readers. Just as predictors in the present study used natural language processing methods to quantify the similarity of an article to the then-most popular topics on Mashable, perhaps (if NLP methods allow) a measure of dissimilarity from any recently published articles could also be taken. Another possibility is a predictor gauging the recent popularity of the topic on websites other than Mashable, such as, for example, Mashable’s major competitors, as well as general search engines. As different readers may, despite sharing similar interests, often be loyal to and seek out information from different sources, the recent popularity of an article topic on Mashable’s competitors is likely to mirror the popularity the topic will gain on Mashable, possibly even more so than it would the topic’s recent popularity on Mashable itself. A final proposition posed by the authors would be to fine-tune some of the present predictors to improve them. The article category variable, for example, places articles into very general categories and likely generates noise in the process. It is quite possible that these broad categories discard some potential discriminatory power offered by their constituent subcategories – for example, within the “entertainment” category, it could be the case that Mashable readers are highly likely to share articles on video games, but not so much so for those concerning TV shows, or vice-versa. This would be a potential area for exploration in uncovering presently concealed information hidden in the data.

Citations:

K. Fernandes, P. Vinagre and P. Cortez. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal.