Corinne Smith, Erin Lee, Adam Gonzalez, Jon Le, Debbie Lu

Dr. Hersh

MGSC 310-01

18 December 2020

MGSC 310 Final Project

Introduction

Our research question was: what are the factors that predict average book ratings and how can we predict how a book will be received by the public? We analyzed Goodreads datasets and built models to try to answer this question. The guiding principle for our analysis was to start with simpler supervised learning models and increase the level of complexity in response to our results. We knew that we would employ regression analysis techniques since we were predicting a continuous variable. In order to predict a quantitative outcome based on a variety of factors, we leveraged a variety of models: linear regression, elastic net, bootstrapped decision tree, and random forest. This summary of our findings provides information accounted for by our four models as well as revealing what information could not be determined by our models.

Motivation

In the big picture, we want to predict book ratings and find which factors affect book reviews the most, and upon understanding those factors ourselves, to know what to read, buy, and sell. We knew that book reviews on Goodreads provide the information of readers' opinions or editorial reviews about a book along with a simple rating (a Likert scale of 1-5). However, common sense indicates that book reviews contain more qualitative than quantitative data. Another consideration is that book reviews are subjective, so scoring ratings may differ in weight by person. An example of this would be a reviewer giving a book 5 stars for simply liking it, whereas another reviewer who liked the book equally may give it 3 stars. Acknowledging these realities, we wanted to understand the biggest predictors of higher or lower ratings and the factors that affect the book rating.

Data Overview

On Kaggle, we obtained a books dataset scraped from the Goodreads API that contained many variables relating to the books on Goodreads, including average book ratings, the number of pages in the book, how many reviews had been published about the book, etc. To supplement this data, we found another dataset from Goodreads containing more information about authors. We joined these two datasets, matching authors to all of their books in the original dataset; all the books left had single authors. We did this to better understand the story behind each book; the relative fame of an author affecting someone's opinion on their work seems plausible.

Another idea we wanted to take into account was the feeling evoked by the book's title. To do this, we used the package sentimentr to gather the sentiment of the words used in each title. This package looked at the positive or negative connotation of each word, put them in context (i.e. "happy" is positive but "not happy" is a negative statement), and then averaged those connotations for the entire title.

In terms of data cleaning, we made sure to remove all missing values from the columns, only leaving them in the author birth and death date variables that we planned not to use for any modeling. There were many duplicated titles of books and books without any ratings, so we removed them. In terms of outliers, the author "Anonymous" had hundreds of thousands of works attributed to them, so we figured it was best to eliminate their one book in our set. In terms of outliers in page numbers, there were few books that had page counts of over 2,000. Most of those were collections, and the majority of the books in the dataset were under 2,000 pages, so we chose to cap the page count at 2,000 and set the minimum to 10 pages. To simplify our dataset, we eliminated variables such as author location, the ISBN of the book, author Twitter username, and other less useful information that should not show an effect on the book's average rating. This set us up with a clean and clear dataset to utilize for solving our problem.

Summary Statistics:

Variable	*Min*	*Mean*	*Max*	*St. Dev*
Average Book Rating	1	3.9	5	0.3

Number of Pages	10	350.4	1952	195.2
Book Ratings Count	0	21480.8	4597666	120423.5
Text Reviews Count	0	696.7	94265	2844.8
Average Title Sentiment Score	-1.4	0	1.3	0.3
Author's Work Count	1	231	5204	488.1
Author's Fan Count	0	12050	709826	55558.6
Author's Ratings Count	27	658708	24511114	1727055.4
Author's Reviews Count	1	26511.1	579250	58813.9

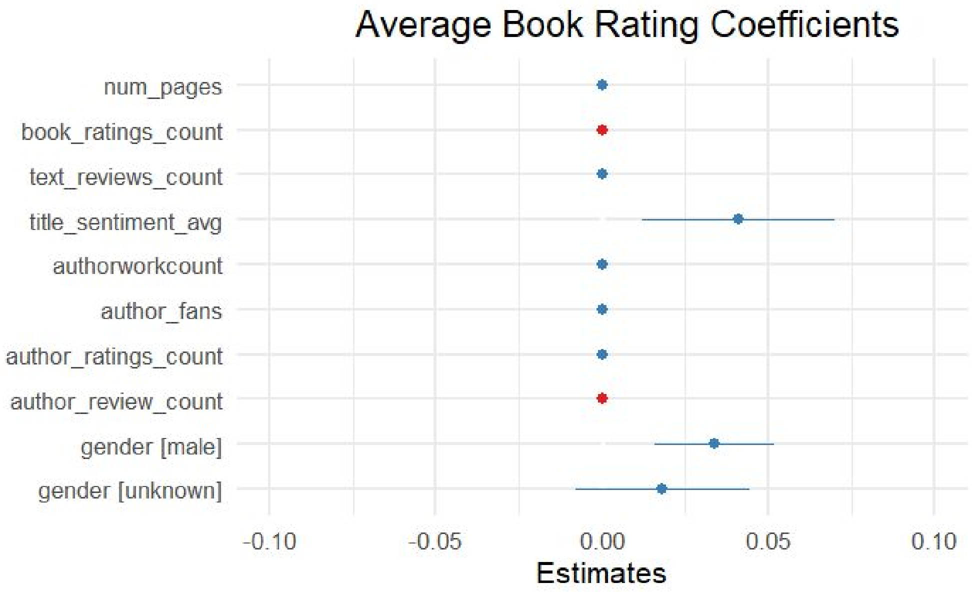
Models

We first selected the set of variables to use in all four models. We chose the response variable of average book rating and nine predictor variables of number of pages, book ratings count, text reviews count, title sentiment average, author work count, author fans, author ratings count, author review count, and gender. We did not know in advance which sort of model would work best with our data. However, the linear regression provided an ideal starting place as it is both straightforward in terms of the build as well as interpretability of results.

Linear Model

Since linear regression is both sensitive to outliers and prone to multicollinearity, we were careful to account for these potential issues during our data cleaning steps. The parameters for this model, the regression coefficients, tell us how much our response variable, average book rating, could be determined by our predictor variables.

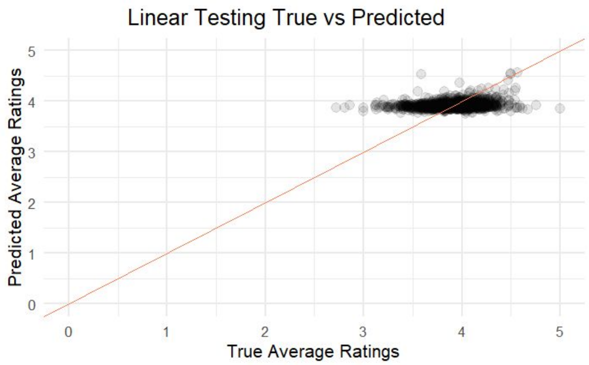
For model validation, we used the train-test-split method to create two subsets of our original data set; 80% of the data was used for training our model and 20% was left for testing. When we tested the model, we saw quite a few variables with small regression coefficients, meaning small relationships to our outcome, which could be due to random noise or variation in the data.

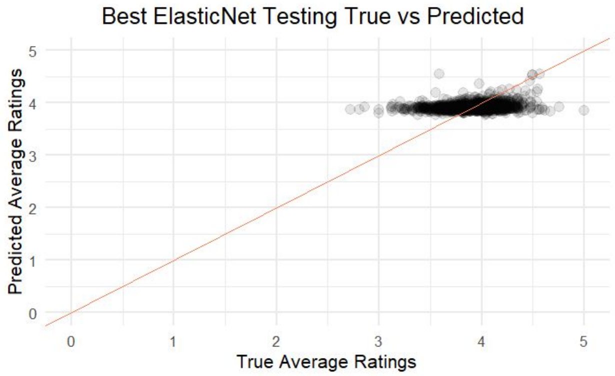


In terms of performance, we saw a decrease in the R-squared (rsq) value from the training data's 6.87% to the testing data's 6.59%. An ideal rsq would be close to one, so this low rsq value indicates that a linear model does not accurately predict the average book ratings. With the slight decline in the already low rsq value, the model being overfit seemed possible. To see if we could improve the model, we turned to dimensionality reduction to help determine which variables were most important to our predictive model. We used this technique in order to minimize or eliminate any extraneous coefficients to make sure that the relationships between our predictors to the outcome were legitimate. To move forward, we built a model to perform either Lasso, Ridge, or a combination of both regularizations.

ElasticNet Model

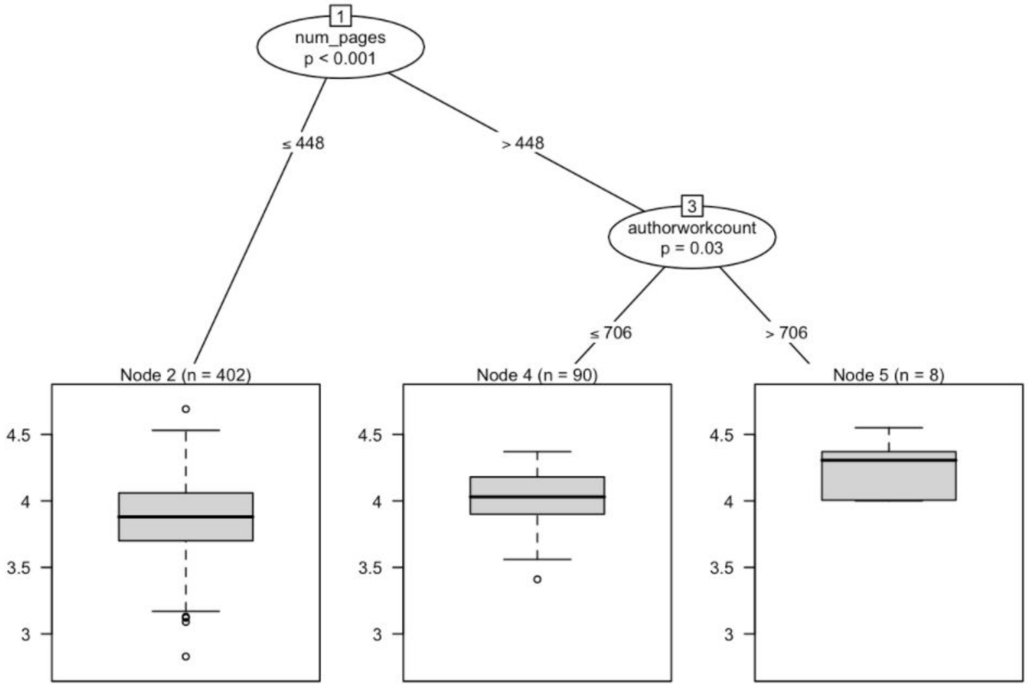
Modeling with ElasticNet allows us to find an optimal combination of Lasso or Ridge regularizations, and we were able to isolate the model returning the least amount of error. Our best model gathered from ElasticNet had an alpha value of 0.1, which utilizes mostly Ridge and slight Lasso regularization. We used the minimum lambda value, which minimizes cross-validated error. This gave us a model with an rsq that changed from 6.87% to 6.57%. These rsq values are almost identical to those of the original linear model. Regularization did not significantly improve the linear model, further driving home the point that linear regression does not successfully predict average book ratings from our predictors. Below indicates how similarly poor these linear models performed:





Bagged Decision Tree

Witnessing the failure of these models was a pivotal moment as we recognized that a linear model was not appropriate for our data. Since we needed a non-linear form of modelling, we utilized tree-based methods, such as the Bagged Decision Tree and RandomForest models. Instead of deriving a trend line that passes through or near as many data points as possible, this approach divided the data based on a series of binary decisions into predictor spaces. We started by bagging a decision tree model, and one of its tree has been transplanted here:

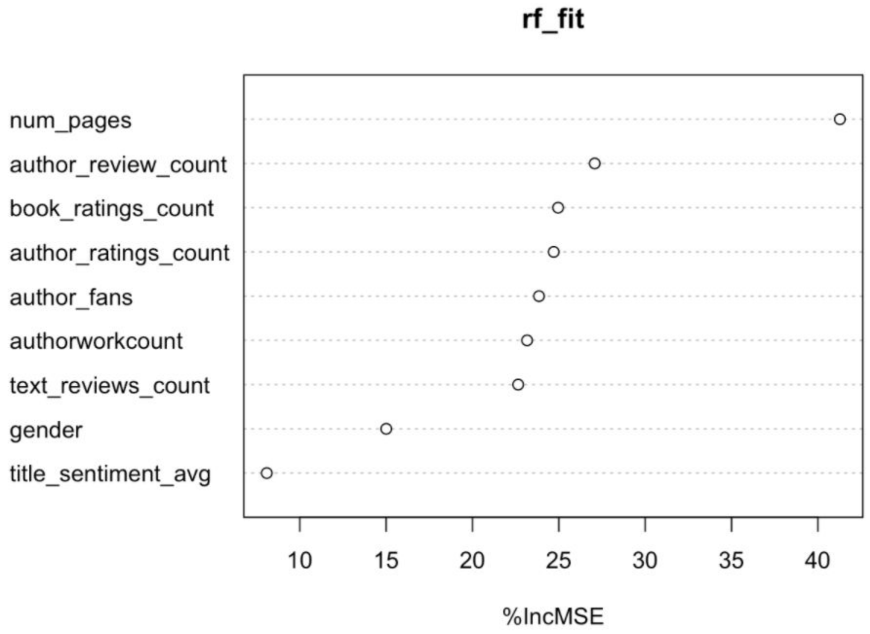


We bootstrapped 100 trees with each using 500 lines of randomly selected data. As an aggregated model, the bagging helped avoid overfit and diminished variance. Our out-of-bag predictions returned an rsq value of 10.58%. This was about a 3-4% improvement from our previous linear-type models, but the model still performed poorly.

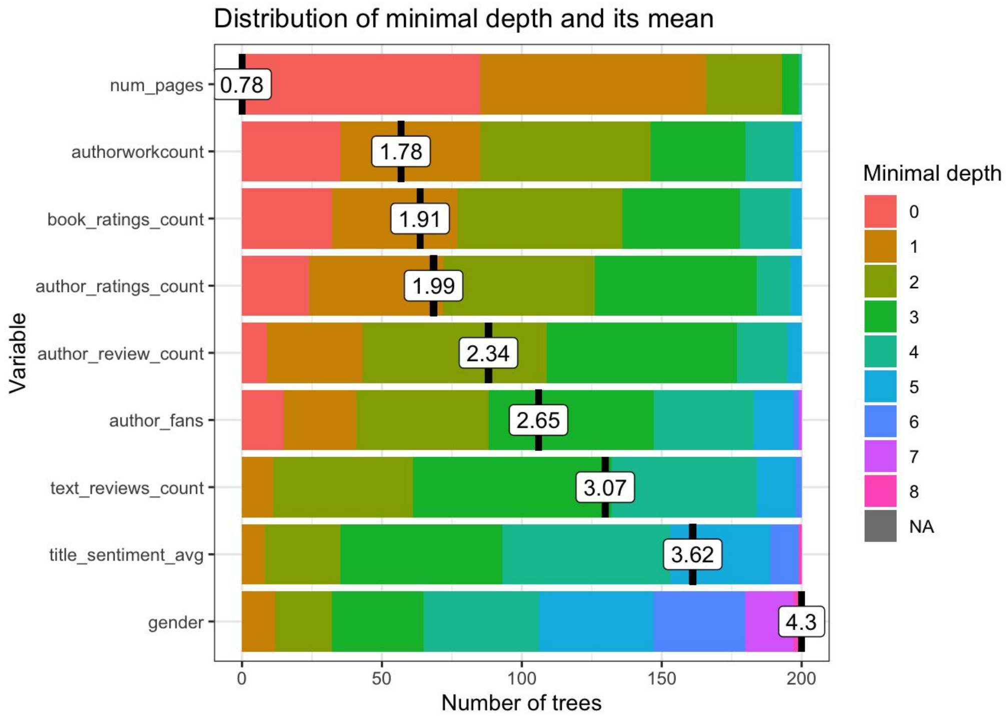
RandomForest

Finally, we turned to a RandomForest model. The random forest provided the flexibility needed to achieve a greater prediction accuracy. We chose to test 4 variables (using the heuristic of 11/3 predictors rounded to the next nearest integer, as the gender variable has 3 factor levels) at each split, and created 200 trees. To make this model less computationally intensive, 100-150 trees should be used because they provided a low error rate similar to that provided by 200 trees.

In the following plot, the nine predictor variables show how much being replaced with random noise would impact the model. Removing the number of pages (num_pages) increased the error of the model (%IncMSE) by about 40%, the most out of any of the variables. Variables with a high %IncMSE demonstrate importance in this model, so the number of pages, author's reviews count, book ratings count, author's ratings count, number of fans of the author, author's work count, and the amount of published reviews written about the book were the most significant.



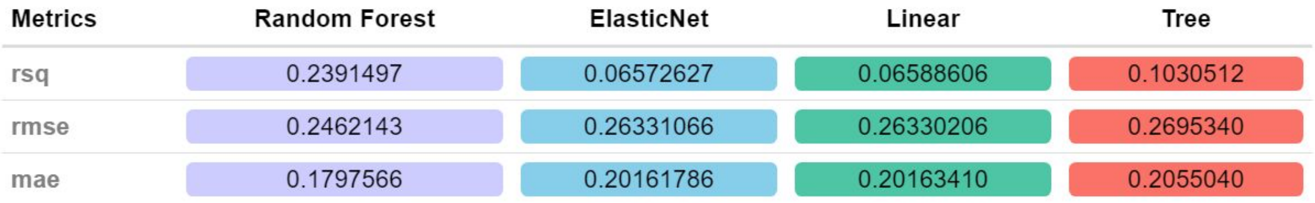
We can also see how important variables in RandomForest are by looking at their minimum depth. This plot reframes variable importance by showing when the model makes decisions with them in its trees; the sooner or shallower implies higher importance.



Overall, this random forest model performed relatively well compared to the others. Its out-of-bag predictions returned an rsq of 23.91%, around a 13% improvement from the bagged decision tree model. However, an rsq of 23.91% does not give us a model that can predict a book's average rating with confidence.

Model Analysis

None of the models we tested returned rsq values higher than 24%. Our two error metrics were the lowest with the random forest model, and its rsq was by far the highest at 24%, making it the most "successful" model.



Metrics	Random Forest	ElasticNet	Linear	Tree
rsq	0.2391497	0.06572627	0.06588606	0.1030512
rmse	0.2462143	0.26331066	0.26330206	0.2695340
mae	0.1797566	0.20161786	0.20163410	0.2055040

Conclusions and Beyond

We do not believe that any of our models are ready for production. Our results indicate that we cannot model how average book ratings are decided using our datasets.

There are several things we'd like to change or add to the data to improve our results. For one, this data is missing important consumer information; we only know average ratings right now, we have no idea about the people who have submitted ratings and reviews for the books. There is no breakdown of the composition of the average ratings; an average of 3 stars could possibly contain half 5 star ratings and half 1 star ratings. This would give insight into more of the book's reception: generally well-liked, half-hated or half-loved, or generally disliked. It would be great to know some consumer demographic data, so we could predict whether certain groups would like a book or not. In addition, it would be interesting to see a dataset for one book, with all the reviews thereof with relevant consumer data. This information would be great for a book selling website or review aggregator to recommend certain titles to readers based on what kind of books similar customers have enjoyed.

Additionally, when we take a look at our results from a business perspective, we do not have the right information to make significant decisions. Data like profit and sales figures are absent from the dataset, crucial information for book-selling businesses. The idea of business success differs from average ratings or even general ratings. If a poorly-rated book still sells well, the factors that make that book profitable need to be found and understood.

From the beginning, we knew we had pre-existing patterns in our data with which we could use to make a prediction, making this a supervised learning problem. However, with more data on customers, we could utilize unsupervised machine learning techniques to cluster the groups of people who might be more likely to leave a rating and other useful insights for business. This would be useful because we do not feel our model paints a complete picture for business use. Our model does not take into account how the information on the customers themselves might factor into the book rating, such as what are the different types of customers. Any data that would provide customer segmentation would be helpful from a marketing perspective. The analysis, as is, is not reliably applicable for use by authors, publishers, or customers.