TEXT-BASED PREDICTION OF BOOK REVIEW POPULARITY

Bridget Daly
Department of Statistics
Stanford University
bdaly2@stanford.edu

1 Introduction

Goodreads is a popular book cataloguing and community site, allowing readers to track, rate and review books as well as like and comment on others' reviews. These reviews are sorted on a book's home page by an unknown default algorithm which tends to present reviews with many likes and comments first. Higher visibility leads to more likes and comments. Reviewers with a strong presence on Goodreads are more likely to receive Advanced Reader Copies, books distributed free to a select audience before they are published for the general public [1]. Given the perks of writing good book reviews - increased readership from prominent page placement and free reading material - this research explores the syntactic and semantic elements that distinguish popular reviews from unpopular ones by building popularity classification models and evaluating feature importance.

Review popularity has been explored primarily in the context of online reviews after product purchase. One study identified valence, depth, age, credibility, and framing as characteristics influencing review popularity and helpfulness using a Poisson GLM to predict review votes [14]. Another found that textual review variables contributed more to popularity than non-textual variables using XGBoost and skip-gram to predict review readership [10]. It is not clear that these findings would hold for book reviews, as product reviews motivate future purchases while book reviews motivate future readership. The evaluation of a book might involve very different elements than the evaluation of a utility-focused product. Previous research using book review data focused on building book recommendation systems [11] and evaluating broader book impact [7, 13]. This study is the first known to explore book review popularity.

2 Dataset and Features

2.1 Data Collection and Processing

Two datasets were used to analyze review popularity: a review dataset containing the text of 15.7M reviews along with metadata and a book dataset containing 2.3M books with metadata. The datasets [11, 12] were collected by researchers at UCSD in late 2017 and updated in May 2019 by scraping users' public shelves from goodreads.com. For this analysis, reviews with no likes or comments were removed. These made up 70% of reviews and were considered uninformative as it was unknown whether the reviews were unpopular or unseen by other users. Books with fewer than 10 reviews and books with fewer than 60 combined review likes and comments were removed as these books did not have enough readers to assess review popularity (respectively removing 14% and 42% of reviews). Finally, the fastText language detection model [9] was applied, and only reviews predicted to be English with probability at least 0.9 were retained (84% of reviews). The filtered dataset used for the remainder of the analysis included 1.98M reviews on 54K books.

2.2 Outcome and Feature Engineering

A binary review popularity indicator was constructed to serve as the outcome for classification. Each review's "likes share" was calculated, dividing the number of combined likes and comments on the review by the total number of combined likes and comments on all reviews of the book. This provided the review's relative popularity and allowed fair comparison between reviews on different books with varying readership. The binary review popularity indicator took value 1 for popular reviews with likes share greater than 0.02 (21% prevalence).

Three categories of features were developed: non-textual characteristics, textual characteristics, and text representation. Non-textual included number of reviews written by user, days since review posted, user book rating, and difference between user rating and average book rating. Textual included review length in words, average sentence length, average word length, percent nouns, percent verbs, percent adverbs and adjectives, quotation indicator, and compound sentiment score ranging from -1 to 1 (negative to positive). These were constructed using the Natural Language Toolkit NLTK, specifically the Punkt tokenizer, the Penn Treebank POS tagset, and the VADER sentiment analysis tools [2, 5]. Two methods of text representation were tested after removing stopwords and lemmatizing using NLTK: Bag of Words (BOW) on the top 10,000 most commonly occurring words and term frequency-inverse document frequency (TF-IDF). BOW represents each review as a collection of word counts while TF-IDF weights the word counts (term frequencies) by the inverse document frequency in order to emphasize rare terms over common ones.

3 Methods

A 15% holdout set was used to test accuracy of all models and a 15% validation set was used for models requiring hyperparameter tuning. Three types of models - logistic regression, gradient boosting, and neural network - were run on four feature subsets each: (A) 4 non-textual features, (B) 12 non-textual and textual features, (C) non-textual and textual plus BOW, and (D) non-textual and textual plus TF-IDF. Because the classes were unbalanced (79/21%), the training set was undersampled so that the number of unpopular reviews equaled the number of popular reviews. For each model an appropriate feature importance measure was applied as described below.

3.1 Logistic Regression

Logistic regression is a generalized linear model for a binomial outcome in which the logit of the probability of the outcome being 1 (the review being popular) is modeled as a linear function of the features with coefficients for an intercept and each feature determined by maximum likelihood.[1] Because of the number of features present in the text representation subsets C and D, regularized logistic regression was applied with an L2 penalty determined using the validation set. The L2 ridge penalty constrains the sum of the squared coefficients and has the effect of shrinking coefficients towards zero thus reducing model variance.

Logistic regression was chosen to serve as a baseline model because the learned model coeffecents have a natural feature importance interpretation with the coefficient indicating the increase or decrease in logit probability with a one unit change in feature value. For subsets A and B features were not scaled (doing so greatly diminished model accuracy), so features could not be ranked based on magnitude but the sign could indicate the direction of impact. Features were scaled for subsets C and D allowing a ranking of feature importance based on absolute coefficient magnitude.

3.2 Gradient Boosting

A decision tree uses a series of splitting rules based on feature values to predict an outcome. Gradient boosting is an ensemble learner made up of many decision trees, with each tree added sequentially to the model to minimize loss. Extreme gradient boosting (XGBoost) [3], an algorithm used to efficiently train with L2 regularization, was used to build a decision tree ensemble for each feature subset. The validation set was used to determine the appropriate number of trees to use, the maximum depth of each tree, and the learning rate. XGBoost was chosen for its previous success in predicting online purchase review popularity with text representation [10] and for a natural feature importance interpretation based on the number of times a feature is used in a decision tree to make a split.

3.3 Neural Network

A neural network was chosen for the final model type as this has been shown to outperform simpler models for text classification [4]. A simple network with one densely connected hidden layer was used for all feature subsets. The number of input nodes was determined by the number of features in the subset, the number of nodes in the hidden layer was set to approximately two thirds the number of features[2], and the output layer had a single node for the binary classification. The rectified linear activation function (ReLU) was used for the input and hidden layers and the sigmoid function for the output. The neural networks were trained through cycles of forward and backward propagation in which output was calculated based on current network parameters then mini-batch gradient descent was used to update the parameters based on the cross entropy loss function. Training stopped when the validation loss did not reach a

[1] Models for subsets A and B were optimized via the Newton-Raphson method as implemented in the Python's statsmodels and subsets C and D via the SAGA incremental gradient method as implemented in Python's sklearn.

[2] Node selection was inspired by a post on StackExchange [6] with full awareness that these heuristics are often not optimal but do help in arbitrarily picking a number to start with.

Table 1: Model Evaluation Results

				Test Set Metrics			
Feature		Under					ROC
Subset	Model	Sample	Hyperparameters	Accuracy	Sensitivity	Specificity	AUC
A	Logistic	No		0.80	0.08	0.98	0.53
B	Logistic	No		0.80	0.11	0.98	0.54
A	Logistic	Yes		0.73	0.50	0.78	0.64
B	Logistic	Yes		0.71	0.57	0.75	0.66
C	Logistic	Yes	λ[a]$= 0.1$	0.73	0.60	0.77	0.68
D	Logistic	Yes	λ[a]$= 1000$	0.70	0.65	0.71	0.68
A	XGBoost	No	1000 trees, depth 6, α[b]$= 0.3$	0.82	0.23	0.98	0.60
B	XGBoost	No	1000 trees, depth 6, α[b]$= 0.3$	0.82	0.25	0.97	0.61
A	XGBoost	Yes	1000 trees, depth 6, α[b]$= 0.1$	0.69	0.64	0.71	0.67
B	XGBoost	Yes	1000 trees, depth 6, α[b]$= 0.1$	0.69	0.67	0.70	0.69
C	XGBoost	Yes	1000 trees, depth 4, α[b]$= 0.3$	0.71	0.69	0.72	0.70
D	XGBoost	Yes	1000 trees, depth 4, α[b]$= 0.3$	0.71	0.70	0.71	0.70
A	Neural Net	No	32 epochs[c]	0.80	0.11	0.97	0.54
B	Neural Net	No	157 epochs[c]	0.80	0.16	0.97	0.57
A	Neural Net	Yes	132 epochs[c]	0.68	0.62	0.70	0.66
B	Neural Net	Yes	175 epochs[c]	0.68	0.67	0.68	0.68
C	Neural Net	Yes	22 epochs[c]	0.69	0.71	0.68	0.70
D	Neural Net	Yes	24 epochs[c]	0.69	0.71	0.69	0.70

[a] Regularization strength
[b] Learning rate
[c] Epochs determined by early stopping as detailed in Methods Section 3.3

new minimum for 20 epochs, and the parameters were set to those from the epoch after which the model obtained the highest validation accuracy.

It is difficult to assess feature importance in a neural network given the complex nature of the model structure. Lundberg and Lee [8] introduced a framework called Shapley Additive Explanations (SHAP) for understanding predictions coming out of complex machine learning models based on Shapley values, a game theory method for dividing credit among cooperative players. SHAP values are based on conditional expected values of the model given values of input features. They are calculated for individual predictions and an overall assessment of feature importance can be achieved by taking the mean absolute value of the SHAP value for each feature over several predictions. Calculating SHAP values is computationally intensive, so for the neural networks a sample of 5000 training observations were used to calculated expected values and a sample of 1000 predictions were used to determine global importance.

4 Results

For test set predictions several evaluation criteria were calculated: accuracy (correctly classified/total), sensitivity (correctly classified popular/total popular), specificity (correctly classified unpopular/total unpopular), and Area Under the Receiver Operating Characteristic Curve (ROC AUC), the curve resulting from plotting the true positive rate (sensitivity) against the false positive rate (1-specificity) at different decision thresholds. Results are shown in Table 1.

5 Discussion

The models achieving the highest accuracies (80-82%) were those using the full training samples; however, these models also had very low sensitivities (8-25%) and ROC AUCs (53-61%). Further, since the prevalence of the outcome was 21%, a model simply classifying every review as unpopular would achieve an accuracy of 79%, so these highest accuracies are not very impressive. Given the goal of understanding the features important to review popularity, it was necessary to improve model sensitivity by under sampling the training set. While models trained on under sampled data achieved no more than 73% accuracy (logistic subsets A and C), sensitivity and ROC AUC were greatly improved.

Comparing the three model types, neural networks and XGBoost were similarly performing and slightly outperformed logistic regression with best sensitivities of 0.70-0.71 versus 0.65 and best ROC AUCs of 0.70 versus 0.68. In all cases,

Table 2: Logistic Regression Top 10 Features by Absolute Coefficient Magnitude

(C) BOW		(D) TF-IDF	
Feature	Coefficient	Feature	Coefficient
user reviews	0.64	user reviews	0.35
user rating	-0.55	user rating	-0.07
rating difference	0.51	number of words	0.06
number of words	0.40	avg word length	-0.06
"tm"[a]	0.09	"author"	0.04
avg word length	0.08	"short"	0.04
"feyre"[b]	-0.07	"book"	-0.04
"br"[a]	0.07	percent verbs	-0.04
"cinder"[b]	-0.06	"novella"	0.03
"author"	0.05	"arc"	0.03

[a] Author initials

[b] Book character or title

the best performance came from the models trained using text representations with little difference between BOW (C) and TF-IDF (D). However, there was not a great deal of improvement from using these text representations versus only textual and non-textual features (B), especially considering the massive difference in computational cost of training with 13 versus 10,000+ features.

A different type of feature importance measure was derived for each model type. While all measures ranked features based on determined importance, logistic regression was the only to indicate the direction of the feature's influence on popularity by the sign of the feature coefficient. Based on results from logistic regression on feature subset B, review length, average word and sentence length, and quotation inclusion improved probability of a review being popular as did the number of reviews a user has written. Negative reviews positively impacted popularity, as the coefficients for user rating and sentiment were negative. The coefficent for rating difference was positive indicating that disagreeing with popular opinion also boosted review popularity.

The top ten features based on absolute coefficient magnitude from logistic regression with text representation can be found in Table 2. These results indicate that even in text representation models, non-textual features were still the most important in classifying popularity. In both the BOW and TF-IDF models, the number of user reviews and user rating were the most important features. Number of words and average word length also appeared in both top ten lists. Four of the five most important words in the BOW model were book or author specific whereas all were general review terms in the TF-IDF model, reflecting a difference in how text is represented in each and suggesting that TF-IDF might be better for understanding book-agnostic keywords to include in a review to boost its popularity.

Number of user reviews was also the first or second most important feature based on number of splits for gradient boosting and mean absolute SHAP value for neural networks with every feature subset. Figure 1 provides the full ranking for subset B features. Rating difference was in the top three for both models and days since review in the top five, but there was some disagreement over the importance of the remaining features. For example, sentiment was ranked fourth in importance based on tree splits but last on SHAP value. On the other hand, user rating was ranked third in importance based on SHAP value but second to last on tree splits. These features provided similar information regarding the positivity or negativity of the review, so it seems each model found one valuable and the information conveyed in the other extraneous.

As with the logistic regression text representation models, the top features in neural networks were textual or non-textual features rather than word specific. Along with user reviews, the number of words in the review, days since review, user rating, and rating difference rounded out the top five features for both BOW and TF-IDF models. Unlike in the logistic regression model, top words were similar for the BOW and TF-IDF models; these included "book", "author", "character", "short", "think", "liked", and "loved".

The XGBoost text representation models differed from the other rankings in that word specific features dominated the top feature rankings. As with logistic regression, many highly ranked BOW features were book and author specific such as "katniss" (rank 2), "potter" (6), and "rowell" (7). Several of the top words - "exchange" (5), "thank" (11), "thanks" (15), and "free" (16) - were related to reviewers receiving ARCs in exchange for writing a review indicating that these reviewers had past success with review popularity and likely large user followings. The TF-IDF top words also tended

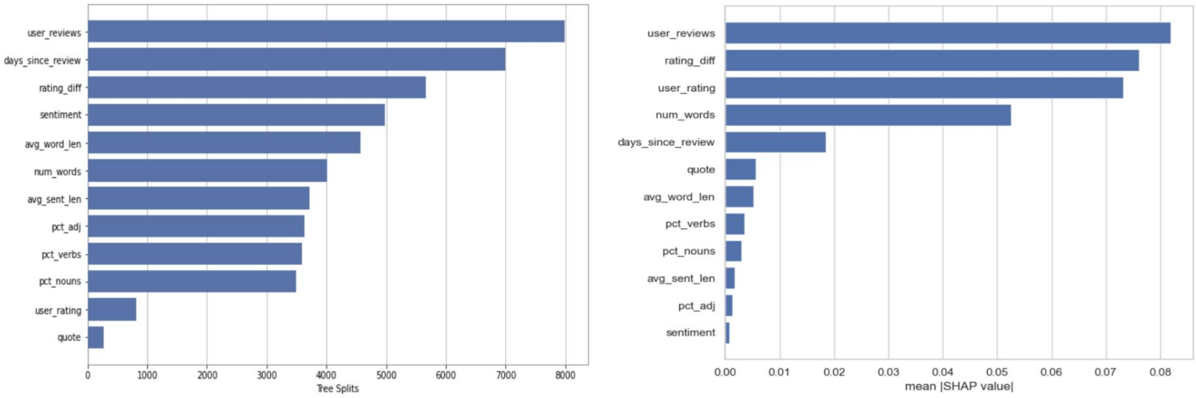

Figure 1: Subset B feature importance based on number of tree splits in undersampled XGBoost (left) and mean absolute SHAP value in undersampled neural network (right).

to be book and author specific and overlapped with the BOW top words, a difference from what was seen in the logistic regression and neural network models.

In constructing the binary popularity outcome, an arbitrary cutoff was chosen at a certain percentage of likes share. A significant difference between, for example, a "popular" review with a likes share of 0.025 and an "unpopular" review with a likes share of 0.015 was not expected. To understand if misclassification errors were driven by this arbitrary classification, the correlation between likes share and prediction probability was explored for two models. A high correlation was expected given that greater likes share implied greater popularity and a model should thus predict popularity with greater probability, but the correlation was only 0.29 for logistic regression with subset B and 0.33 for XGBoost with subset D suggesting room for model improvement. In exploring the distributions of features across reviews correctly classified as popular and incorrectly classified as popular (false positives), there was not significant difference whereas there were noticeable differences in comparison to the distribution of features across incorrectly classified unpopular reviews. This suggests perhaps that additional features that would help differentiate popular and unpopular reviews were required.

6 Conclusion

Logistic regression, gradient boosting, and neural networks were used to classify book reviews as popular or unpopular in order to understand the features important to writing a popular book review. The number of reviews a user has written was the most important predictor of popularity for almost all model and feature subsets. This suggests that reviewers with more experience write better reviews, but it also suggests an important missing feature which might explain the models' failure to achieve high accuracy. A reviewer with many reviews has probably amassed a large following, and more eyes reading a review means more likes on the review. Unfortunately user followers was not available in the dataset but would be an important feature to explore in future work.

Review length was often ranked as important with logistic regression suggesting longer was better, perhaps giving readers more substance worth liking or commenting on. Another theme in feature importance was the sentiment or rating of the review. In particular, negative reviews, especially those contrary to popular opinion, seem to have an advantage likely because there are fewer reviews to compete with and because others who disagree with a majority opinion might feel more inclined to like a review mirroring their feelings. Finally, text representation slightly improved model accuracy but non-textual and textual characteristics still typically ranked higher in importance. Important words tended to be book or author specific, especially in BOW models, or broad literature terms such as "author" and "character". This last finding along with that regarding review length suggests that reviews going into detail about the author and characters, in other words going beyond a basic opinion, are more popular.

Aside from obtaining information related to a user's followers, future work would benefit from additional hyperparameter tuning such as trying L1 regularization for logistic regression, subsampling features for gradient boosting, or varying numbers of layers and nodes for a neural network. It would be beneficial to explore predicting a continuous measure of popularity such as likes share directly rather than arbitrarily creating a binary classification. Finally, given the importance of the user, it would be interesting to predict popular users instead of reviews and to take into account book metadata such as genre and length in understanding the types of books these popular users read and review.

The code for this project can be viewed at: `https://github.com/bridgetdaly/goodreads_ML`

References

[1] 30 proven ways to get free advanced reader copies (arcs). *BookSirens*.

[2] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[3] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[4] Y. Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

[5] C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.

[6] jj_ (https://stats.stackexchange.com/users/80688/jj). How to choose the number of hidden layers and nodes in a feedforward neural network? Cross Validated. URL:https://stats.stackexchange.com/q/180052 (version: 2018-05-31).

[7] K. Kousha, M. Thelwall, and M. Abdoli. Goodreads reviews to assess the wider impacts of books. *Journal of the Association for Information Science and Technology*, 68(8):2004–2016, 2017.

[8] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[9] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[10] L. T. K. Nguyen, H.-H. Chung, K. V. Tuliao, and T. M. Lin. Using xgboost and skip-gram model to predict online review popularity. *SAGE Open*, 10(4):2158244020983316, 2020.

[11] M. Wan and J. J. McAuley. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 86–94. ACM, 2018.

[12] M. Wan, R. Misra, N. Nakashole, and J. J. McAuley. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 2605–2610. Association for Computational Linguistics, 2019.

[13] K. Wang, X. Liu, and Y. Han. Exploring goodreads reviews for book impact assessment. *Journal of Informetrics*, 13(3):874–886, 2019.

[14] J. Wu. Review popularity and review helpfulness: A model for user review effectiveness. *Decision Support Systems*, 97:92–103, 2017.