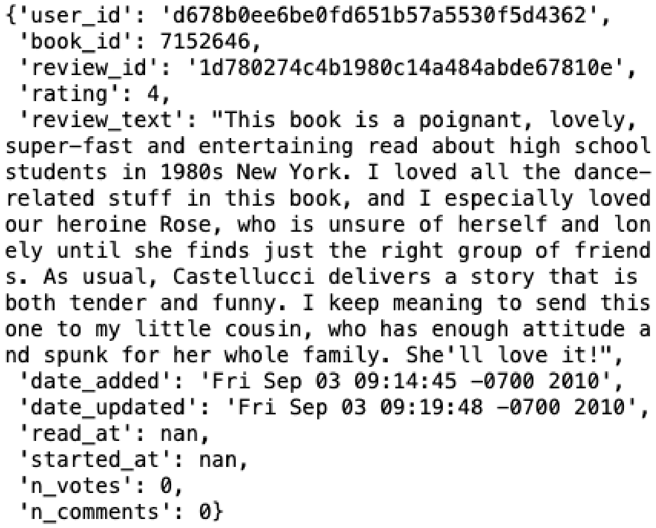
Identify and Explore Dataset

What makes a good book? When rating books, what factors most influence people's ratings? How can a model predict whether or not a given person is going to enjoy a given book? These are some of the questions we are seeking to answer. We began by identifying a suitable dataset to address the pressing questions we seek to pursue.

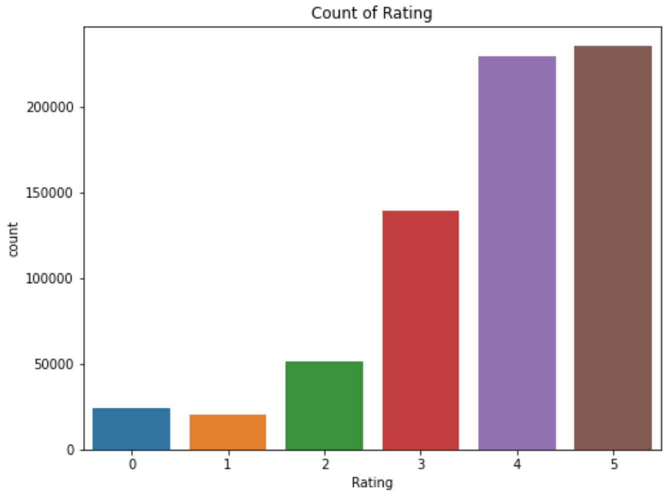
Every year millions of people read books and log their interactions on a social cataloging site called GoodReads. GoodReads allows users to review, rank, sort, and explore books, annotations, and quotes in a variety of languages. We took a dataset that is a subset of user provided reviews of young adult novels. These universal, often coming of age, literary works are read by a wide swath of the population all over the world. It is for this reason we decided to build our model from this subset.



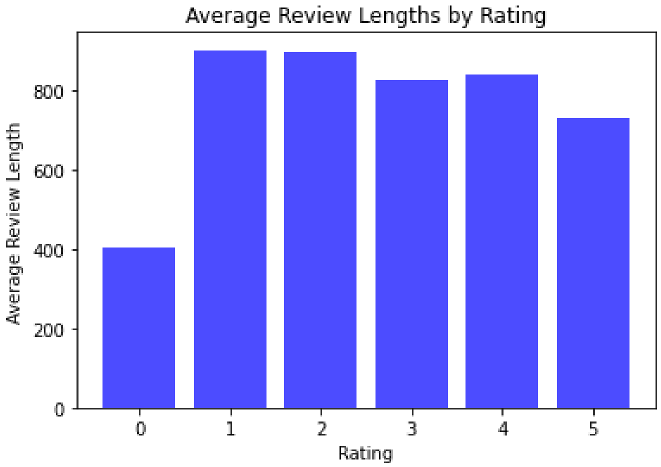
```
{'user_id': 'd678b0ee6be0fd651b57a5530f5d4362',
 'book_id': 7152646,
 'review_id': '1d780274c4b1980c14a484abde67810e',
 'rating': 4,
 'review_text': "This book is a poignant, lovely,
super-fast and entertaining read about high school
students in 1980s New York. I loved all the dance-
related stuff in this book, and I especially loved
our heroine Rose, who is unsure of herself and lon
ely until she finds just the right group of friend
s. As usual, Castellucci delivers a story that is
both tender and funny. I keep meaning to send this
one to my little cousin, who has enough attitude a
nd spunk for her whole family. She'll love it!",
 'date_added': 'Fri Sep 03 09:14:45 -0700 2010',
 'date_updated': 'Fri Sep 03 09:19:48 -0700 2010',
 'read_at': nan,
 'started_at': nan,
 'n_votes': 0,
 'n_comments': 0}
```

The full set from which we pulled consisted of 2,389,900 detailed reviews from book lovers around the world, in more than 10 different languages. For model simplicity, we chose to focus on 700,000 of these reviews. Each review consisted of a user_id, book_id, review_id, rating, review_test, date_added, date_updated, read_at date, started_at date, n_votes, and n_comments. As we began to work with the data, there were a few simple statistics we wanted to discover.

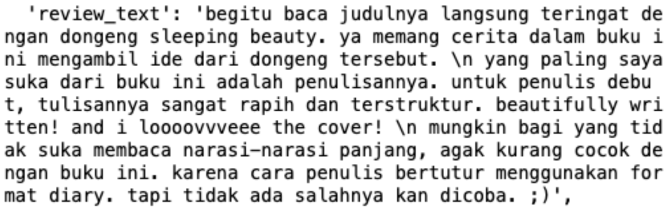
We began by splitting our 700,000 review dataset into subsections of 80% train and 20% test data. We now have 560,000 reviews to train a model on, so as to predict what rating each of the 140,000 user and book ids in the test set would give. Where to begin?



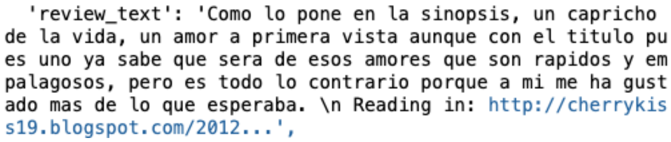
Our first step was to see whether or not we were dealing with a balanced dataset. Each book is given a rating 0 to 5, integers only. Was each rating equally represented? Or were the reviewers overly critical/positive? It turned out the users were more likely to provide positive reviews than negative ones. This could be because the books were incredibly well written, or because the users were mostly selecting books they knew they would like. Regardless, we can do some more digging on this idea and it is helpful to know moving forward for our future model that 4 and 5 star ratings were overrepresented in our data, representing about 60% of all reviews.

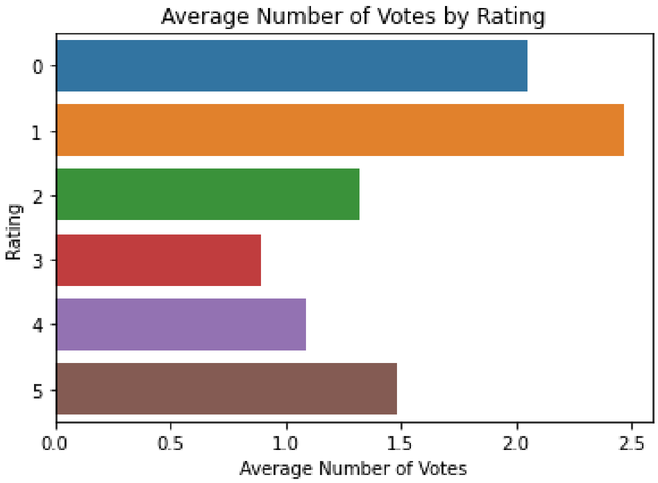


Our next instinct was that people potentially wrote longer reviews for the books they really liked. This could be represented by the average review length in characters being longer for all the 5 star reviews as compared to the 1 or 2 star reviews. This ended up not being the case, as the 1 and 2 star ratings had the longest review lengths. This may imply that people were more likely to write longer reviews for the books they didn't enjoy rather than for the ones they did. The difference wasn't too significant however so this isn't a factor we will likely utilize in our future model.

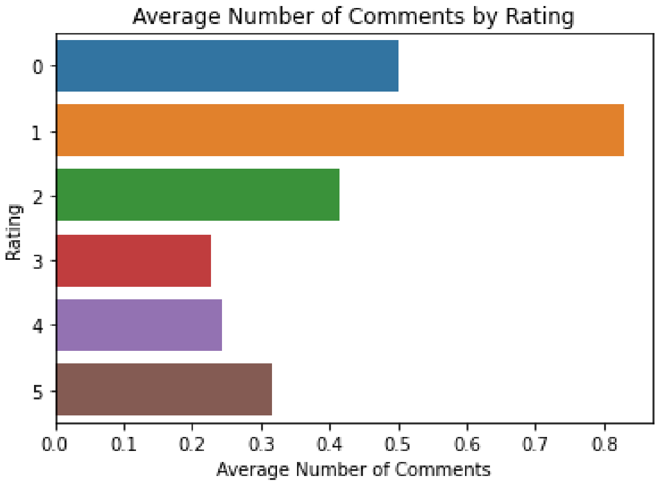

```
  'review_text': 'begitu baca judulnya langsung teringat de
 ngan dongeng sleeping beauty. ya memang cerita dalam buku i
 ni mengambil ide dari dongeng tersebut. \n yang paling saya
 suka dari buku ini adalah penulisannya. untuk penulis debu
 t, tulisannya sangat rapih dan terstruktur. beautifully wri
 tten! and i loooovvveee the cover! \n mungkin bagi yang tid
 ak suka membaca narasi-narasi panjang, agak kurang cocok de
 ngan buku ini. karena cara penulis bertutur menggunakan for
 mat diary. tapi tidak ada salahnya kan dicoba. ;)',
```

One idea we will make sure to keep in mind in any Natural Language Processing we pursue for our model is that there are many different languages included in this dataset. People from all over the world love reading and reviewing Young Adult novels. If we are going to perform sentiment analysis or other NLP techniques, we will need to include packages for the variety of languages included in our diverse dataset.

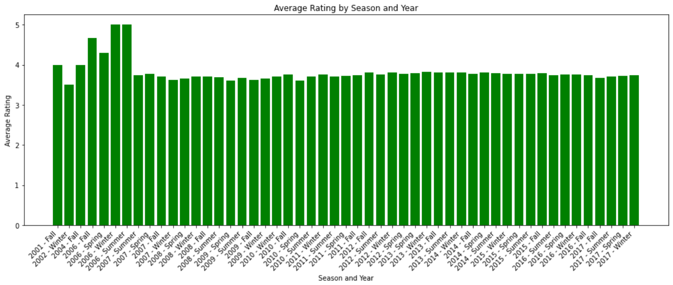

```
  'review_text': 'Como lo pone en la sinopsis, un capricho
 de la vida, un amor a primera vista aunque con el titulo pu
 es uno ya sabe que sera de esos amores que son rapidos y em
 palagosos, pero es todo lo contrario porque a mi me ha gust
 ado mas de lo que esperaba. \n Reading in: http://cherrykis
 s19.blogspot.com/2012...',
```



Our data set also had a category called "n_votes" that represented the number of times users voted on the book. We wondered whether perhaps the number of votes that each book received possibly impacted its overall rating. As shown in the chart above, lower rated books tended to receive more votes. Maybe this, similar to the longer average review lengths, was because users tend to interact more with books they enjoy less. A bit of a glass half empty look at book reviews, but let's see if we notice that same trend in the number of comments (n_comments) left on the review.



Again, here we see the lower rated books, receiving more comments on average, than the higher rated books. Likely this is because other users are chiming in to express their agreed distaste of the book, or it could be people coming to the defense of the book. Either way, this is a trend we are noticing and can possibly exploit in our model we will use later on.



We were able to get started on this project well in advance and spend a lot of time in the exploratory phase. We investigated many different aspects of our data, most of which we did not end up incorporating in our model, but still led us to some insightful takeaways, and meaningful learning. One of which was time series data. We broke down the data by minute, hour, day, month, season and year. For our data, the season seemed to yield the most telling results. We thought maybe the time of year, specifically the season in which a user was performing a review on a book, may be an indicator we could represent in our model. It turned out to not be as helpful as we originally hoped, but a valuable analysis nonetheless.

Overall, we feel confident that through scrupulous searching, we have found a rich dataset that is ripe for a predictive task. We are excited to take the next step in identifying a predictive task that we can build a model around for GoodReads.

Predictive Task

What would be valuable to GoodReads is a system that is able to understand each individual user and book, and pair them together in a way that maximizes user experience. The predictive task that we are pursuing does just that. We will design a model that, given a user_id and book_id combination, predicts the exact score that would be given by that user for that book. This is an invaluable tool to a website like GoodReads as, when done well, they can let a user know what books they might like to read next and when a user is considering their next read, tell the user what rating they will likely give it.

Evaluation of our model will be pretty straightforward. We have split our 700,000 complete reviews into 80% train and 20% test sets. We will train the model on the training data, and then evaluate it on only the user_id and book_ids from the test data. We know what ratings these users actually gave those books. Each rating is an integer 0-5, so we will predict one of these ratings for each user, book combination, and then compute the mean squared error from the actual ratings.

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Our exploratory analysis clued us into a few important features that we will be using to inform our model. It was clear from our initial examination of the dataset that each user and book had tendencies. Some books were consistently rated below average, some users frequently were overly positive in their ratings, giving ten 5 star reviews, for every lower review they provided. It became obvious to us that we could leverage the trends seen by individual users and books, bringing a collaborative filtering approach to the dataset.

We processed the data by developing dictionaries that corresponded to the ratings given by each user and to each book. This gave us a sense of how their ratings compared to the average ratings seen generally throughout the data set. We were then able to assign a bias to each user and book, based on the rating history for each. These ended up being the most important features for our future model.

In order to establish a relevant baseline for our task we implemented a linear regression model. This model used the features developed above to assign a beta coefficient to each individual user and book as well as a global alpha coefficient. We then were able to get our baseline prediction by adding the alpha value to each book's bias to the bias for the given user. This is a simple and novice baseline, that our future model will build and improve upon.

Our model's predictions will be valid as we know the true ratings of the users and books in our test set. Once we have trained our model on all 560,000 train reviews, and evaluated its performance on the

140,000 test reviews, we will be confident in the model’s ability to perform accurately on any new user, book pairs GoodReads has in the future.

In summary, we will be predicting, given any user and book combination, what rating will occur from that interaction. This will bring immense value to GoodReads, and increase user reading experience overall.

Model Description

We ended up using a 3 model ensemble to get our final prediction. The models were linear regression (baseline), Singular Value Decomposition (SVD), and Jaccard similarity.

Linear Regression:

We used linear regression with a squared regularization coefficient. This is also known as ridge regression. This of the form:

$$rating \ = \alpha + \ \beta_{user} * x_{user} + \beta_{book} * x_{book} \qquad (1)$$

Every user and book has their own Beta coefficient which is a numerical value for that user or item's bias. So when given a book review, x, which has a user_id and book_id we find that review’s rating prediction by adding the user bias to the book’s bias. We found the beta values by solving the below:

$$argmin_{\alpha,\beta} \Sigma_{u,b}(\alpha + \beta_u + \beta_b - R_{u,b})^2 + \ \lambda(\Sigma_u B_u^{\ 2} + \Sigma_b B_b^{\ 2}) \quad (2)$$

This is calculated using a modified version of gradient descent. We iteratively take the gradient for both beta values and alpha and set that gradient to 0. Eventually this converges to the best values for beta_u, beta_b, and alpha.

The right hand side of equation (2) is the regularization coefficient. This controls the tradeoff of training accuracy vs how the model generalizes to prevent overfitting. With a lower lambda value, the model might overfit while a higher lambda value may lead to not as accurate predictions. We set the lambda coefficient to 1 for the baseline for the sake of simplicity.

This model does a good job at capturing basics about if a book is good and if a user tends to rate highly. The weakness of this model is that it will not capture complex relationships between the two variables as it assumes there is no interaction between the user and the book. It does not say anything about how a user might rate specific types of books.

Singular Value Decomposition:

SVD can be viewed as an improved model on linear regression. Like linear regression, it will have beta_u and beta_b, which are the user and book biases respectively. It also has gamma terms which tell us the user’s preferences and book’s properties. Updated equation below:

$$rating \ = \ \alpha \ + \ \beta_u + \beta_b + \gamma_u * \gamma_i \qquad (3)$$

The interaction terms between the users and books are not in the data so they are thought of as hidden. They can represent things such as a book’s genre and if a user likes that genre. To implement SVD we used SVDpp from the surprise library. This proved to be a challenge as the model has many parameters to tune and training the model takes a lot of time. We ended up making a grid of different values of parameters we wanted to tune. These parameters were n_factors, lr_all, reg_all, n_epochs. N factors is the number of hidden variables, lr_all is the learning rate for all variables, reg_all is the regularization coefficient for every variable, n_epochs is the number of iterations of gradient descent that are performed to find the parameters. We then used grid search which tries all combinations of these parameters and returns which combination performs best on the train data in terms of mse. The combination of parameters which minimized the train data turned out to be n_factors= 2, lr_all=0.01, reg_all = .1, n_epochs= 20. We think there is a more optimal combination of parameters but gridsearch runs very slowly so it was hard to test more parameter combinations. By itself, SVD’s predicted values gave an MSE of 1.253 which is a nice improvement over the baseline. SVD proved to be a strong predictor.

Jaccard similarity:

For a given rating, j, prediction using Jaccard similarity looks at all of the other ratings a user has given and computes a weighted average of those ratings with the weights being determined by the Jaccard similarity between the given book and the other books the user has rated. Shown in equation (4):

$$1/Z \ * \sum_{j \in I_u \setminus \{b\}} Jaccard(b, j)R(u, j) \qquad (4)$$

Jaccard similarity is a similarity function that looks at all of the users that read the book, j and compares them to all of the other users who read other books the user, j has read. It is defined below, equation (5):

$$Jaccard(B_i B_j) \ = \frac{|B_i \cap B_j|}{|B_i \cup B_j|} \qquad (5)$$

Jaccard is simple but it is not great with sparse data. This means it will not be very accurate for the books with few ratings because those few ratings will have a huge impact on the final prediction. By itself, Jaccard similarity gave an MSE of 1.525 which is much lower than even our baseline. This suggests that Jaccard is not the strongest estimator of rating. Nonetheless, including it helped our final model when it is ensemble with the other models.

Ensembling:

All three of the models discussed produced predictions for our test set. To create a final prediction we made an ensemble of the models. We did this using a weighted average where the weight applied to each model's predictions was a hyperparameter we tuned. Our best model gave the weights .7, .1, .2 to SVD, Jaccard, regression respectively. This shows that SVD dominates the predictions while the other models are not quite as effective. **This gives us a final mse of 1.248 which is an.improvement over the baseline which was 1.279.**

Areas for improvement:

As mentioned in the exploration step, there were other features in the review that we thought could have been useful in making predictions. Examples of these features are the review text, n_votes, and n_comments. We were not able to figure out how to use these features given that we wanted to make a system that would predict a users rating given only the user_id and book_id. We think we could have implemented these features in regression but exactly how to do this eluded us.

Literature

As mentioned in the "Identify and Explore Dataset" section, the dataset used in this exercise was pulled from Goodreads. This platform serves as a social cataloging website that allows book readers to rate and review books and share their experience with others on the platform. This dataset is not a direct input into the Goodreads application; rather, it's an openly accessible output from the user community, providing a unique opportunity for comprehensive data exploration.

The dataset structure consisting of user and item profiles, several interaction-related features, temporal information, and reviews, all accompanied by user ratings, is fairly common in the data science space. Some of the predictive tasks that could be applied to this data, along with others similar to it, include whether a user would interact with an item, what a given user would rate an item, a user's sentiment towards an item, and many others. For this dataset, we chose to analyze what a user may rate a given item, regardless of whether there is a record of that user interacting with the item in the past.

There's a wide array of models and approaches to analyzing this type of data, and seemingly countless exploration tasks that individuals can run on these datasets. Ultimately, recommendation systems aim to put an item in front of a user at the right time, and that's what these models do:

Deep learning models like Convolution Neural Networks can unpack user item interaction tendencies and can find some of the underlying patterns in data that the human eye cannot see. Collaborative filtering is based on the idea that if a user has the same opinion on a topic as another user, they likely will have the same thoughts on other topics. Matrix factorization models like Singular Value Decomposition, which plays a pivotal part in collaborative filtering, serve to find the strength of linear relationships between columns and rows in a matrix. While in this case, SVD was used for the purpose of recommendation systems, it's also a powerful tool in the data compression and image processing space. Embedding methods like word2vec can find text-based information that contributes to user's preference for an item. Graph Neural Networks can model user-item interactions in a graph format. These complex networks can capture datetime information, sequence-based tendencies, user or item metadata, and other information provided. This makes it a very strong tool in the machine learning space. These are just a few of the well-studied analysis tools on review/interaction datasets.

The approach used in this exercise was an ensemble of SVD, Jaccard Similarity, and a latent-factor model (linear regression). The approach to modeling and application of the dataset was similar in this exercise to outside work. It's worth noting that our conclusions regarding specific ratings may differ slightly from prior studies on book data, given the unique subset we worked with, distinct from datasets employed in earlier research. Here are some examples of past studies.

From a study posted on Medium [1], entitled "Recommending Goodreads Books using Data Mining", a similar structure in data was examined that leaned towards higher book ratings. While working with a slightly different dataset, the aim of this model was to predict the average rating of new books, across the entire base of users reviewing them, and a linear regression model was fitted on the data to come up with a generalization for the book ratings. The MSE observed was slightly better, as the study was looking at market rating interaction versus user rating interaction.

Another study, posted on Kaggle [2], with a title of "Predict Book Rating with Linear Regression" looked to predict ratings as well. While the dataset was different than the one used in this study, a similar approach was taken to data analysis. A linear regression model was used and trained on most of the fields either numerical or encoded by a label encoder. The model performed decently on the test set but had no hyperparameter tuning. Our model was more functional in the sense that it's able to predict a user's rating on a given book just given those two features as opposed to needing the entire dataset. This is more typical of a fully collaborative filtering model.

Results and Conclusions

The task we took on was to predict how a user would rate a book they had not read before. To make this as real world as possible, we assumed that we should only be given a user_id and book_id to test on. This is because if a user has not read a book there will be no information about the review text or other features related to the review.

During feature exploration, we found that certain features in the review looked like they would be highly predictive of the rating. For example, a high number of both votes and comments for a review indicated that the review would be low. We found this difficult to bring into the models because it is information we would not have in testing. A great way to improve the models performance in the future would be to figure out how to include these highly predictive features. We also found that there is a significant class imbalance. There are much more 3, 4, and 5 star reviews than the lower review scores. We think downstream this might affect the models created. It could lead to our model overpredicting the high ratings because doing so still lowers the MSE. To handle this better, we could have undersampled the data. This is where we sample less of the majority classes to even out the distribution but this would lead to us losing information. We could also have used models that are more resistant to class imbalances such as Random Forests.

The MSE found using linear regression was set as the baseline to beat. This model assumes no interaction between the user and book. This is unrealistic as a user will have preferences for certain books and authors and each book can be defined by it's genre and target demographic. This MSE this gave was 1.279.

Ultimately, we went with a three model ensemble to predict the rating a user would give a book. Those models were regression (the baseline), SVD, and Jaccard similarity. SVD has parameters for the user bias, the book bias, and interaction terms. The interaction terms specify if a user would like a specific book which is an important improvement over linear regression. Jaccard predicts the book's rating based on all of the other ratings a user has given and computes a weighted average of those ratings using Jaccard similarity to determine the weights. The ensemble of these three models produced a MSE of 1.248 which beat the 1.279 the baseline gave. The optimal blend of the weights of each model's prediction was .7 for SVD, .1 for Jaccard, and .2 for linear regression. This shows that SVD dominated our predictions while still being assisted by the other models.

The results of our modeling prove that we can get pretty close to predicting how a user would rate a given book. This could be useful when Goodreads recommends books to a user. Even if we cannot give an exact rating prediction, we are in the ballpark enough to say if a user will enjoy a book. This can drive user engagement on the website, promote books that have paid for advertising, and lead many people in the world to enjoy jumping into a new book they didn't know they would love.

References

[1] Guna, Karthic. “Recommending Goodreads Books Using Data Mining.” *Medium*, Analytics Vidhya, 3 Oct. 2019, medium.com/analytics-vidhya/do-you-love-reading-lets-use-data-mining-and-find-some-good-reads-for-you-e5bf1b576316.

[2] “Predict Book Rating with Linear Regression.” *Kaggle*, Kaggle, 29 Aug. 2019, www.kaggle.com/code/data13/predict-book-rating-with-linear-regression.