Alexandra CURRAN, Mathilde ROMAN, Amiirah CODABACCUS, Srimalika POTLURI, Selvalakshmi RAMAGOPALAN

Machine Learning Goodreads Project Report

Data Analysis

In our initial trial run, we started out the data analysis process by using the pandas **describe()** function as our springboard. From there, we had a few intuitions that were confirmed later in the project:

Ratings Count	We assumed there would be a normal distribution when comparing with average ratings. We considered that the higher the count of ratings, the higher the ratings, but at some point, it achieves an equilibrium of around 3-4 in average rating.
Text Reviews Count	Same thing as the ratings count.
Publication Date	We did not think the publication date would have an impact on the average rating. Especially considering that most of the dataset is comprised of contemporary books.
Average Ratings	We expected a normal distribution of values. We assumed two points: that most books are considered average, and that excellent or poor books constitute a small portion of overall books. Additionally, people typically tend to rate an average value, as opposed to the minimum or the maximum.

There are some intuitions that were proven wrong:

Authors	We thought that the popularity of an author would influence the average ratings of their books. This was disproven through our EDA. In reality, popular authors represent a small percentage of overall authors. Many non-popular authors still have a good overall average rating.
Number of Pages	We did not think the number of pages would have much of an impact on the average rating. This was disproven through our EDA. The median number of pages seems to be higher for books rated 3 to 4. It is likely the length of a book impacts the story's pacing.
Series	Some titles included additional information on the series name and the volume number. We theorized that certain volumes (first and last) and certain series could have a link to a higher average rating. This was disproven through our EDA. After feature testing, we realized that the volume and series universe had little impact on the rating.

There was a big issue with the dataset as well: only 25% of the books had a rating below 3.77. Our dataset was skewed, and so our model would be biased for more well-rated books. We naively assumed this would be an easy fix as many book websites exist, meaning we should in theory be able to retrieve book metadata. This part of the process ended up taking up a large portion of our project time.

After much research and testing of different APIs, we settled on using a Goodreads data dump from 2017 to append additional data points. This data dump was scraped from Goodreads users’ public shelves. We ended up with about 38,000 lines, as opposed to the 11,000 we started with. While it added additional data points for lower rated books, it was still lower than the books rated 3 or 4.

In the first couple of runs on the original dataset, we tried both extremes of feature engineering: keeping many features that were even slightly relevant (encoding the non-numerical ones) and reducing the features to the minimum. The outcome of these trial runs was that a model performed better with features that seemingly had a relation to the average rating. The issue at this point was that there did not seem to be a feature that was solidly correlated to the average rating. This led us to create additional columns by computing ratios. These ratios included: title occurrences, author occurrences, publisher occurrences, publisher popularity (average ratings counts/average text reviews), author popularity (average ratings counts/average text reviews) and a ratings count/text reviews count ratio. This last ratio is the one we discovered had a good correlation to the average rating, confirming our initial intuition that the ratings and reviews count had a part to play in predicting the average rating of a book.

Plots & Graphs

For every feature, we generally plotted out outliers as well as at least one graph to visualize the relation to the average rating of books. From here, we were able to tailor down the features we considered had little impact on the target column. Much of what we first observed in the beginning of the data analysis was confirmed visually in this section. Notably, the correlation matrix validated the features we were considering i.e., number of pages, ratings count, text review count, and ratings/review count ratio.

- Number of pages: it seems that the higher the number of pages the higher the average rating until around a rating of 4.
- Ratings count & text review count: it seems that the higher the rating, the higher the count of ratings and text reviews
- Ratings/reviews count ratio: showcases another aspect of what we observed above but making the relation more apparent.

Features Selection

There were three different reasons that led us to excluding certain features:

Unique identifiers	bookID, isbn, isbn13	These were features that only served to identify a given book and had no influence on the average rating.
Low-impact features	authors, first author, language_code, publication_date, publisher, format, series exists	These were features that we deemed had little correlation to the average rating through our EDA process.
Unneeded Ratios	title_occ, author_occ, publisher_occ	These were features that we created from calculating with initial features. They were included in some of our initial modelling runs, with little impact on the performance.

Model Selection

For our models, we chose some popular regression models:

- Linear Regression
- Polynomial Regression
- Ada Boost & Decision Tree
- Random Forest Regression

We were pleasantly surprised by how well the Random Forest model performed. We think it outperformed all other models mainly due to its robustness to outliers, its solid ensemble learning, and being less prone to overfitting

Model Results & Interpretation

Here are the scores of our Random Forest model:

MSE	RMSE	R2	MAE
0.03	0.19	0.81	0.11

The RMSE score indicated that on average the model's predictions were 0.188 units away from the correct average rating. It measures prediction errors. The R2 score of 0.81 indicated that our model can explain 81% of the variance in the average ratings. It means that with our selected features, the model can explain the different average ratings satisfyingly well. The MAE score indicated that the absolute value differences were on average 0.11 units away from the correct average rating.

In other words, our model performs quite well. There are relatively low errors in the predicted values (RMSE) and it can capture and understand approximatively 81% of what's happening in the data (R2).

How Our Model Can Be Improved

We were happy to achieve a model that fits well with the data and can predict the average book ratings with minimal errors. A model can of course always be improved. Here are some additional ideas we had:

Adding more features	This would give us more data on the books and perhaps a better understanding of what other aspects play a part in a book's rating e.g. genre, volumes, contributors and their roles, etc.
Growing the dataset	Specifically for two points: - For lower-rated books: we are still not satisfied with the amount of data points - For older books: it would allow us to test if there truly is not a correlation with the average rating
Testing with additional models	Both regression and classification models to see if there is a better performing model than the Random Forest one.

Improved outlier processing	We think that our outliers tailoring could be further improved and as such, boost the model’s performance.
Evaluate book popularity outside of average rating	Implementing different ways of measuring a book’s popularity outside of counting reviews and ratings e.g. if a book has a movie produced from it, does it tend to have a higher rating?
Parse text reviews	It would allow us to gain a better understanding of the positive and negative aspects of the book for people