Date: January 31st, 2024 **Authors:** Lydia Meftah, Andrew Wieber, Loïc Martins, Phuc Nguyen, Evan Kim	**Class:** Machine Learning with Python Labs **Project:** Book Rating Prediction Model

BOOK RATING PREDICTION MODEL REPORT

I. Introduction

Goodreads is the world's largest site for readers and book recommendations. The ability to predict the rating of a book before it goes on sale can allow Goodreads to use these predictions to highlight specific books. Being able to make these predictions with good accuracy can become an asset for the company.

To achieve this, we had a dataset composed of more than 11,000 rows. One line equals one unique book ID, hence one book, or at least one book "edition". We also had 12 input variables (columns): 5 categorical variables and 6 numerical variables. And of course, we have one target variable, "average_rating".

II. Methods

To begin with, regarding the type of target variable, we can conclude that it is a regression problem because we need to predict precise values (3.8, 4.3...). But we need to explore the data before choosing our model.

As we explored the dataset, we noticed three important points. First, we had different variables that were not useful for our model: different ID variables and all categorical variables with a name. Second, we had a significant range of values for the different variables, with perhaps a few outliers. Third, no variable has a true Gaussian normal distribution. Finally, using the correlation matrix, we concluded that we did not have a real relationship between our input variables and our target variable (see the first 5 variables in Chart 2, which is presented later).

The dataset structure before cleaning is:

- The dataset consists of 11,123 entries. There are 12 columns.
- Data Types:
  - The dataset includes a mix of numeric (int64, float64) and object (string) data types.
- Sample Data:
  - The first few entries include popular titles like "Harry Potter and the Half-Blood Prince" with various attributes like ratings, number of pages, and publication details.
- Statistical Summary:
  - average_rating: The average rating varies from 0 to 5, with a mean of around 3.93, suggesting a central tendency towards higher ratings.
  - num_pages: Book lengths vary widely, with an average of 336 pages.
  - ratings_count and text_reviews_count: There's a significant range in the number of ratings and text reviews, indicating a varied popularity among the books.

Before working with a model, we carried out an important cleaning part: check for duplicate and missing values but most importantly check variable types and possibilities in specific variables. First, we identified the same publishers with different spellings and forms. So, we created a function to clean up the variable to have the "root" of the publisher's name. This function made text lowercase

and removed all spaces, accents, special characters, and noisy words. This allowed us to group the same publishers together. The author's column was already clean, and we didn't touch the “title” column (just removed lower case, removed accents, special characters, and extra spaces). Second, we separated the authors column to have” first author” and “other participants”. Third, we purged books that had too few reviews (less than 7), about 5% of total books, because they were not relevant to our dataset. For example, a book with 2 ratings indicates no real information about the quality of the book.

After the cleaning part, we spent a lot of time on the feature engineering part because we didn't have enough relevant features. We added the following features:

- books_by_main_author -> showing how many books have been published by the author.
- books_by_publisher -> showing how many books have been published by the publisher.
- length_title -> showing length of the book title.
- nb_authors -> showing the number of authors/participants.
- avg_author_rating -> showing the average rating for the authors.
- avg_publisher_rating -> showing the average rating for the publishers.
- language_code -> inside the variable we replaced language containing “eng” (eng US...) by “eng” and all other languages were grouped together. English becomes 1, others become 0.
- year, month, and quarter -> taking the information contained in publication_date to see if this level of precision was important.

We used different plots and a correlation matrix for our exploratory data analysis. We concluded that we didn’t have, again, Gaussian normal distribution. The distribution of numerical data is described in Chart 1.

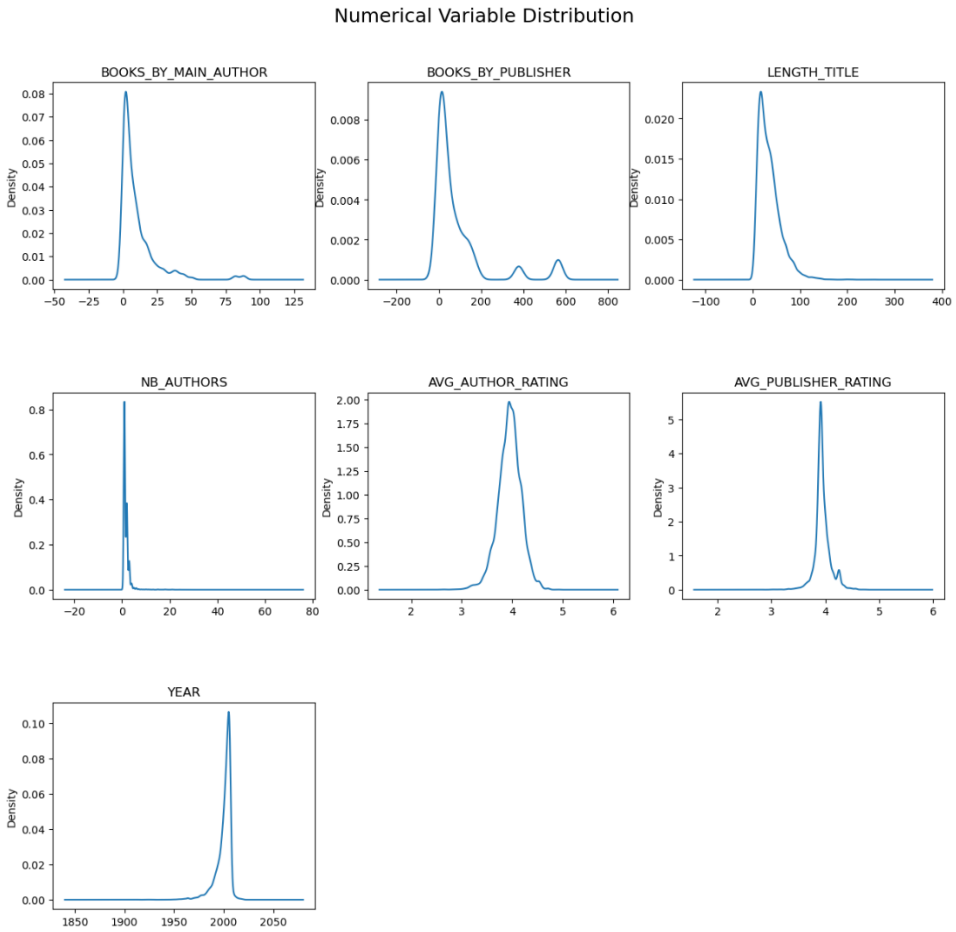

Chart 1: The distribution of some numerical attributes after the cleaning part

The correlation matrix (chart 2) indicates that several attributes show little to no linear correlation with each other, as evidenced by correlation coefficients close to 0. For instance, the month and quarter of publication (month, quarter) show no substantial correlation with the average rating of the book (average_rating), indicating that the time of publication does not linearly affect the book's received rating.

Notable exceptions to this pattern include the strong correlation between average book rating and average author rating (0.83), and a moderate correlation with average publisher rating (0.53). These relationships are outliers within the matrix and signify that the reputation of authors and publishers may have a more pronounced impact on how books are rated.

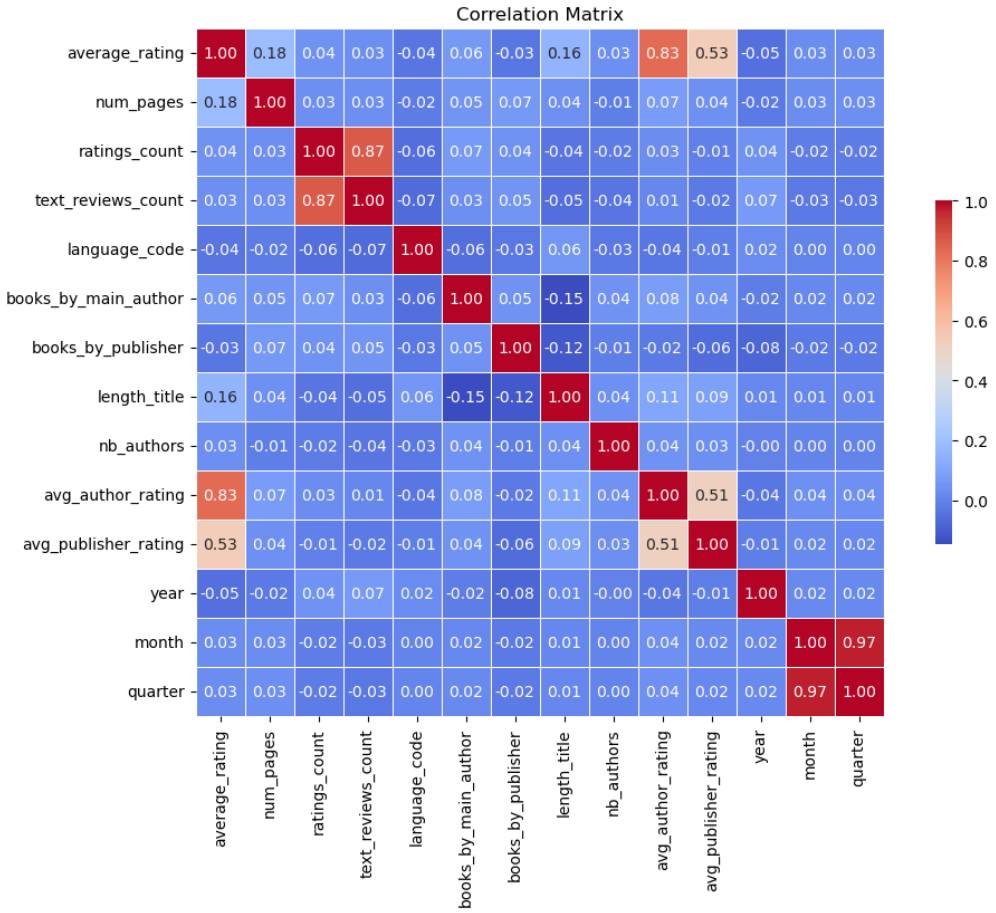

Chart 2: The correlation coefficients of attributes after the cleaning part

Through feature engineering, without scraping more data, we were able to add relevant data from our main variables and conclude that we now had variables with a good correlation to the target. At this step, we decided to not delete outliers because it was relevant data, and we didn't have enough rows in our dataset.

Using this conclusion, we decided to try two regression models:

1. Polynomial Regression of degree 2: because this model can detect more complex relationship between inputs and target variables, in comparison to a basic linear regression.
2. Random Forest Regression: this model is more complex; using the average of decision trees can help us to have a better accuracy in our predictions.

Before running the model, we finished with a transformation part to create a dataset readable by the model. First, we dropped irrelevant features (title, authors, isbn, isbn13, text_reviews_count, publisher, first_author, publication_date, quarter, z_scores, num_pages) because they were not useful or because they had relations with each other. We then transformed the ratings_count to a gaussian distribution and then standardized. After that, not having a Gaussian distribution for most

variables, we used data normalization, except the language_code and ratings_count. These variables are binary, so normalization is not necessary.

III. Results

The full list of features used in our models includes the avg_author_rating and avg_publisher_rating. When the author and publisher are already present in the data, then the full model can be used. However, when one or both is new to the model, the corresponding feature must be removed. Finally, we ran our two models, and we were able to conclude that our Random Forest Regression performed slightly better (see Tables 1a/1b).

Table 1a: Polynomial Regression results (base features):

	MSE	RMSE	MAE	R2-score	Adj-R2_score	Target min value	Target max value
Cleaned data	-	-	-	-	-	2.40	4.91
Final features	0.020	0.143	0.0992	0.737	0.737	2.65	4.88
New author	0.051	0.225	0.1713	0.347	0.346	3.19	4.74
New publisher	0.022	0.148	0.1005	0.719	0.719	2.66	4.92
New author & publisher	0.071	0.267	0.2087	0.083	0.083	3.69	4.60
Features used	num_pages / books_by_main_author / length_title / nb_authors / avg_author_rating / avg_publisher_rating						

Table 1b: Random Forest Regression results (base features):

	MSE	RMSE	MAE	R2-score	Adj-R2_score	Target min value	Target max value
Cleaned data	-	-	-	-	-	2.40	4.91
Final features	0.019	0.138	0.0919	0.754	0.754	2.68	4.75
New author	0.052	0.228	0.1701	0.333	0.332	2.98	4.74
New publisher	0.021	0.145	0.0949	0.731	0.730	2.66	4.77
New author & publisher	0.072	0.269	0.2054	0.071	0.071	3.33	4.58
Features used	num_pages / books_by_main_author / length_title / nb_authors / avg_author_rating / avg_publisher_rating						

Before concluding we decided to scrape extra data and retry our models. We scraped book format and book genres. Using this data, we concluded that the model was performing better (see Tables 2a/2b).

Table 2a: Polynomial Regression results (scraped features included):

	MSE	RMSE	MAE	R2-score	Adj-R2_score	Target min value	Target max value
Cleaned data	-	-	-	-	-	2.40	4.91
Final features	0.020	0.143	0.0981	0.738	0.738	2.66	4.88
New author	0.050	0.224	0.1705	0.357	0.356	3.14	4.81
New publisher	0.022	0.147	0.1004	0.722	0.721	2.65	4.87
New author & publisher	0.069	0.263	0.2052	0.108	0.107	3.72	4.61
Features used	num_pages / books_by_main_author / length_title / nb_authors / avg_author_rating / avg_publisher_rating / audio / hardcover / other_format / paperback / fiction / nonfiction						

Table 2b: Random Forest Regression results (scaped features included):

	MSE	RMSE	MAE	R2-score	Adj-R2_score	Target min value	Target max value
Cleaned data	-	-	-	-	-	2.40	4.91
Final features	0.018	0.136	0.0893	0.764	0.761	2.69	4.75
New author	0.044	0.210	0.1537	0.434	0.428	2.91	4.76
New publisher	0.019	0.139	0.0894	0.753	0.750	2.66	4.76
New author & publisher	0.052	0.229	0.1690	0.325	0.318	3.44	4.59
Features used	num_pages / books_by_main_author / length_title / nb_authors / avg_author_rating / avg_publisher_rating / audio / hardcover / other_format / paperback / fiction / nonfiction / classics / fantasy / literature / historical fiction / history / mystery / novels / romance / childrens / philosophy / science fiction / young adult / contemporary / historical / biography / thriller / humor / adventure / short stories / crime / audiobook / science fiction fantasy / horror / mystery thriller / literary fiction / memoir / american / politics / suspense / reference / religion / comics / poetry / graphic novels / middle grade / school / science / psychology / british literature / adult / essays / chick lit / war / paranormal / drama / self help / plays / france / picture books / manga / biography memoir / art / spirituality / 20th century / magic / comedy / travel / high fantasy / anthologies / animals / mythology / business / detective / christian / 19th century / adult fiction / american history / autobiography / theatre / epic fantasy / contemporary romance / sociology / magical realism / christianity / military fiction / vampires / action / urban fantasy / realistic fiction / graphic novels comics						

The polynomial regression model was unable to use all the genre features. As a result, the random forest regression model is superior (except for runtime, which is slightly longer). In particular, it performs much better when the author and publisher are new.

IV. Examination of Model Predictions

Both Random Forest Regression and Polynomial Regression models perform very well with the full feature set. Random Forest is slightly better and has the advantage of being capable of handling all the scraped data. The inclusion of additional features such as book format and detailed genre classifications likely contributed to this slightly enhanced performance, allowing for a more nuanced understanding of the books. Both models can handle complex, non-linear relationships. Random Forest is robust against overfitting, which may also contribute to its better performance. Since Polynomial Regression cannot incorporate all the scraped data, it is the least effective when authors and publishers are new. This makes Random Forest Regression the preferred model.

Examining more closely the results. We see that books that are the sole publication of an author have excellent predictions, with the mean residual and standard deviation of residuals very close to zero (see Table 3). This is probably due to the rating of the book being already available in the the avg_author_rating feature since the average rating for 1 item is simply a copy of that. These books should be predicted without using the avg_author_rating feature.

Number of Published Books	Number of Samples	Mean of Residuals	Standard Deviation in Residuals
1	522	0.0016	0.033
2	209	0.0068	0.15
3	134	0.0036	0.124
4	123	0.0379	0.176
5	116	-0.0045	0.198
6	79	0.0038	0.147
All possibilities	2126	0.0045	0.136

Table 3: Residuals and their standard deviation compared to number of books published by the author

Looking at specific examples of good and bad performers, we notice the following:

- Good predictions:
    - The average author rating is very close to the rating of the book. Considering the high correlation between the two parameters, this is likely the reason the prediction was excellent.
    - Examples:
        - The Complete Greek Tragedies Volume 1: Aeschylus by *Aeschylus* – bookID 1527
        - Alice in Wonderland by *Lewis Carroll* – bookID 13023
- Bad predictions:
    - The real book rating is very different from the average author rating. This is the probable reason for the bad prediction, since the model learned that the two parameters are supposed to be highly correlated.
    - Examples:
        - Trimalchio by *F. Scott Fitzgerald* - bookID 4723
            - Underrated by 0.6 points
        - The Younger Gods (The Dreamers #4) by *David Eddings* – bookID 18880
            - Overrated by 0.8 points

A post-modelization analysis shows that the Random Forest model is not excellent for ratings below 3.5, probably because there are few samples of this ranking, which makes it harder to predict. The model can predict particularly well between 3.6 to 4.4 since the RMSE is below 0.02.

V. Conclusion

In the end, we can conclude that the models have good performance only when the author is already known. An unknown publisher does not significantly affect the results. We can also ask ourselves if it was relevant to predict a precise value like 4.65 for Goodreads. Depending on the demand and the business problem, it may be interesting to transform our model into a classification problem. We can transform our target variable into 3 categories: low rating, average rating, high rating. Maybe this information is enough to solve our business problem and we can achieve better accuracy in our predictions.

We could envisage applying a SMOTE to the data, in particular for an average_rating < 3.5. A quick first look showed that an R2-score of 82% is possible when tripling the number of samples in this subgroup, but that the error increased for other average_scores.

As a final remark, we are obliged to wonder... why is it that a website dedicated to books shows a rating correlation of - 0.15 for novels? Yes, a negative correlation. The answer clearly lies in the details.