Lucy Akitt
Hasina Angelina Rajoelimbololona
Lorenzo Ierfino

Machine Learning Project - GoodReads - LINK

The project involves developing an end-to-end machine learning pipeline to predict a book's rating using a provided dataset. The tasks include performing exploratory data analysis, feature engineering, model training, and evaluation, followed by the deployment of the final model.

Below is the information provided regarding the dataset attributes:

1) **bookID**: A unique identification number for each book.
2) **title**: The name under which the book was published.
3) **authors**: The names of the authors of the book. Multiple authors are delimited by “/”.
4) **average_rating**: The average rating of the book received in total.
5) **isbn**: Another unique number to identify the book, known as the International Standard Book Number.
6) **isbn13**: A 13-digit ISBN to identify the book, instead of the standard 11-digit ISBN.
7) **language_code**: Indicates the primary language of the book. For instance, “eng” is standard for English.
8) **num_pages**: The number of pages the book contains.
9) **ratings_count**: The total number of ratings the book received.
10) **text_reviews_count**: The total number of written text reviews the book received.
11) **publication_date**: The date the book was published.
12) **publisher**: The name of the book publisher.

Approach

Firstly, we identified any unclear or incomplete fields to get a clean and exploitable dataset. This included:

- eliminating misaligned rows in the dataset
- eliminating entries where the ratings count was 0 as if no ratings have been submitted for the book in question, we cannot use this entry to train our model
- cleaning up title and authors columns so that we had one, easy to read value in each row

We then used the isbn library package to extract the isbn Metadata, in order to try to better map the authors and publishers. Unfortunately the metadata presented a significant number of missing values, particularly for publishers, and the inconsistencies in similarly named publishers and multiple authors remained.

That led us to manually explore every repetitive but not identical value, using the ‘*COMMON PHRASES’* Method for title, and split string for authors and publishers. Data being cleaned, we started to plot with the objective of understanding and defining the features we would use for our machine learning model.

Plotting every aspect of the data helped us to identify imbalance in our dataset, notably, the main column ‘average_rating’ having mainly values going from 3.5 to 4.5.

We found that in grouping the titles by common phrases we could see more of an impact on the average rating.

We experimented with the authors column by comparing ratings when using the first author listed in the field, compared to the author from that row who appears the most frequently in the data set. The results concluded very similar plots, from which, we can conclude that having an author being famous (highly represented in our dataset) does not necessarily lead to a higher average rating for the book.

We can see from value counts that the language codes french, spanish, german and english make up 99% of our data and, in plotting these with respect to average rating, we can observe that all 4 main language codes had an almost identical spread of average rating. We therefore consider that language code will not be very useful in helping a model to predict book rating as the only significant perturbations in rating with respect to language code occur in highly underrepresented categories.

We then experimented with splitting up publishers in terms of main publishing house and branches. The results showed similar average ratings across all branches of numerous publishing houses. We also explored average rating with respect to the size of the publishing house using frequency encoding. The results showed little to no correlation.

On observing average ratings with respect to ratings count, we noticed that we have some particularly low ratings counts in our dataset. For a more reliable reflection of ratings and to eliminate human bias (if only 2 ratings are submitted and these both come from close family members of the author, our predictions would be subject to bias), we decided to eliminate the entries with less than 3 ratings submitted on Goodreads.

Calculating a 'text_review_conversion' (text reviews count/ ratings count), helped us to find a feature with a more important impact on the ‘average_rating’ than the ‘text reviews count’ and ‘ratings count’ separately. This feature tells us, of the people who left a rating for the book, what percentage of them took the time to write a text review. The results indicate that the lowest ratings were generally obtained by books receiving a higher text review conversion rate (people are more likely to write a written review on a book that they didn’t enjoy).

We have chosen to use a classification model as for the practical usage of the ratings prediction, small increments in rating, which would be represented in regression, are unlikely to make a difference to a reader. For example, an average rating of 3.8 would be considered just as good as an average rating of 3.9.

The size of the dataset being limited and the data representation for ‘average_rating’ unbalanced, we have chosen to classify the ranking to give a more readable understanding of the results.

- 0 - 3 = Poor
- 3 - 3.5 = Below Average
- 3.5 - 4.5 = Good
- 4.5 - 5 = Excellent

Model Selection

SMOTE (Synthetic Minority Over-sampling Technique) was used in our machine learning project to balance the dataset by generating synthetic samples for the minority classes (‘Excellent’, ‘Below Average’ and ‘Poor’), to help improve the model's performance and reduce bias toward the majority class (‘Good’).

We decided to focus on decision tree models as they perform well with categorical data as well as with large data sets, and they are able to perform feature selection, taking into account the most significant variables for the prediction. They also mirror human decision making, which is the behavior that we are trying to predict.

- We chose to use RandomForestClassifier as it is versatile and robust, due to the aggregation of numerous trees, while reducing overfitting and scaling well. Additionally, it is easy to understand and implement and it works well with SMOTE to address imbalanced datasets.
- We chose GradientBoostingClassifier as it is known to provide a more accurate, strong prediction by using a combination of precedent predictions, and it allows fine tuning of multiple hyperparameters. Using Grid Search cross validation, we found that using 300 trees at a learning rate of 0.2 and max depth 8 was optimal in getting a good prediction without slowing down the training process too much.
- We then chose to use KNeighborsClassifier as it works well with uneven decision boundaries, and is non parametric which suits the non standard data distribution in our data set. It is easy to understand and interpret, and also provides flexibility by allowing us to adjust the number of neighbors to fine-tune performance (best results at k=3, k=4). We also found better results when using distance to weight the nearest neighbors.

Evaluation Metrics

We used cross validation to evaluate the models performance on different train test splits. This resulted in consistent accuracy scores across the board meaning that our models perform consistently well and are unlikely to be overfitting.

The models were evaluated on Accuracy, Precision, Recall and F1 score.

Both the Random Forest Classifier and Gradient Boosting Classifier performed very similarly, receiving an accuracy of around 95% with near perfect scores across all evaluations for the Poor and Excellent classes (97-100%). They performed slightly less accurately, but still satisfactory for our use case, when classifying Good and Below Average ratings (89-93%).

With the confusion matrix showing us that the majority of incorrect predictions were Good being classified as Below Average and vice versa.

The K Nearest Neighbors Classifier generally performed well with an accuracy score of 83%. Its results showed us a less accurate version of similar trends to the previous models. With the recall in the Good category being the lowest (57%) and the precision of the Below Average being the lowest (76%). Again from the confusion matrix we can see that the majority of these cases are incorrectly classified between the Below Average and Good classes.

6. Conclusion

Overall, our selected models performed relatively well. While we restrained the overall accuracy by categorizing our data, we obtained a realistic result that we think is exploitable as it is. Moreover, the dataset being unbalanced and our model being limited by the category, the result could be subject to bias.

Furthermore, we believe that it could be improved with a bigger and better balanced dataset which would enable us to get rid of the categories and go further with a model that can predict more precise future ratings of a book.

7. References

https://www.kaggle.com/code/yedigeashmet/goodreads-prediction-with-some-eda

https://www.kaggle.com/code/hoshi7/goodreads-analysis-and-recommending-books

https://fr.wikipedia.org/wiki/International_Standard_Book_Number