Shiqi Wang
Brown University, Data Science Institute
GitHub Repository
December 9, 2023

Amazon Kindle E-books Rating Prediction

1. Introduction

1.1. Motivation

Amazon Kindle, a leading platform for digital reading, offers a vast collection of e-books accessible worldwide [1]. The star ratings of e-books on Kindle are pivotal in guiding customer purchases, reflecting the quality and appeal of the content. Accurate prediction of these ratings can significantly aid Amazon in crafting targeted business and marketing strategies, tailoring recommendations to customer preferences, enhancing user satisfaction, and potentially boosting sales. This predictive analysis is key for Amazon to maintain its competitive edge in the dynamic digital book market.

1.2. Data Description

The Amazon Kindle Book Dataset was sourced from Kaggle, and it was originally scrapped from publicly available data source in October 2023 [2]. It encompasses information on e-book publications available on the Amazon Kindle platform, accessible to any user. With the data collection occurring shortly before the commencement of this project, the dataset is current and reflects the latest trends.

The raw data contains 133,102 entries and 16 columns. Continuous target variable “stars” represents the ratings given to each e-book, with a possible score ranging from 1 to 5. During the data cleaning process, entries with missing target variables and columns unrelated to the regression questions were removed. The refined data has 129,920 data points and 11 features: 2 continuous features, 6 categorical features, and 3 time-based ordinal features.

1.3. Previous Work

The Amazon Kindle Book Dataset's novelty means few machine learning studies have been conducted with it, with one notable effort using a CatBoost regressor to reduce RMSE from roughly 0.7 (baseline) to 0.5 stars. However, the study's method of filling missing targets with 0 creates a baseline RMSE that greatly diverges from this project, limiting its relevance as a benchmark. [3]

2. Exploratory Data Analysis

2.1. Target Variable

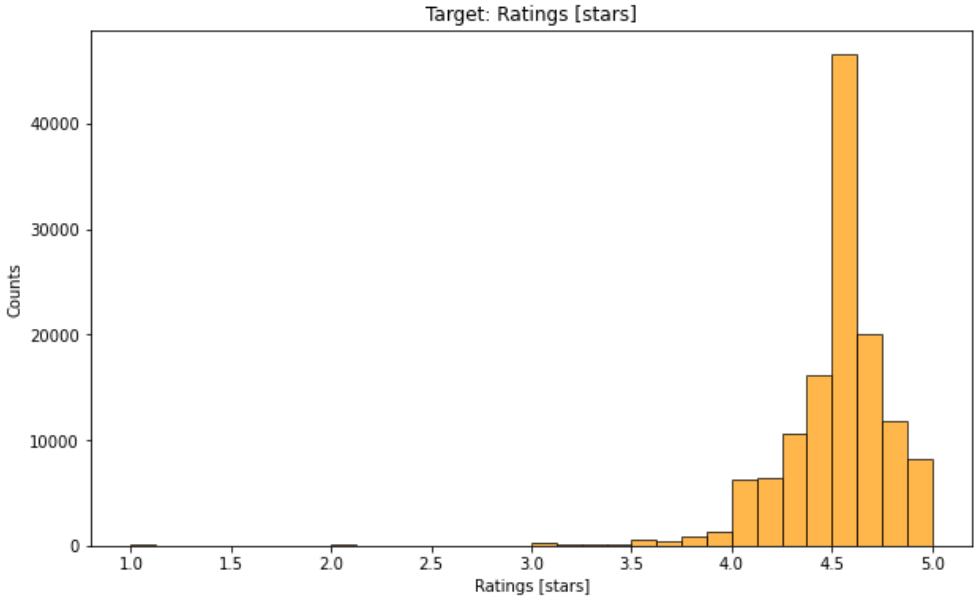

Figure 1: Distribution of Target Variable: Ratings in [stars]

Target variable to be predicted for the regression question is ratings, with a unit in stars and a range from 1 to 5. The target is strongly left-skewed, with several outliers at the lower end. Thus, stratifying approach should be used when splitting the data in the pipeline. Also, majority of the data points are concentrated in the 4 to 5 range.

2.2. Features Analysis

2.2.1. Contiguous Features

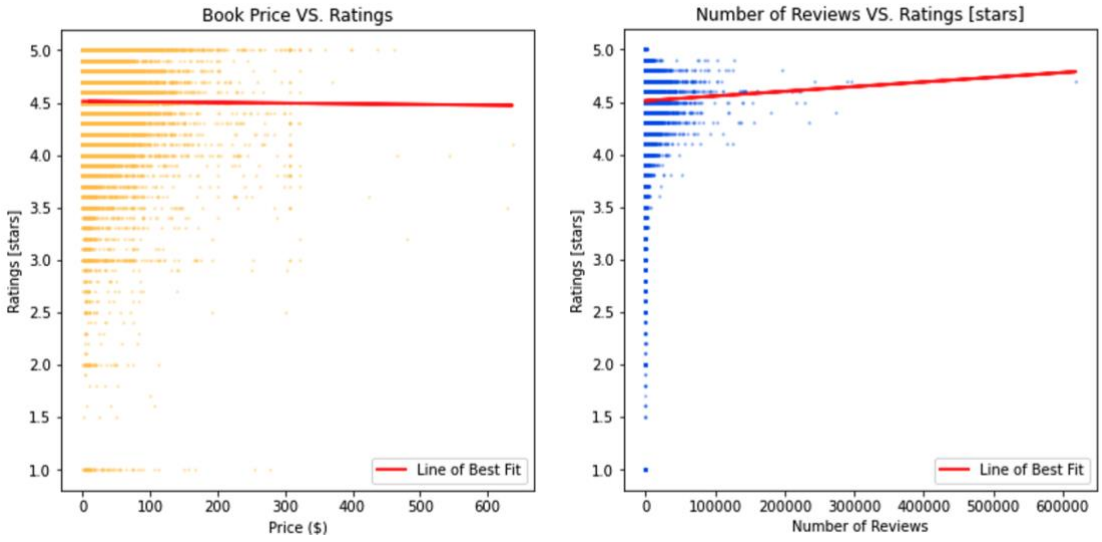

Figure 2: Scatter plots of Price VS. Target variable, and Reviews VS. Target variable

The features 'price' and 'number of reviews' show a weak linear relationship with the target variable 'ratings'. The best-fit line for 'price' is nearly horizontal, indicating negligible linear predictive power. Meanwhile, 'number of reviews' has a slightly positive but still weak correlation with 'ratings', suggesting limited linear predictive capability for this feature.

2.2.2. Categorical Features

Figure 3: Box plot of Ratings distribution for the most numbered 12 publishers, rank by count

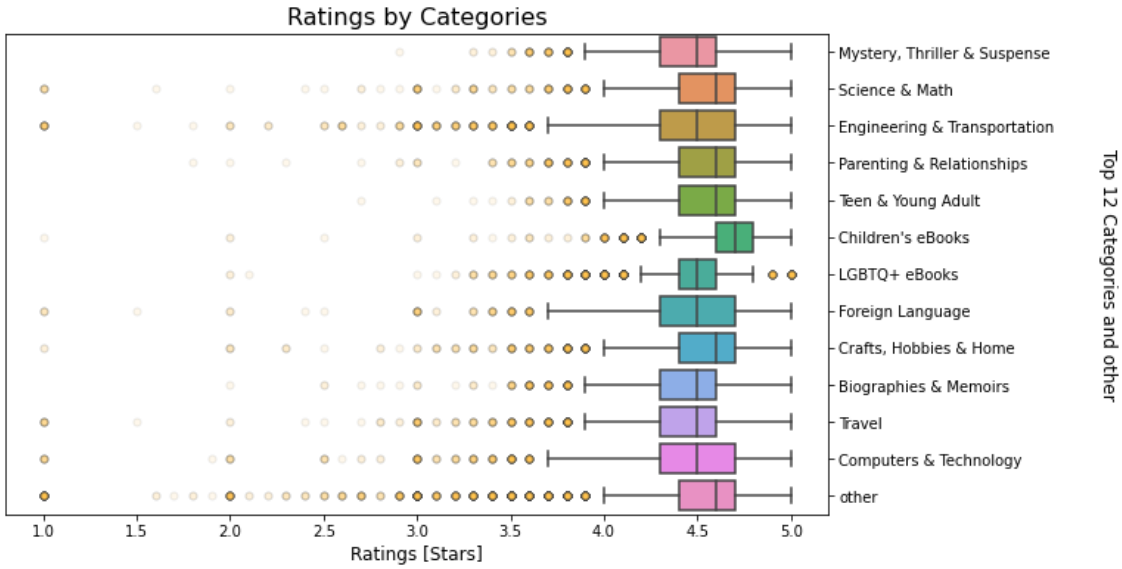

Figure 4: Box plot of Ratings distribution for the most numbered 12 book categories, rank by count

The visualizations for 'publisher' and 'categories' features indicate that the ratings vary across different publishers and categories, with each exhibiting a distinct distribution pattern. Thus, these features might provide substantial discriminative power. Notably, the majority of the ratings across these features are concentrated in the 4 to 5 range, indicating a trend towards higher ratings in the dataset.

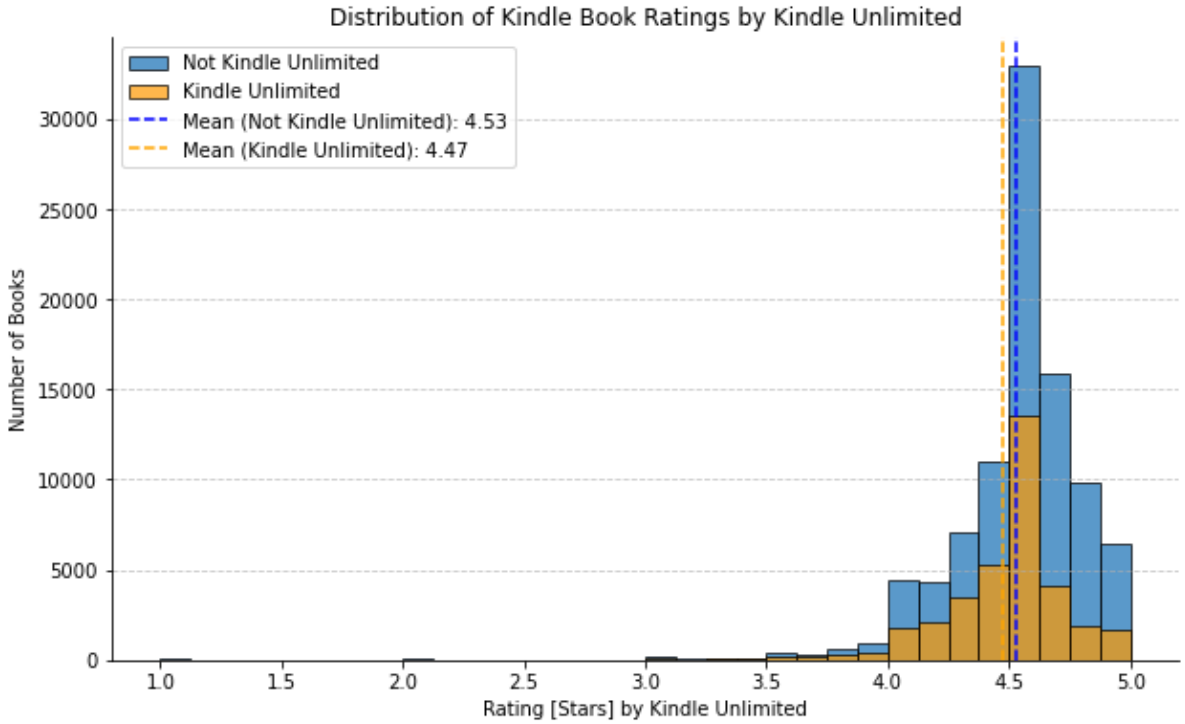

Figure 5: Ratings distributions of e-books that are Kindle Unlimited and are not Kindle Unlimited

The analysis of the binary feature 'Kindle Unlimited' reveals a discernible difference in the distribution of ratings. Interestingly, e-books that are not part of the Kindle Unlimited membership program tend to have higher overall ratings than those that are included.

3. Methods

3.1. Data Splitting

The left-skewed nature of the target variable necessitates stratification in splitting the data to ensure balanced and representative training and evaluation sets. To accommodate stratification with a continuous target variable, the ratings are binned into 'Low', 'Medium', and 'High' categories. The data is then split with stratification into an 'other' set (80%) and a 'test' set (20%), with each set reflecting the original distribution of the rating categories. Stratified K-fold (4 folds) is applied within the 'other' set for cross-validation, maintaining the same distribution of binned ratings. This results in the data being divided in a 3:1:1 ratio across training, validation, and test sets.

3.2. Data Preprocessing

In preprocessing, Ordinal, OneHot and StandardScaler are used. The time-based features are encoded ordinally, assigning an integer to each unique category according to a predetermined order. This step is crucial to preserve the natural ordering of the data, allowing the model to accurately

interpret these temporal patterns. Simultaneously, nominal features and binary features undergo one-hot encoding, which transforms them into binary columns, enabling the model to process categorical data without any inherent order. Continuous features are standardized using the StandardScaler to neutralize the scale of the features and preventing any one feature from disproportionately influencing the model's predictions.

There are 37% missing values in the three time-based features and 7% in 'publisher' feature. The most frequent value is used as imputer, which statistically represents the most likely occurrence. There is no missing value in continuous features.

3.3. Machine Learning Pipeline

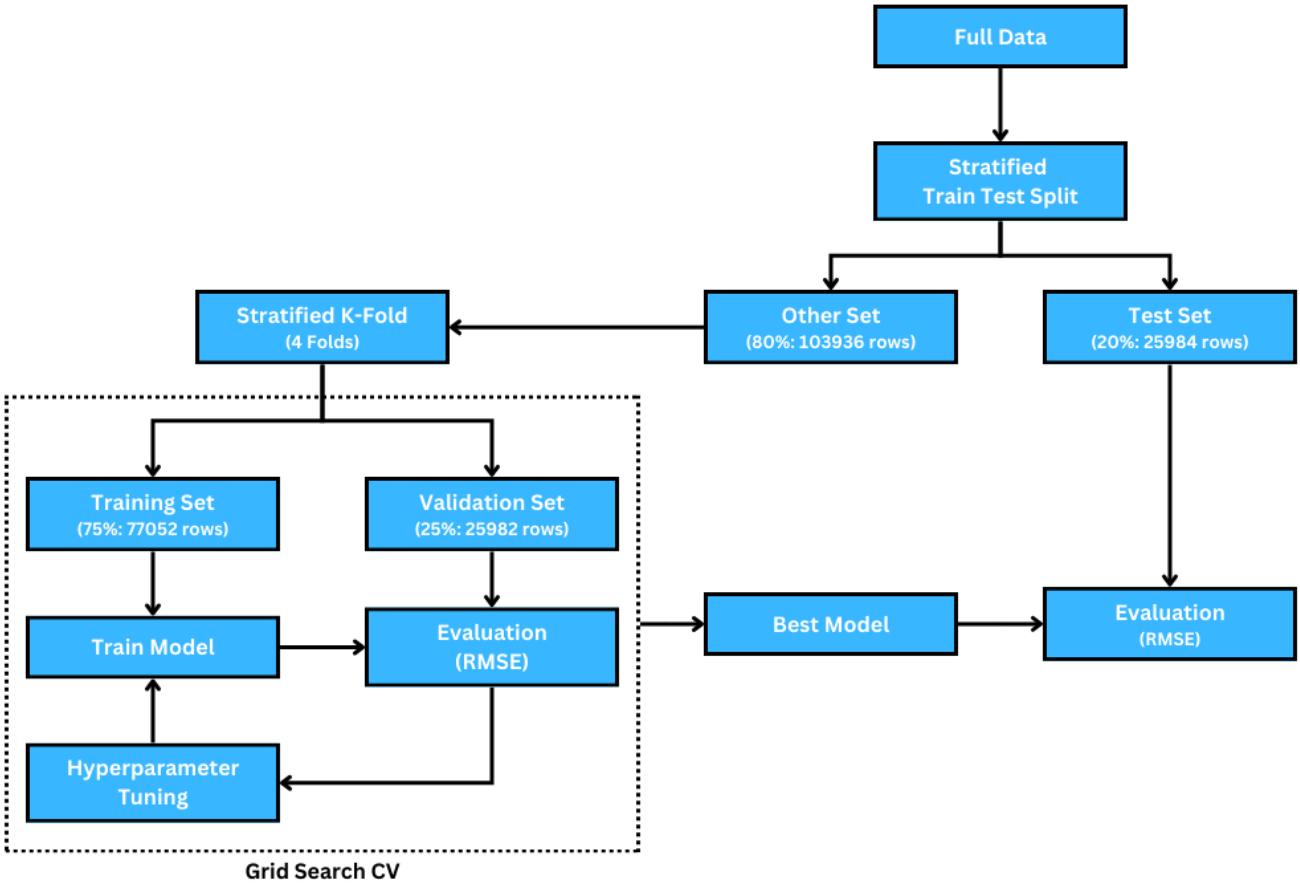

Figure 6: Machine Learning Pipeline Diagram

The machine learning pipeline is repeated over 3 random states to evaluate their performance stability and account for variability in non-deterministic models.

3.4. Machine Learning Algorithms and Hyperparameters Tuning

Algorithm	Parameters
Linear Regression: **Lasso**	alpha(L1 regulation): [0.0001, 0.001, 0.01, 0.1, 1, 10]
Linear Regression: **Ridge**	alpha(L2 regulation): [0.01, 0.1, 1, 10, 100, 1000]
Linear Regression: **Elastic Net**	alpha: [0.0001, 0.001, 0.01, 0.1, 1, 10] l1_ratio: [0.0, 0.25, 0.5, 0.75, 1.0]
Random Forest	n_estimators: [10, 50, 100, 200, 300] max_depth: [3, 5, 10] max_features: [0.25, 0.5, 0.75, 1.0]
XGBoost	max_depth: [2, 3, 4, 5, 6] learning_rate: [0.01, 0.1,0.3] n_estimators: [200, 300] reg_alpha(L1 regulation): [0, 0.01, 0.1] colsample_bytree: [0.9] subsample: [0.66]

Table 1: Machine learning algorithms and hyperparameters used for tuning.

In the hyperparameter tuning process, GridSearchCV is used search in for potential hyperparameter values. A pipeline, including preprocessor and the machine learning algorithm, is used to ensure consistent data processing. GridSearchCV performs K-fold cross-validation for each hyperparameter combination, ultimately identifying the combination with the lowest RMSE across folds, thus determining the optimal hyperparameters for the model.

Root Mean Squared Error (RMSE) is chosen as the evaluation metric because it provides a clear measure of model accuracy by calculating the square root of the average squared differences between predicted and actual values. Furthermore, it shares the same unit with the target variable. For better evaluation across algorithms, R^2 is also recorded.

4. Results

4.1. Model Performance

	Baseline	Lasso	Ridge	Elastic Net	Random Forest	XGBoost
RMSE: mean	0.288	0.278	0.278	0.280	0.275	0.270
RMSE: std		0.0020	0.0020	0.0036	0.0019	0.0024
R^2 : mean		0.0633	0.0647	0.0648	0.0835	0.1161

Table 2: ML algorithm best model performance with scores comparing to baseline score.

The results indicate that all machine learning models outperformed the baseline in terms of RMSE, with XGBoost achieving the lowest mean RMSE of 0.270, suggesting it's the most accurate model. In terms of stability, Lasso and Ridge show the least variation in RMSE with a standard deviation of 0.0020. When considering the R^2, XGBoost also leads with a mean R^2 of 0.1161, indicating it explains a higher proportion of variance in the target variable compared to other models.

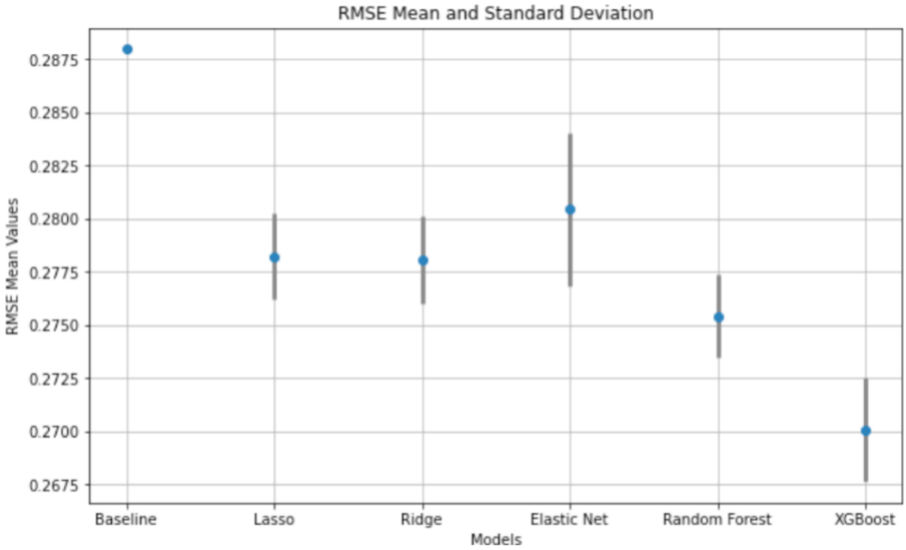

Figure 7: Visualization of RMSE score across baseline and ML models.

The plot showcases non-linear models (XGBoost and Random Forest), outperform linear models (Lasso, Ridge, and Elastic Net), suggesting that the underlying relationship between the features and the target variable may be complex, making non-linear models more suitable for capturing the nuances in the data. Also, Lasso and Ridge are both outperforming Elastic Net, suggesting that the dataset does not benefit from the combination of L1 and L2 regularization.

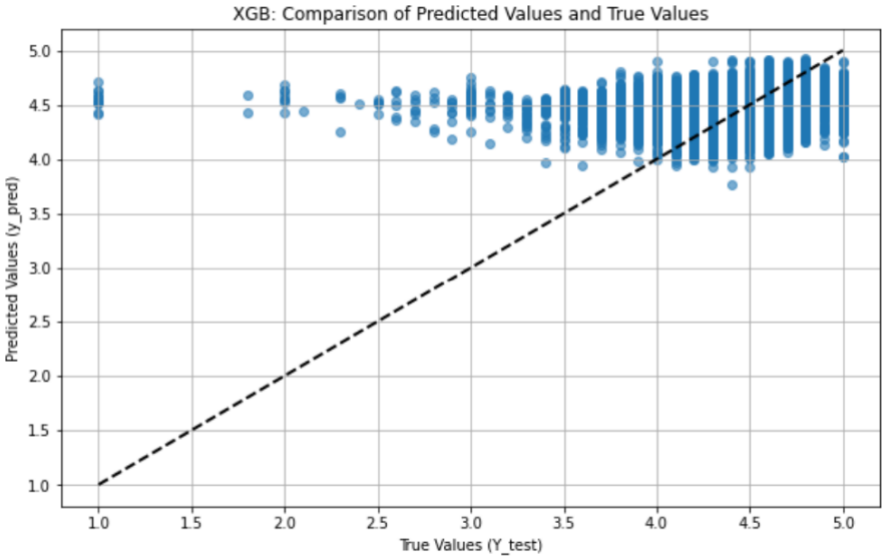

Figure 8: Scatter plot of true test values and predicted test values by the best XGBoost model.

The scatter plot suggests that the top-performing XGBoost model predominantly predicts higher ratings, deviating from the diagonal line which indicates a discrepancy from the true values and thus poor accuracy.

4.2. Feature Importance

4.2.1. Global Importance

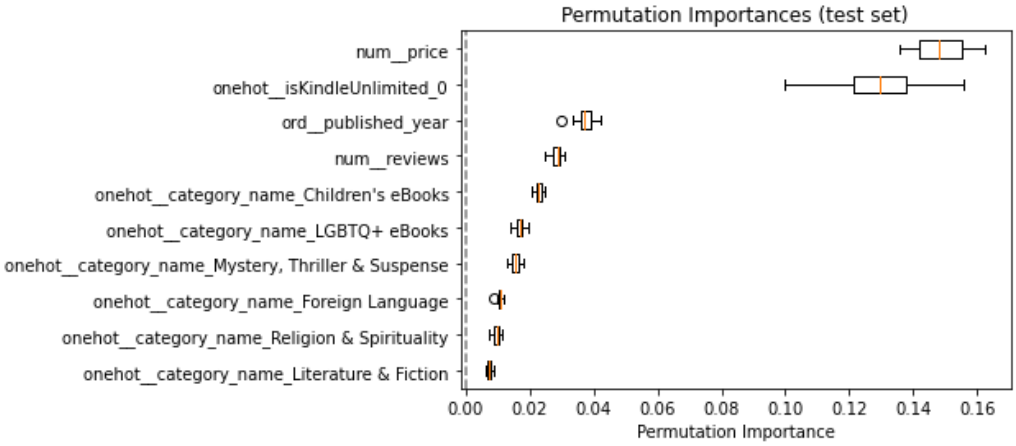

Figure 9: Top 10 permutation importance with best XGBoost model test set

According to the permutation importance plot, the XGBoost model will suffer from most decrease in accuracy if values of 'price' feature or 'Kindle Unlimited (0)' feature is randomly shuffled.

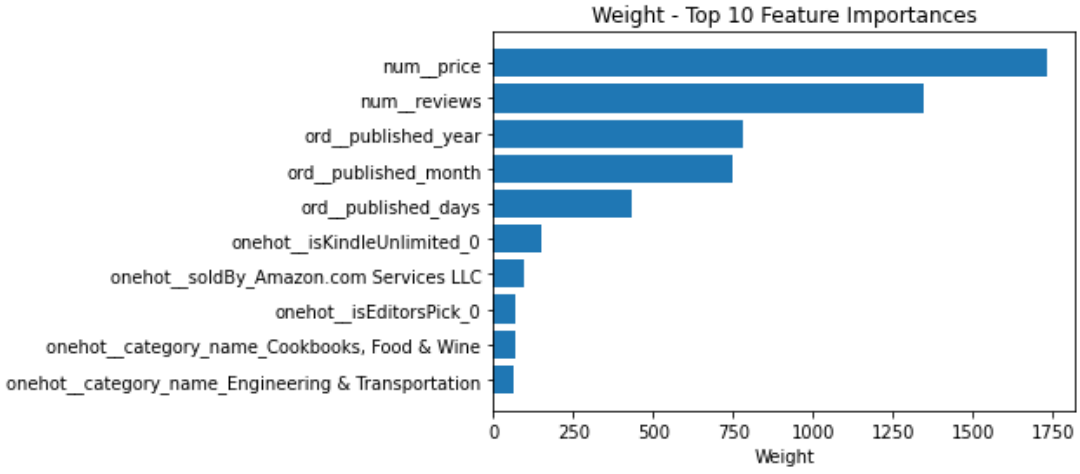

Figure 10: Top 10 important features by XGBoost weight

According to the XGBoost weight metric, 'price' feature and 'reviews' feature have significantly greater importance than other features.

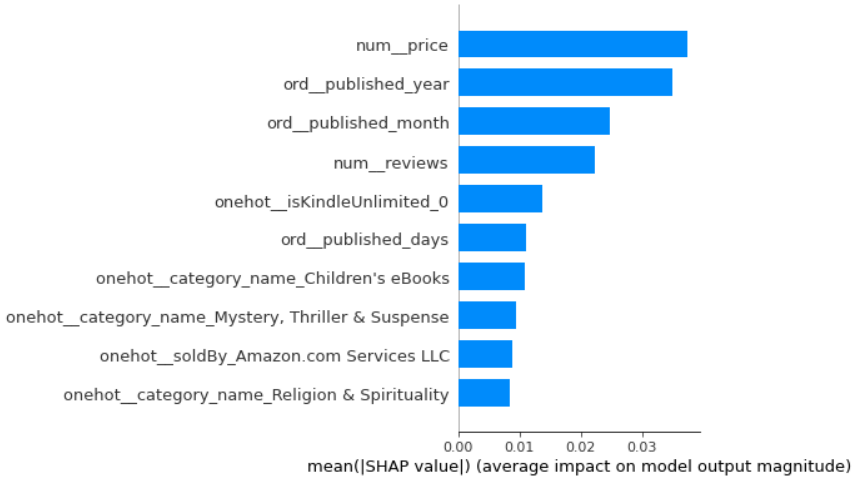

Figure 11: XGBoost SHAP top 10 global feature importance

SHAP global feature importance shows that 'price' and 'published year' are the most important features.

In the three analysis of global feature importance, 'price' emerged as the most important factor across all measures. The features 'reviews' and 'published year' were consistently among the top five in terms of importance. The feature 'Kindle Unlimited (0)', and 'published month' also ranked within the top five in two of the three measures. Moreover, certain book categories and publishers were identified as top features, although their rankings did not consistently align across the different measures of importance. In conclusion, the two continuous features, 'price' and 'reviews' are the most important features, and time-based features also have high importance than other categorical features.

4.2.2. Local Importance: SHAP

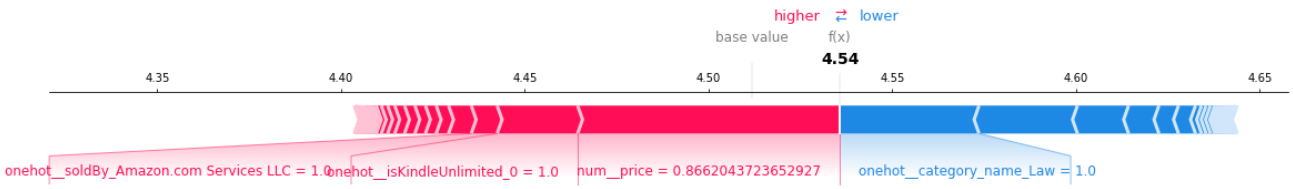

Figure 12: SHAP local feature importance- Index 0

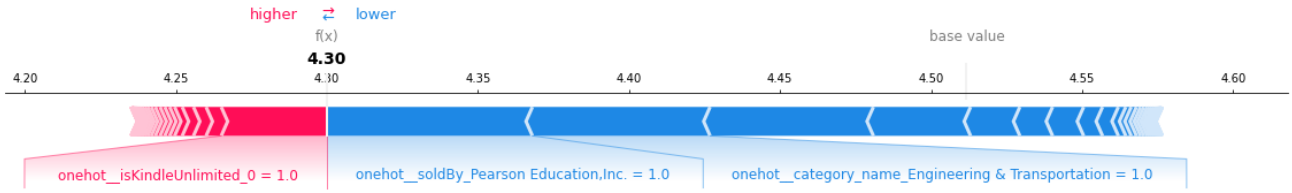

Figure 13: SHAP local feature importance- Index 100

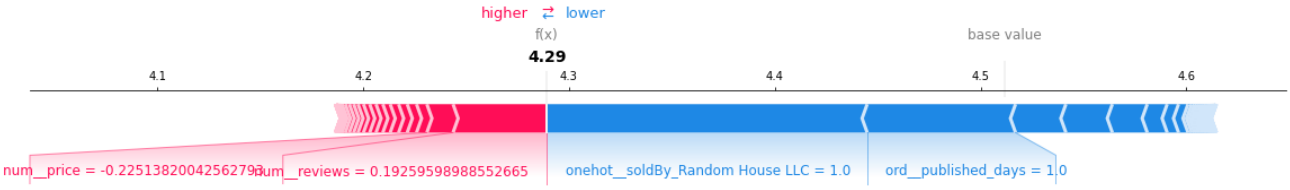

Figure 14: SHAP local feature importance- Index 200

The local SHAP importance plots indicate that the 'price' of an e-book lowers the model's rating prediction across different instances, while 'reviews' and specific categories like 'Law' and 'Engineering & Transportation' tend to increase it. Moreover, the presence or absence of features like 'Kindle Unlimited' and the publisher (e.g., 'Random House LLC', 'Pearson Education, Inc.') also show varied effects on the ratings.

5. Outlook

Looking ahead, there is room to bolster the data collection process. With only 11 related features currently in use, the ability of the model to detect complex patterns may be limited. Adding more features could provide deeper insights into what affects e-book ratings. Moreover, bringing together data from various sources could enhance the dataset, giving a greater picture and possibly revealing patterns invisible with a single source.

In terms of preprocessing, there's a need for improvement, especially since a significant portion (37%) of the time-based data is missing. Finding and using a more advanced method to fill in these gaps could greatly improve the dataset's quality, leading to more precise predictions from the model.

Finally, exploring a broader array of hyperparameter settings could lead to discovering a more effective combination, which would enhance model performance. For instance, increasing the number of estimators in the XGBoost model from the current 300 might yield stronger and more accurate predictions.

Reference:

[1] https://en.wikipedia.org/wiki/Kindle_Store

[2] https://www.kaggle.com/datasets/asaniczka/amazon-kindle-books-dataset-2023-130k-books/data

[3] https://www.kaggle.com/code/dima806/amazon-kindle-books-rating-autoviz-catboost-shap/notebook

[4] https://scikit-learn.org/stable/index.html

GitHub Repository:

https://github.com/ccwxp116/Data1030Project