Feature Engineering On Book Price Prediction

Mohammad Rouintan

Computer Science, Shahid Beheshti University

Abstract

This report presents a detailed analysis of a book price prediction dataset and explores the effectiveness of the Random Forest regression algorithm in predicting book prices. The dataset consists of various features such as title, author, edition, reviews, ratings, synopsis, genre, book category, and the target variable, price. With 5,699 instances, the dataset primarily contains numerical and categorical features, with price being the target variable. Through exploratory data analysis, feature engineering, and the application of the Random Forest regression model, this study aims to provide valuable insights into book pricing and assess the performance of the model in predicting book prices accurately.

Introduction

The pricing of books is a critical aspect of the publishing industry, influenced by a multitude of factors such as author reputation, edition type, customer reviews, genre, and book category. Accurately predicting book prices is vital for both book retailers and consumers. Retailers can optimize their pricing strategies to maximize profitability, while consumers can make informed decisions based on their budget and preferences. In this study, we analyze a book price prediction dataset that encompasses various features including title, author, edition, reviews, ratings, synopsis, genre, book category, and price.

The primary objective of this research is to develop a predictive model using the Random Forest regression algorithm to accurately estimate book prices based on the available features. The Random Forest algorithm, a powerful ensemble learning technique, is well-suited for regression tasks and can handle both numerical and categorical features effectively. By leveraging this algorithm, we aim to uncover the underlying patterns and relationships within the dataset, enabling us to make reliable predictions of book prices.

To begin the analysis, we conduct exploratory data analysis to gain insights into the distribution of book prices and explore the relationships between different features. This analysis helps us identify potential correlations and patterns in the dataset. Through visualizations and statistical techniques, we aim to identify influential factors and outliers that may impact book prices.

Next, we focus on feature engineering, a crucial step in developing an accurate predictive model. Feature engineering involves transforming and extracting meaningful information from the existing features and creating new ones that better capture the relationship between the predictors and the target variable, price. Techniques such as text mining, sentiment analysis, and encoding methods for categorical variables may be employed to enhance the predictive power of the model. Additionally, we explore the impact of genre and book category on book prices and consider their appropriate representation in the model.

Once the feature engineering phase is complete, we train and evaluate the Random Forest regression model. The model is trained using the available dataset, and its performance is assessed using appropriate evaluation metrics such as mean squared error or root mean squared error. We also utilize techniques like cross-validation to ensure the model's robustness and generalizability.

Finally, we summarize our findings, discuss the implications of the results, and provide recommendations for book retailers and consumers based on the performance of the Random Forest regression model. The insights gained from this study can assist in optimizing pricing strategies, understanding pricing dynamics in the book market, and aiding decision-making processes in the publishing industry.

In conclusion, this article presents a comprehensive analysis of a book price prediction dataset, employing the Random Forest regression algorithm to accurately predict book prices. By understanding the factors influencing book prices and leveraging the power of the Random Forest algorithm, retailers and consumers can make informed decisions and navigate the book market more effectively.

Methodologies

Exploratory Data Analysis

First of all, let's say that the names of the two columns, Reviews and Ratings, have been changed and we changed them before starting our work.

Second, let's take a look at the target. Our target is the price of the book, which according to the box

plot and violin plot that I see below, the prices have outliers and this can cause damage to our model, although random forest models aren't very sensitive to noise and outlier data. We removed outliers with ratios of 0 and 0.99.

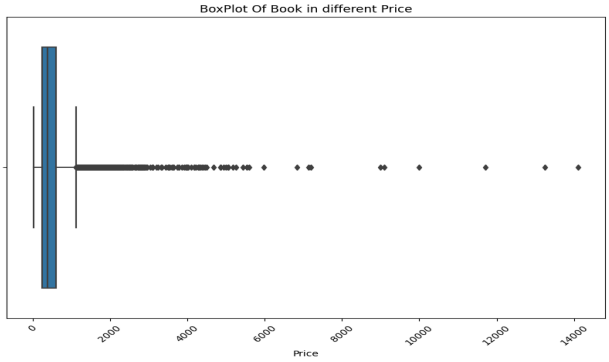

Figure 1: *Box plot of price before removing outliers*

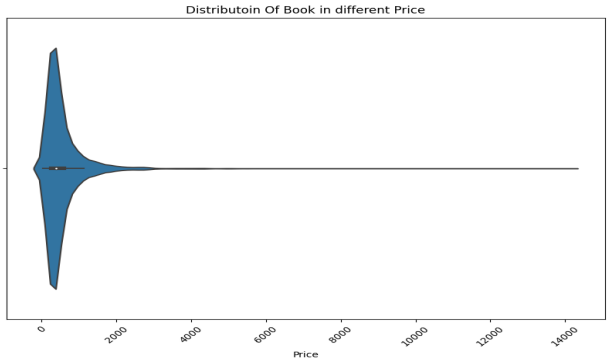

Figure 2: *Violin plot of price before removing outliers*

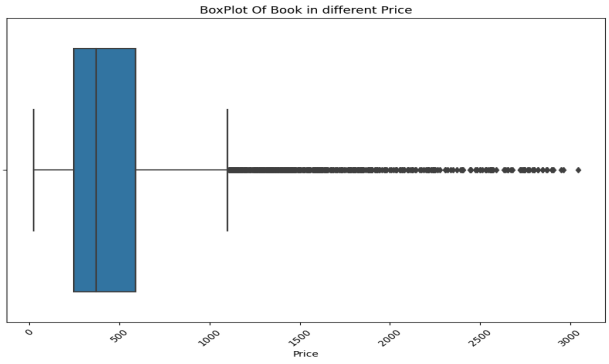

Figure 3: *Box plot of price after removing outliers*

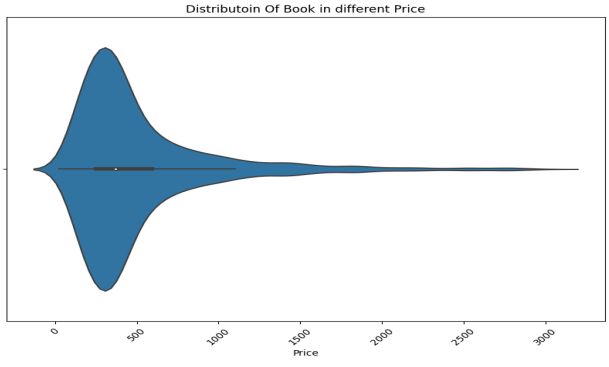

Figure 4: *Violin plot of price after removing outliers*

We have also performed an analysis based on book categories on this dataset, the results of which can be seen below.

BookCategory	NumberOfBook
Action & Adventure	747
Crime, Thriller & Mystery	656
Language, Linguistics & Writing	548
Biographies, Diaries & True Accounts	543
Comics & Mangas	530
Romance	514
Humour	475
Arts, Film & Photography	468
Computing, Internet & Digital Media	451
Sports	419
Politics	291

Figure 5: *This table shows the number of books in each category, and we can see that the largest number of books is related to the Action & Adventure category.*

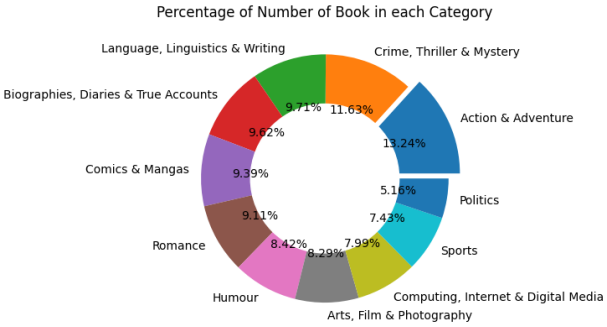

Figure 6: *This Donut plot shows what percentage of all books belong to each category.*

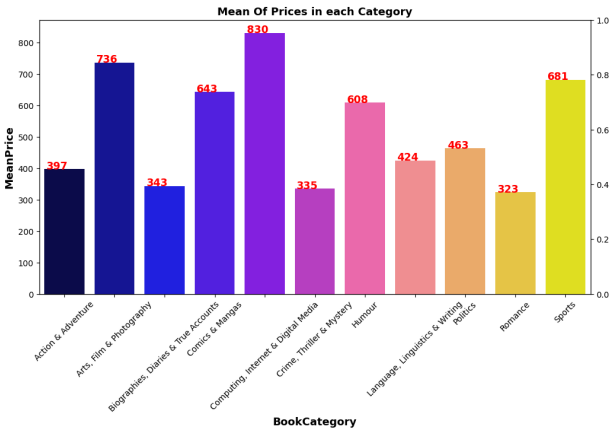

Figure 7: *This Bar plot shows the mean of prices in each category.*

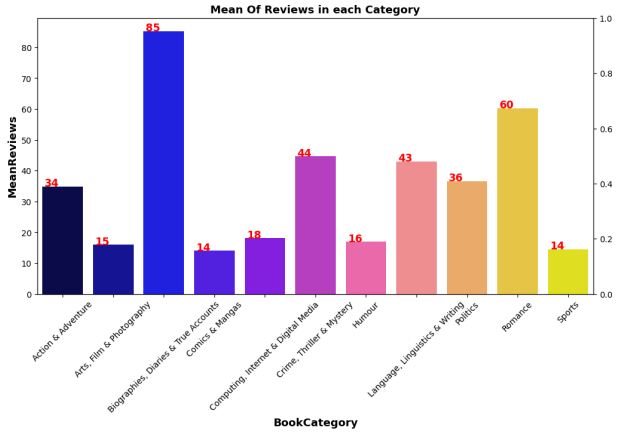

Figure 8: *This Bar plot shows the mean of reviews in each category.*

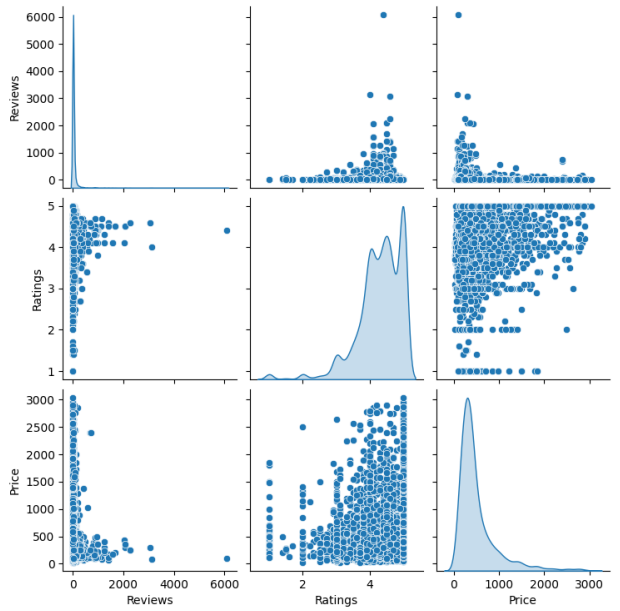

Figure 9: *This pair plot shows the relationship between three features (Reviews, Ratings, Price). The result shows that increasing the rating can increase the price.*

Feature Engineering

The feature "Ratings" is expressed in the form of a phrase in each line, and we extracted the grade given from this phrase and placed it in the same feature. Also, feature "Reviews" are expressed as an expression that shows the number of reviews, which we extracted from the expression and placed in the same feature.

We divided one of the features called "Edition" into two features, "CoverType" and "PublishedYear", the first feature shows the first part of the feature Edition, and the second feature shows the second part of the feature Edition.

Because all the data features are in the form of text, we converted them into vectors using TFIDFVectorizer and CountVectorizer methods so that we can use them to train the Random Forest model. In the following, we briefly explain the two methods of TFIDFVectorizer and CountVectorizer.

TF-IDFVectorizer & CountVectorizer: TF-IDF Vectorizer and Count Vectorizer are both methods used in natural language processing to vectorize text. However, there is a fundamental difference between the two methods.

CountVectorizer simply counts the number of times a word appears in a document (using a bag-of-words approach), while TF-IDF Vectorizer takes into account not only how many times a word appears in a document but also how important that word is to the whole corpus.

This is done by penalizing words that often appear across all documents, reducing the count of these as these words are likely to be less important.

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

Figure 10: *Formula of TF-IDF*

$$\mathrm{TF}-IDF = TF(t,d) * \mathrm{IDF(t)}$$

$TF(t,d)$ = Number of Times Term t Appears in doc, d

We know that's just our CountVectorizer

$$\mathrm{TF}-IDF = CountVectorizer(t,d) * \mathrm{IDF(t)}$$

Figure 11: *Difference between TF-IDF and Count vectorizer*

Before using these methods, we combined some features and created two new features with the names of "CombineFeature1" and "CombinedFeature2". CombineFeature1 is obtained by concatenating "Synopsis" and "Title" features, and CombineFeature2 is obtained by concatenating "Author", "Genre", "Category" and "CoverType" features.

To convert the CombineFeature1 into a vector, we used TF-IDFVectorizer and to convert the CombineFeature2 into a vector, we used CountVectorizer. Using these two methods, we received a vector for each data line, and concatenated these vectors to each of the data lines.

Finally, we removed all the features that are in the form of text, because we converted them into vectors of numbers and added them to the dataset.

Now, before we give the data to train the model, there is a problem. This problem is caused by the use of TF-IDFVectorizer and CountVectorizer, these

two methods have caused the number of features in our dataset to increase a lot. The number of features reaches about 40,000. In order to reduce the number of features, we used the famous PCA method. Of course, before running PCA, we scaled the data using standard scaling.

After doing these things, we gave the data to the Random Forest Regressor model for training, and the following result is based on the Train data and Test data.

MSE Train: 18830.683280194244
MSE Train: 151050.45282689718

Results

The main goal of this project was to perform feature engineering methods to improve the result of the model. There are many methods to do this. But deciding which method to choose should be done first based on the type of dataset and its features, and then based on the chosen model and its challenges. This dataset was mostly in the form of text, and the methods that can convert texts into vectors in a good way are efficient methods. In this project, we chose two relatively simple methods, but we can use deeper methods in the field of natural language processing like BertTokenizer and etc.