Feature engineering on book dataset

Sarah Hosseini Feshtami 400222026

Shahid Beheshti University

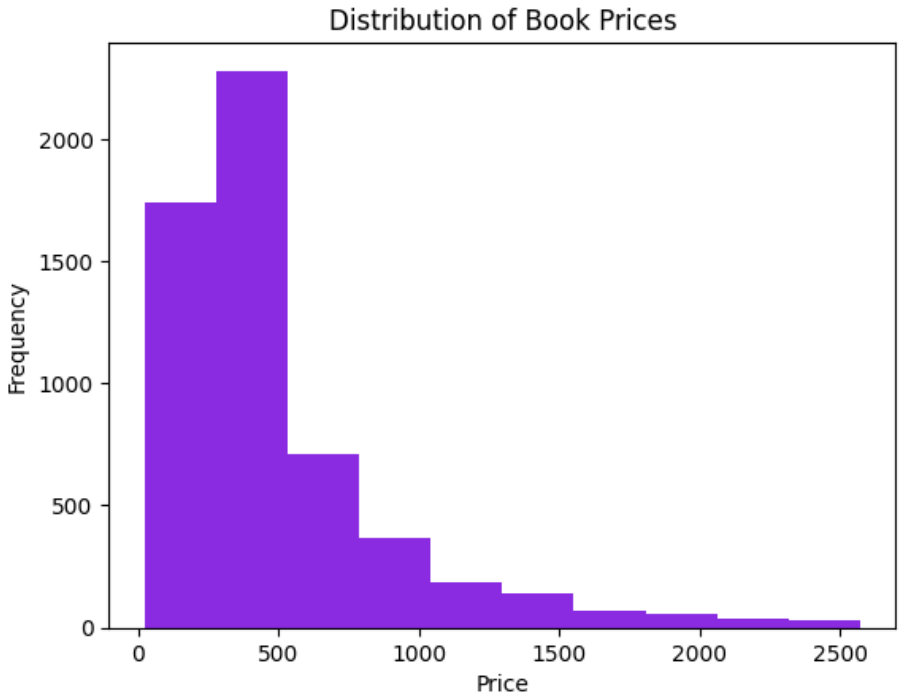
1. Data collection and preprocessing

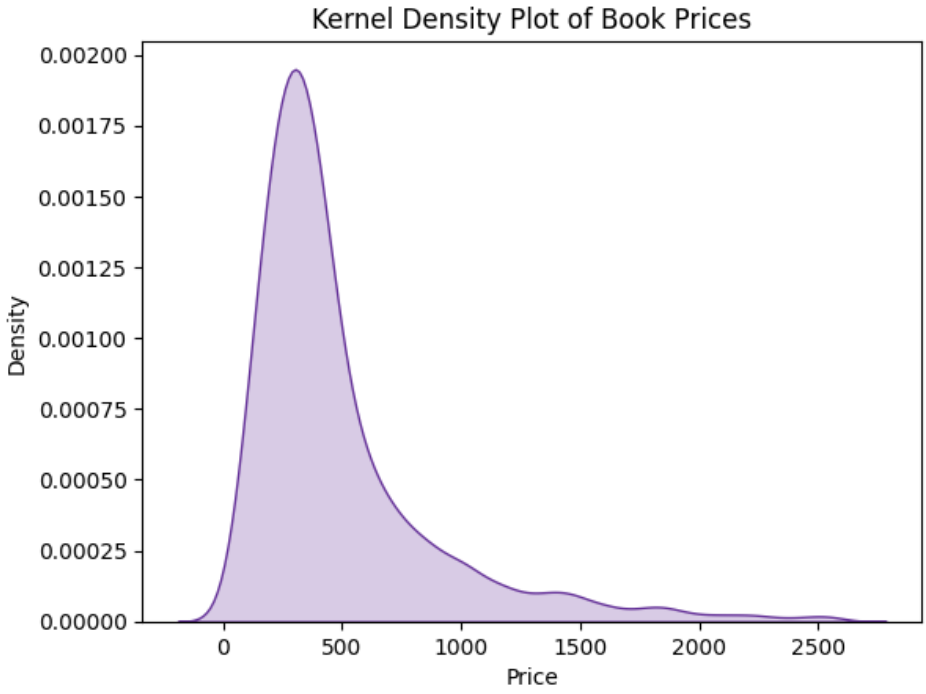
The dataset we will be working on is about books. It has 5699 records with 9 columns. The columns are: ['Title', 'Author', 'Edition', 'Reviews', 'Ratings', 'Synopsis', 'Genre', 'BookCategory', 'Price']. Only one of them is categorical with less than 20 unique values (BookCategory) and only one is numerical (price). The ‘Reviews’ and ‘Ratings’ columns are ordinal but they need some transformation, since all of the records in these columns have the same strings attached to them which is taking up extra space while being completely unnecessary. Also, these columns’ names should be switched, since the ‘Ratings’ contains the number of reviews and ‘Reviews’ contains the stars (rating):

	Ratings	Reviews
0	8 customer reviews	4.0 out of 5 stars
1	14 customer reviews	3.9 out of 5 stars
2	6 customer reviews	4.8 out of 5 stars
3	13 customer reviews	4.1 out of 5 stars
4	1 customer review	5.0 out of 5 stars

We dropped the duplicate records which turned out to be none. Then we identified the unuseful columns (identical or highly missing) which again were none. Then we used the z-score formula to remove outliers and successfully removed 92 records. Then we checked which features had some missing values but found out there were no missing values in our dataframe.

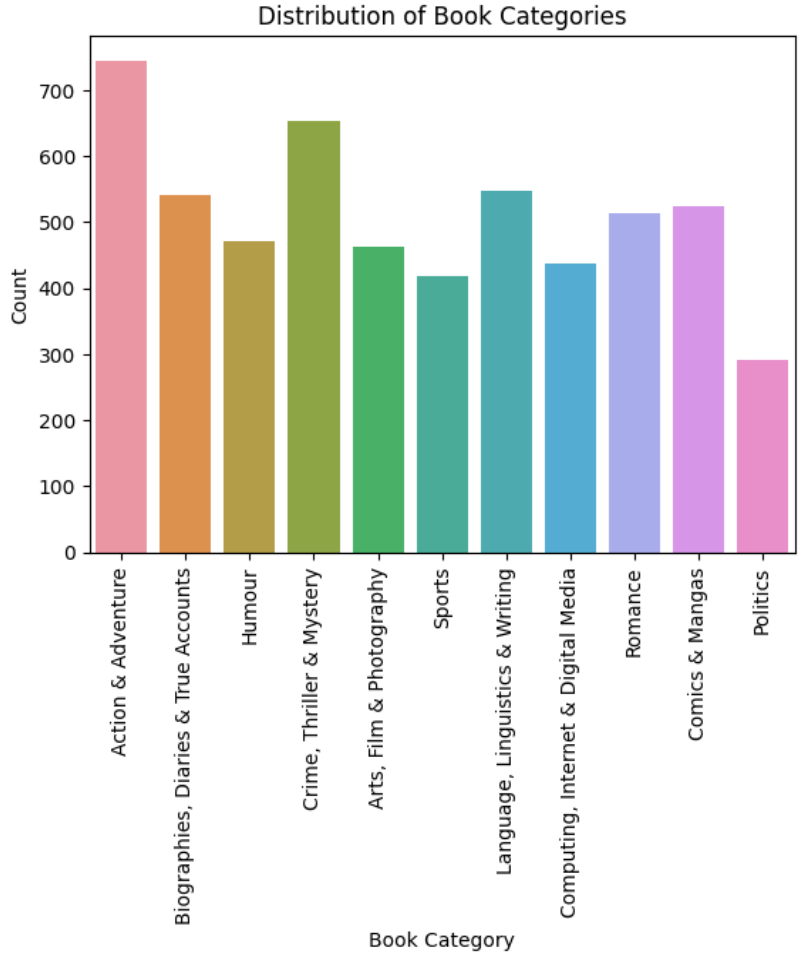
We plotted the frequency of prices and had these plots:





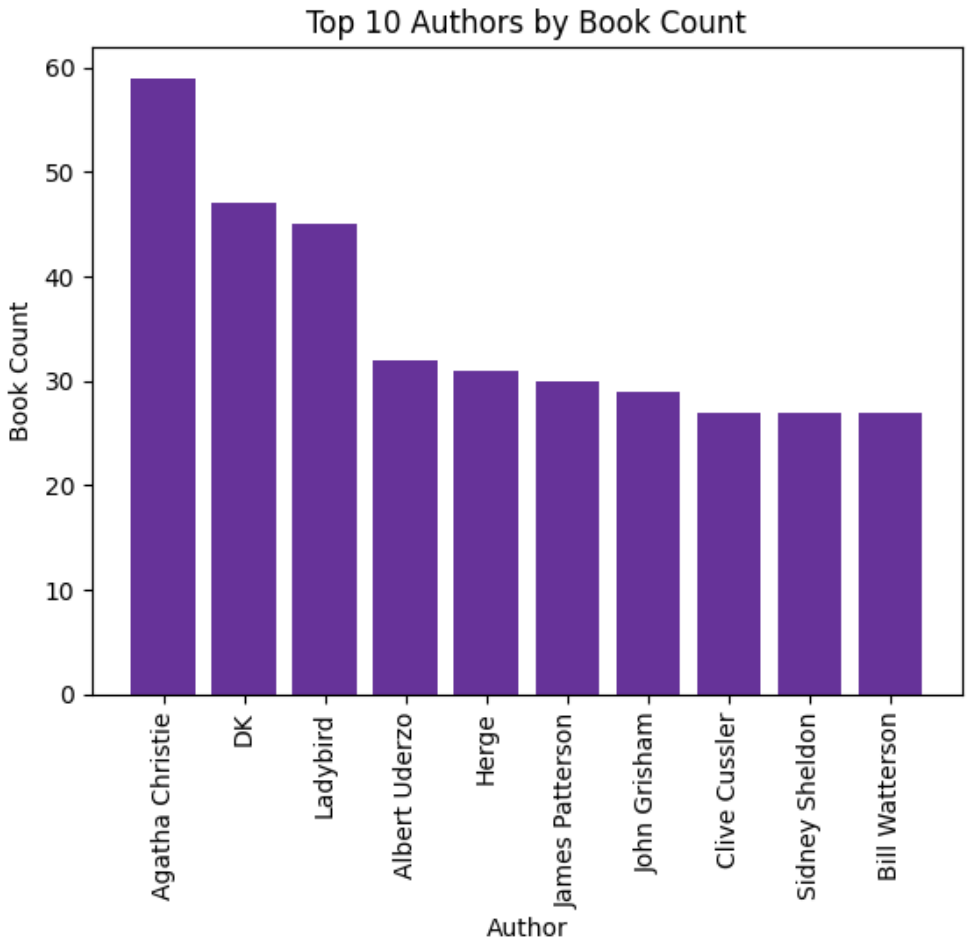
This shows that we have a skewed distribution and most books have a lower price.

This is the Quantile-Quantile (Q-Q) Plot of our data that shows the data follows a right-skewed distribution:



This plot shows the distribution of books based on categories. Most books are in action and adventure and then in crime, thriller, and mystery categories.

Then we visualized the writers with the most count of books in our data:

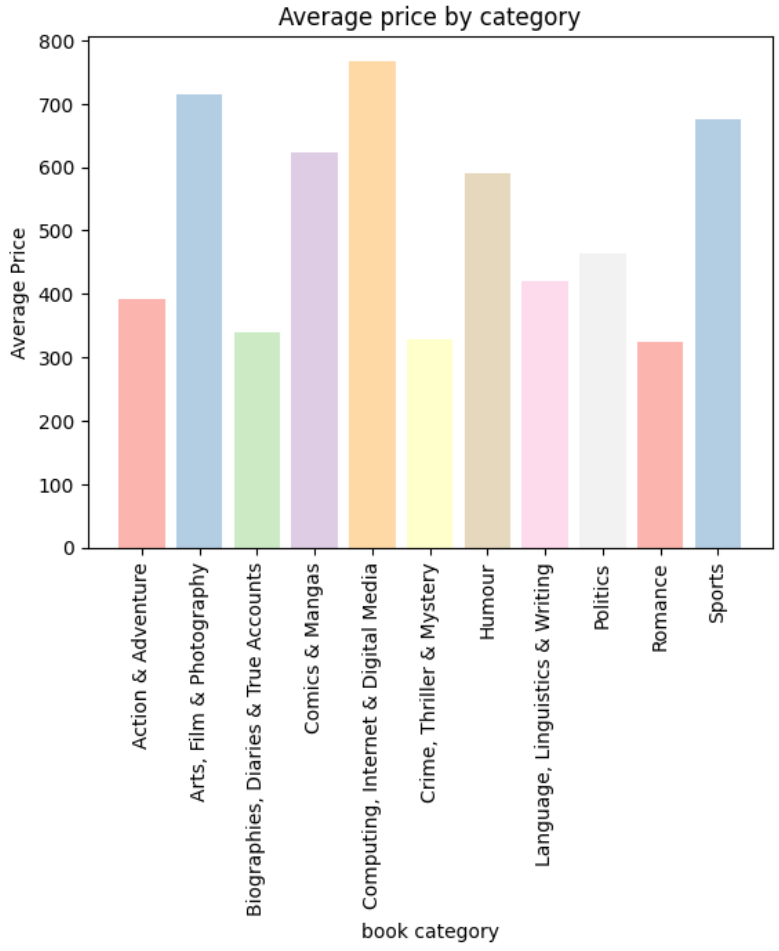


Then, we calculated the top 5 authors with the priciest books which yielded this:

```
Author: Earl Woods     Max Price: 2571.96

Author: Wizards RPG Team     Max Price: 2570.0

Author: Lee Falk Max Price: 2564.0

Author: Sjoukje Zaal   Max Price: 2559.0

Author: Antonio Mele   Max Price: 2559.0
```

We also calculated the average price of books in each category:

Action & Adventure	391.587007
Arts, Film & Photography	714.434989
Biographies, Diaries & True Accounts	339.150018
Comics & Mangas	622.796457
Computing, Internet & Digital Media	767.901991
Crime, Thriller & Mystery	327.883914
Humour	589.525372
Language, Linguistics & Writing	420.742102
Politics	463.727629
Romance	323.626693
Sports	676.318301



This plot shows that the priciest books are ‘computing, internet and digital media’, and then ‘arts, film and photography’.

Feature transformation:

Scaling refers to the process of transforming the values of numerical features to a specific range, often between 0 and 1 or -1 and 1. This is important when features have different scales or units. In our dataset, the price feature can have significantly different value ranges. Scaling this feature can prevent attributes with larger values from dominating or biasing the analysis or model training process. It ensures that the feature contributes proportionately to the analysis and prevents features with larger values from overshadowing others. We scaled this column and created another column called ‘price_scaled’.

Normalization is the process of transforming numerical features to a standard distribution, but since we don’t aim to use an ML technique that requires normality, we don’t use it. Meanwhile, standardization involves transforming numerical features to have zero mean and unit variance. This technique is particularly useful when features have different units or scales, and we need to compare them or use distance-based algorithms. But since we’re not using such algorithms, we skip this method too.

We used one-hot encoding to store the ‘BookCategory’ column just in case we need it later for an ML task. By converting categories into binary features, one-hot encoding preserves the non-ordinal nature of categorical variables and avoids incorrect assumptions about the relationships between categories. This encoding method enables algorithms to process and analyze categorical data, facilitating the discovery of patterns and relationships that may exist within the data.

2. Feature creation

Feature engineering focuses on creating new features that capture relevant information, patterns, or domain knowledge, while feature transformation involves applying mathematical or statistical operations to the existing features to change their representation or distribution.

First of all, we transformed the ‘Ratings’ and ‘Reviews’ columns in a way that they would only contain the numbers that matter to us.

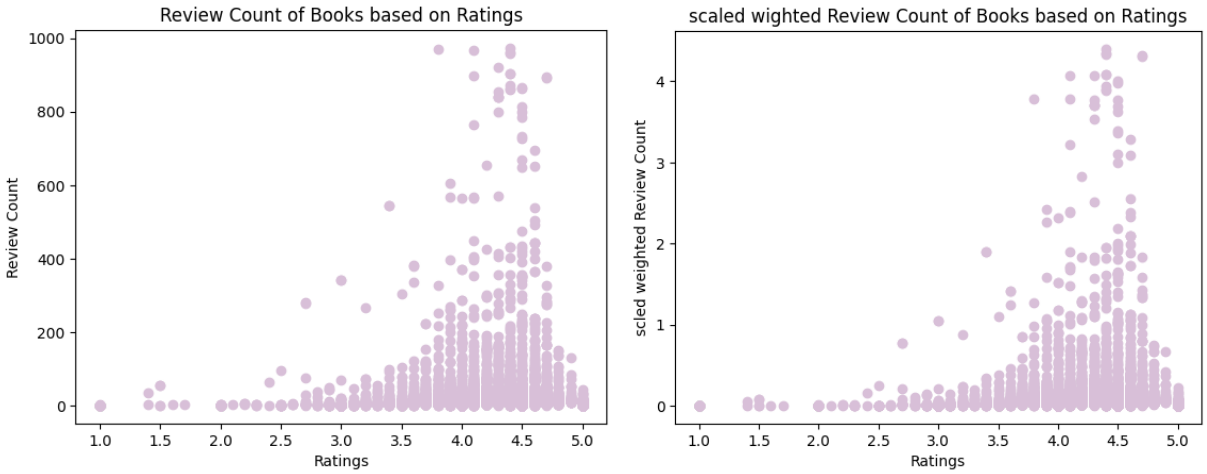
Then, we switched the names of these columns to match their content:

	Ratings	Reviews
0	4.0	8
1	3.9	14
2	4.8	6
3	4.1	13
4	5.0	1

Then, we converted the ‘Edition’ column into two separate columns, one containing the edition and one the date in DateTime format.

Polynomial feature creation involves creating new features by performing mathematical operations, such as multiplication and exponentiation, on existing features. This can capture non-linear relationships between variables and enable the model to learn more complex patterns. But here, we don’t have a numerical column that is appropriate to be subject to polynomial feature creation, so we skip this method.

Interaction features are created by combining two or more existing features. This is particularly useful when there are interactions between variables. Interaction features can help the model capture complex relationships that are not apparent in the individual features. Here, we realized that some books have high ‘Ratings’ (stars) but very little ‘Reviews’. So, we multiplied the scaled ‘Review’ amount by the ratings to get a better sense of the data:



However, the distribution of data doesn’t seem to have changed that much.

We then create a new feature that is the product of ‘ScaledReviews’, ‘scaledRatings’, and (1-‘scaledPrice’), and call it ‘overallQual’ and scale it:

Domain-specific feature engineering involves creating new features based on domain knowledge or insights. This requires a deep understanding of the problem domain and the underlying data. For example, in natural language processing (NLP), domain-specific features could include word counts, n-grams, sentiment scores, or part-of-speech tags. Here, we calculated three new features:

	TitleWordCount	SynopsisAvgWordLength	SynopsisLength
0	6	5.038168	131
1	7	5.267760	183
2	3	5.587302	252
3	7	4.666667	75
4	6	5.346154	104

And some new features about the authors:

	AuthorExperience	PublicationYear	AuthorTotalBooks
0	6.0	2016.0	4
1	0.0	2012.0	1
2	0.0	1982.0	2
3	9.0	2017.0	59
4	0.0	2006.0	1

And some features about the publication date:

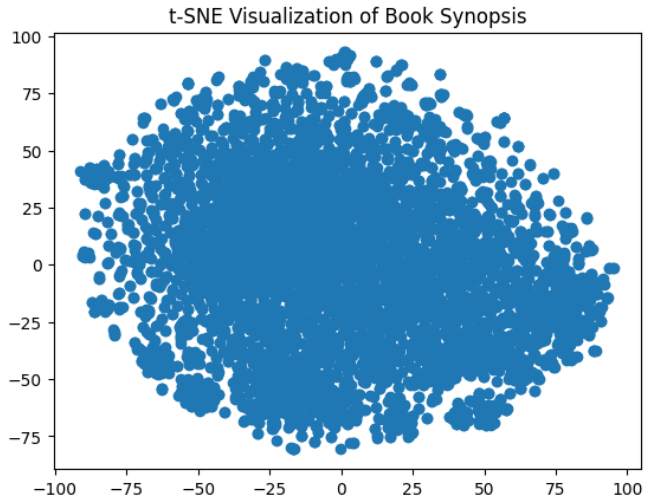
	PublicationMonth	YearsSincePublication	IsSpring	IsSummer	IsFall	IsWinter
0	3.0	7.0	1	0	0	0
1	11.0	11.0	0	0	1	0
2	2.0	41.0	0	0	0	1
3	10.0	6.0	0	0	1	0
4	10.0	17.0	0	0	1	0

The Bag-of-Words (BoW) is a common technique for text feature extraction. It represents text documents as vectors where each feature corresponds to a word or term in the text corpus.

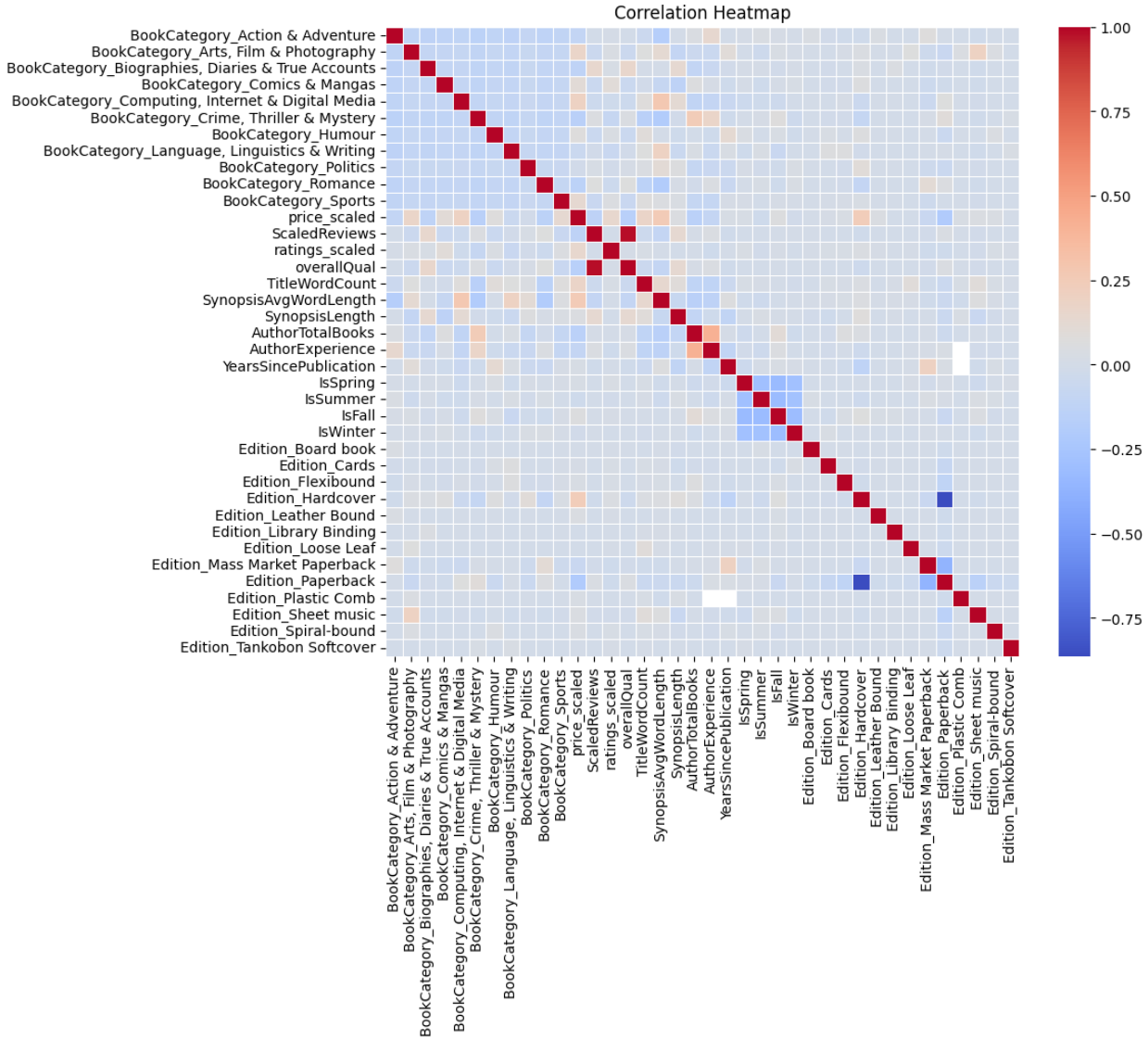
The value of each feature represents the frequency of the corresponding word in the document. We applied the BoW method on the ‘Title’ and ‘Synopsis’ columns and it resulted in a dataframe with 14414 columns.

We then checked the unique values in the ‘Edition’ column and realized that there were some values that only occurred once: ‘(Kannada),Paperback', '(German),Paperback', '(French),Paperback', and 'Perfect Paperback'. All of these can be thought of as ‘paperback’ alone. So, when we are later one-hot encoding the ‘Edition’ column, we map them to the ‘Paperback’ encoding.

After this, we use tSNE to extract some numbers out of the ‘Synopsis’ feature and show each of those with a two-dimensional vector. We add the x and t values of the vectors as another feature of the dataframe.



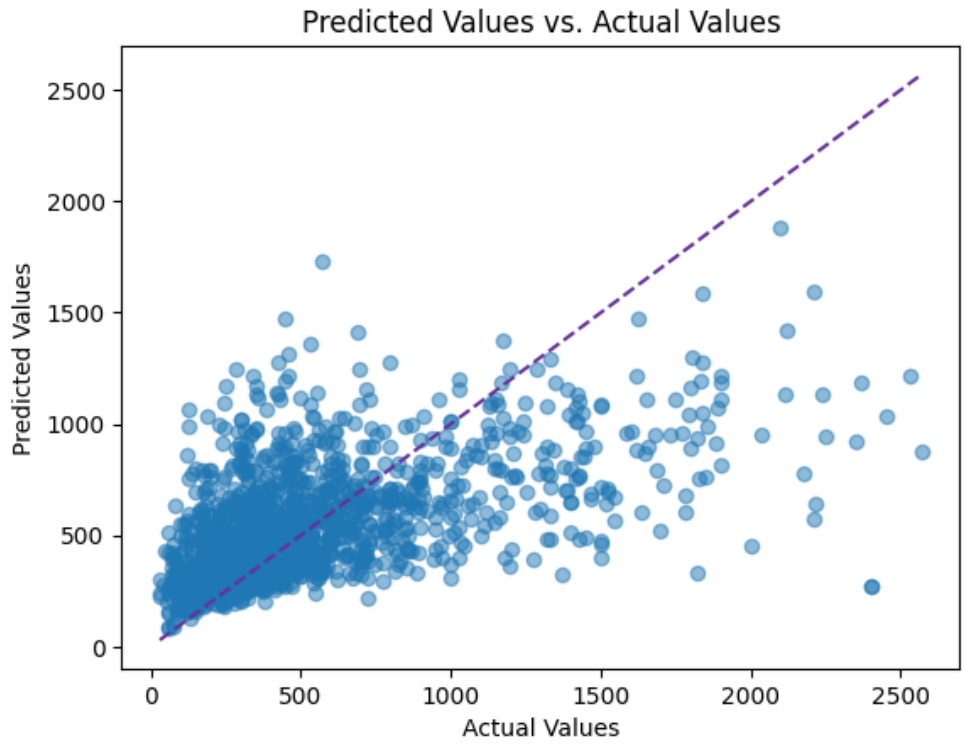
We also drew the heatmap of a correlation matrix between some one-hot encoded categorical variables and some of numerical variables, but couldn’t find any meaningful high relationship there:



We also encoded the ‘Genre’ column using label encoding. That is because the unique values in this column were more than 300 and it would add the dimension of our dataframe too much.

3. Model training

We finally train the random forest regressor model using all those features that we had extracted to predict the price of books. But unfortunately, the loss is very high.



This plot shows how important te features are in the model's prediction: