Feature Engineering

Ali Bakhshesh

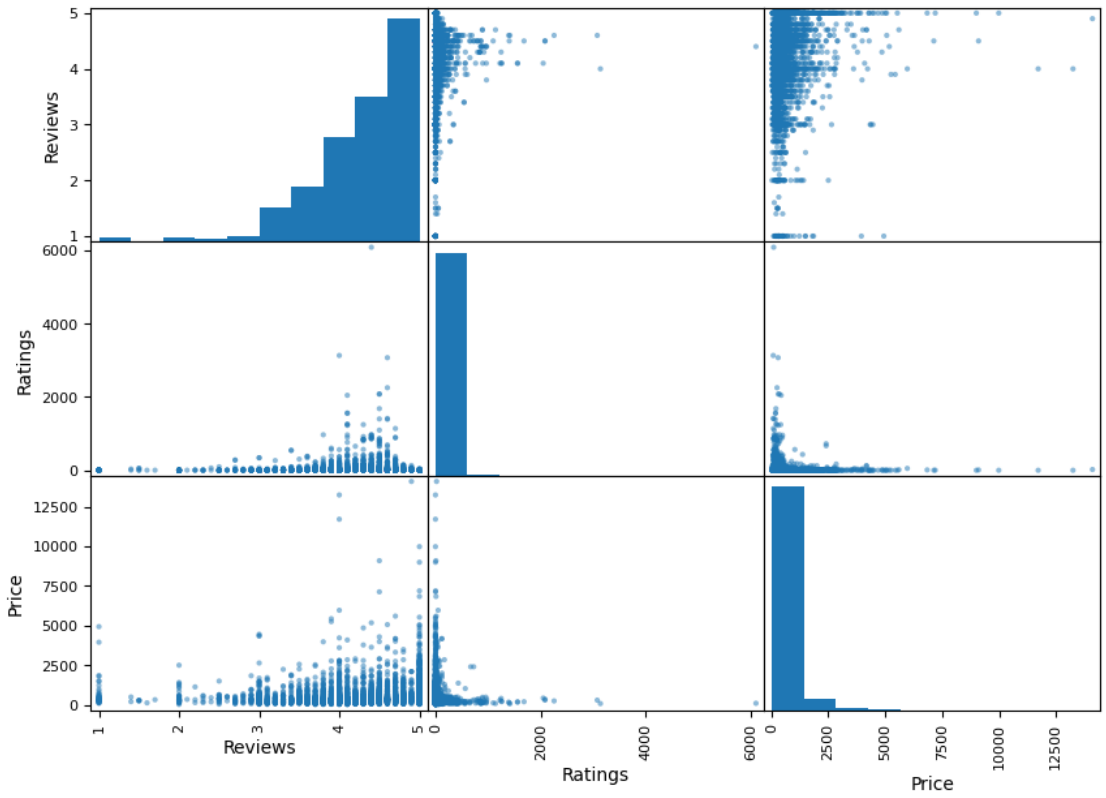
Data Exploration:

The dataset is about books. It include some attributes like review and summery of book and we should use and change these features in such a way that we can predict the price of each book using a *Random Forest Regressor* model. The exact list of the features of the dataset is as mentioned below:

1. Title
2. Author
3. Reviews
4. Ratings
5. Edition
6. Synopsis(abbreviation)
7. Genre
8. BookCategory
9. Price

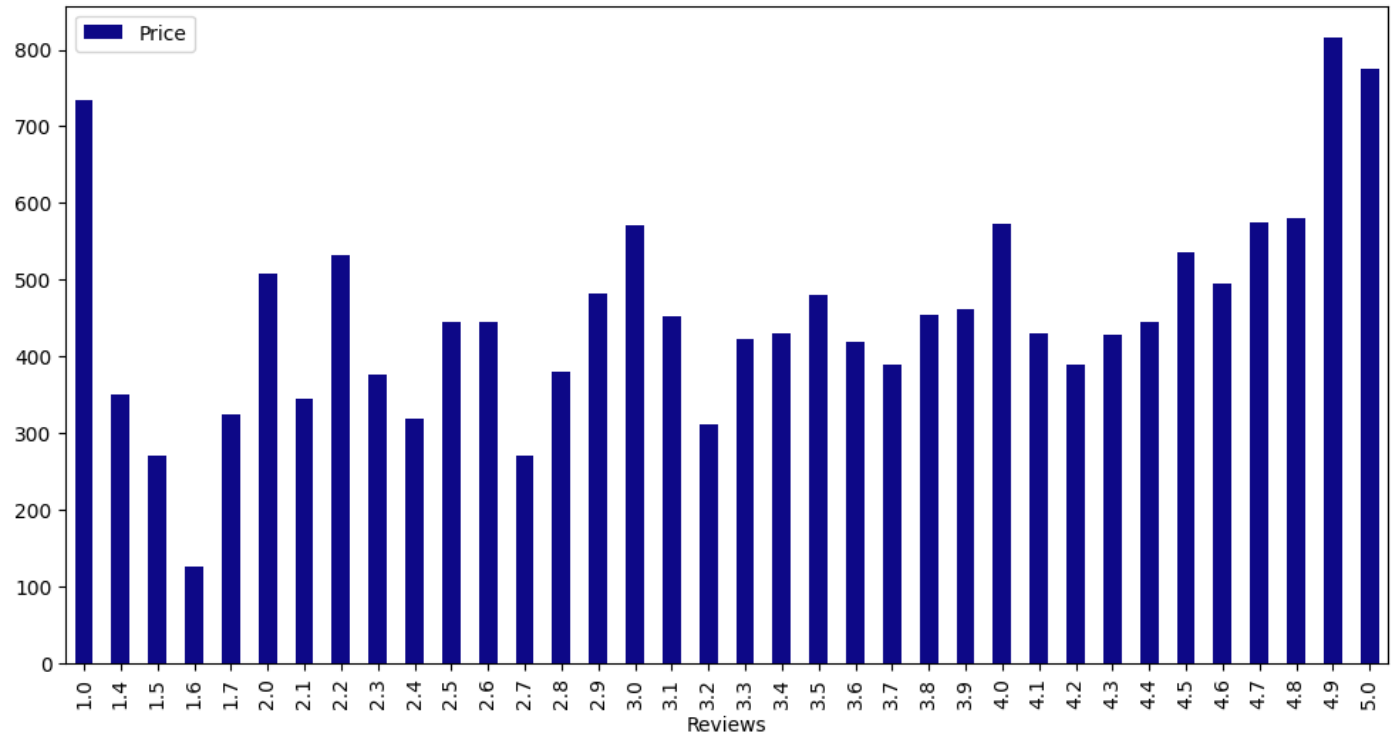
All the features except price are object so at first, we should change them. We started with column Reviews and Ratings, and we could convert them to integer.

By plotting the scatter matrix of the numerical data. We expected that Rating and Reviews have a high effect on the price.

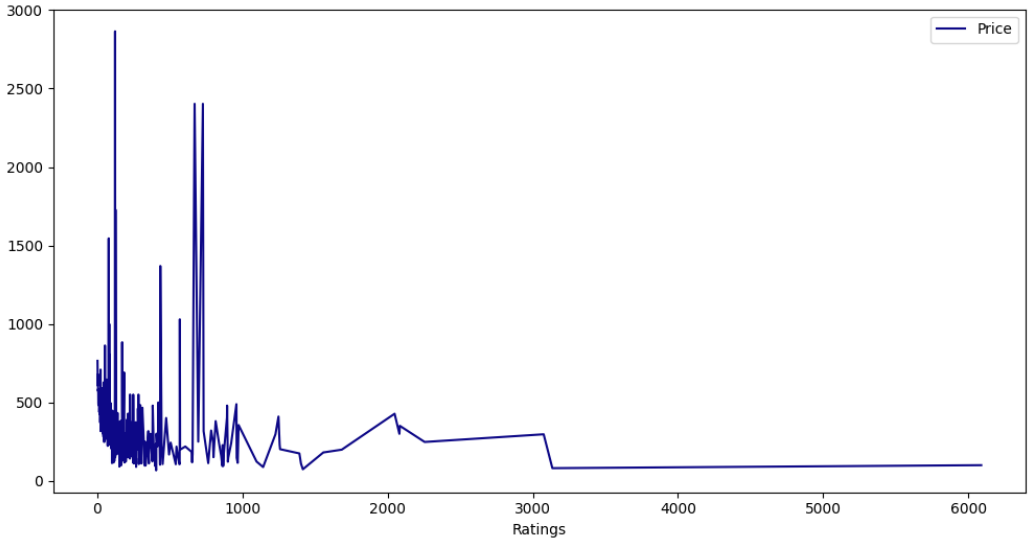


By looking at this plot we see that Reviews has a little effect on Price but there is no effect from Ratings.

Following plot shows the average price of each point (point is in range 1 to 5):

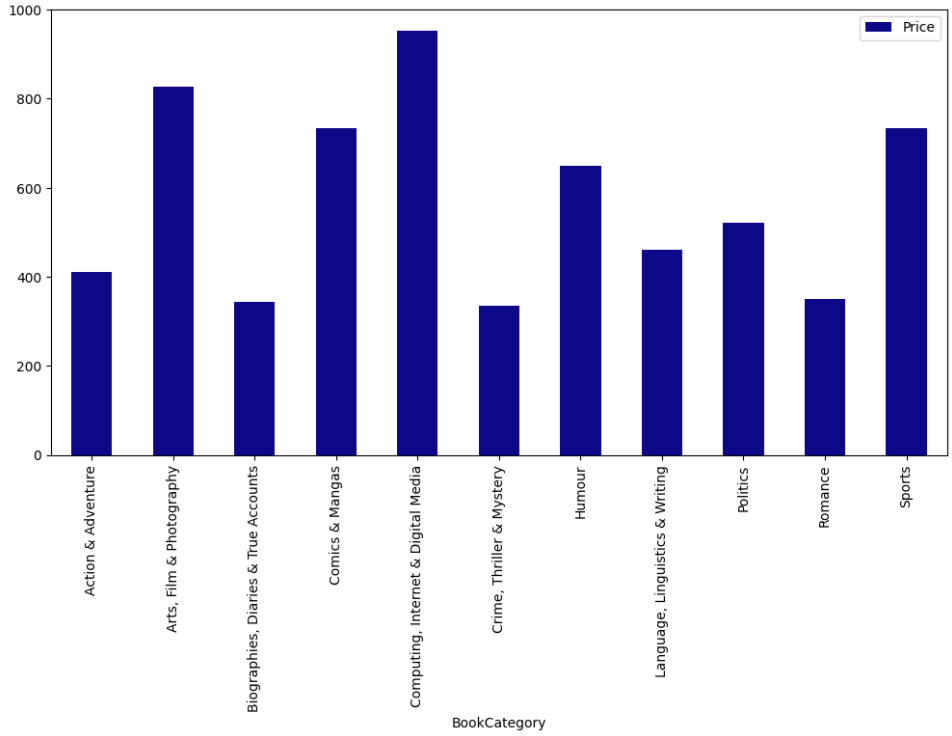


The effect is obvious, but it is not significant.



This plot also shows the relation between Rating and price. As we can see there is no significant co-relation, but it seems that books with more reviews have less price.

We also examined the relation between book category and its price:



In the following we created a new feature of Edition called Type.

After this step first, we separated the train and test data.

From this step on we changed test and train data separately and simultaneously. First, we concatenated following features:

Title – Author – Genre – Book category – Type – Synopsis

And we made a new column called New_Synopsis.

Already there is a problem because this new feature is string, and it is impossible to give it as input to the model, so we changed it.

We have used embedding technique called *tf-idf.* In this algorithm the frequency of each word in each row and this is based on following formula:

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

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

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

Using this method, we converted this column of each row to a vector of numbers. Now we could give the data to the model but there is also a new problem. The dimension of the output matrix of tf-idf algorithm is so high. It has approximately 36000 columns and clearly it is impossible to train a model on such a data. So, to solve this problem we used *the*

PCA algorithm and decreased the dimension. We used PCA algorithm with 1024 as input (means that 1024 features we need).

We also had done it before with smaller numbers, but the results were not satisfying. After doing all these steps we trained the model and tested it either. The MSE for both train and test phase are respectively 60426.59 and 300967.59.