Book Rating Prediction Model
DSTI : Python Labs

Mayank Bhandari
Khalil Bensaid
Ramesh Singh

March 25, 2023

1 Abstract

The Goodreads database has about 1123 books with various features such as publisher, book title, average ratings, published date, and rating counts. Given this large dataset, the question is whether it is possible to predict a book's average rating based on its features. In this project, we attempt to answer this question using machine learning algorithms provided by scikit-learn.

2 Data Cleaning

- Pulled out the published year of a book from the published date and created a new column called published year in the dataframe.
- Calculated the age of each book by subtracting the publication year by 2023 (the current year we are in).
- Dropped ISBN and ISBN12 column.
- Changed language codes with their full name and merged English-based ones (e.g. en-ca, en-us,etc) to English.

3 Exploratory Data Analysis

- Created bar graphs, Box plots, Scatter plots, and Heat maps for numerical variables.
- Created bar graphs to check the ratings of books for a specific language.
- Created a bar graph to check which author received the maximum count of ratings.
- Created a graph to check which author received the maximum ratings.
- Created graphs that show the relationship between a numerical column and the target column.

4 Feature Selection

The Features we selected for predicting the ratings of the books are mentioned below:

- Author
- language Code
- Num Pages
- Ratings Count
- Text Review Counts

5 Model Building

For model building, we have transformed the categorical variables. The 'Authors', 'Title', and 'Publisher' variables were encoded, and the 'Language' variable was transformed into dummy variables.

5.1 Algorithms used

- Linear Regression
- Linear Regression with Cross-Validation
- Decision Tree Regressor
- Gradient Boosting Regressor
- KNeighborsRegressor
- Random Forest Regressor
- Random Forest Regressor with Cross-Validation
- XGB Regressor

5.2 Model Performance Evaluation

for the model evaluation we are using MSE and RMSE matrices.

Algorithm	MSE Train	MSE Test	$R^2 Train$	$R^2 Test$
Linear Regression	6	87837	787	11
Linear Regression with Cross-Validation	0.119	0.124	0.041	0.028
Decision Tree Regressor	0.0	0.189	1.00	-0.490
Gradient Boosting Regressor	0.80	0.110	0.354	0.134
KNeighborsRegressor	0.088	0.139	0.289	-0.094
Random Forest Regressor	0.014	0.103	0.889	0.192
XGB Regressor	0.030	0.109	0.761	0.146

6 Conclusion

From the observations, we see that a Random Forest model performed the best on the test data. However, the overall scores are not very good, but the displayed actual versus predicted data shows some satisfying results. A sample of the random table looks like this:

Actual	Predicted
4.1	3.9
3.8	3.9
3.30	3.92
4.49	4.07
3.43	3.88
3.92	3.941
3.77	3.94