**Report**

**Link to Git :**

<https://github.com/YGueguen16u/Book-Rating-Prediction-Model>

The abundance of books available makes it increasingly difficult for readers to choose the best titles. Leveraging a dataset from Goodreads based on real user information, this project aims to create a model that can predict a book's rating, making it easier for users to select books that they are likely to enjoy.

Goals :

The primary objective of this project is to train a model that accurately predicts a book's rating based on the dataset provided. The project is expected to include:

* Data Analysis and Cleaning
* Exploratory Data Analysis (EDA)
* Feature Engineering and Selection
* Model Training and Evaluation
* Deployment

To achieve the goals stated above, the project is broken down into the following specific objectives:

1. **Import Data**: Load the dataset into a DataFrame for analysis.
2. **Data Analysis and Data Cleaning**:
   * Handle errors in the 'Author's' column.
   * Define 'book\_id' as the primary key.
   * Deal with missing data.
   * Analyze publication dates.
3. **Feature Engineering**:
   * Create 'authors\_list' and 'number of authors' columns.
   * Consolidate language codes (e.g., 'eng', 'eng-US', 'eng-GB').
   * Verify that all average ratings are within the 0-5 range.
   * Confirm that there are no aberrant values in the number of pages and that rating counts and review counts are positive.
4. **Model Training**: Select and train machine learning models for rating prediction.
5. **Model Evaluation**: Utilize metrics like MSE, RMSE, MAE, and R^2 Score to evaluate the performance of the trained models.

**Summary and Model Ranking for Inclusion in the Report**

In the Jupyter Notebook, various data processing and analysis steps were performed on a book dataset. These included:

1. Cleaning the data by fixing errors in author names and publication dates.
2. Analyzing various columns like the number of authors, average rating, and number of pages.
3. Ensuring data integrity by removing aberrant values and checking for outliers.

The dataset was then used to evaluate the performance of different regression models: Linear Regression, Random Forest, AdaBoost, and Decision Tree.

**Model Ranking Based on Performance Metrics**

The evaluation metrics used to assess the models were Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R^2 Score, Max Error, and Explained Variance Score. Based on these, the models can be ranked as follows:

1. **Linear Regression**
   * MSE: 0.101
   * R^2 Score: 0.032
   * Strongest performer in terms of MSE and Explained Variance Score.
2. **Random Forest**
   * MSE: 0.105
   * R^2 Score: -0.005
   * Better MAE compared to other models but slightly worse on MSE.
3. **AdaBoost**
   * MSE: 0.115
   * R^2 Score: -0.099
   * Moderately performing, with the largest Max Error among the considered models.
4. **Decision Tree**
   * MSE: 0.226
   * R^2 Score: -1.162
   * Worst performer, likely overfitting the data.

**Key Takeaways**

* Linear Regression provided the best MSE but suffered from a low R^2 score, indicating that it may not be capturing the complexity of the data.
* Random Forest performed slightly worse in terms of MSE but had a better MAE.
* AdaBoost had moderate performance but was outperformed by both Linear Regression and Random Forest.
* Decision Trees performed the poorest, suggesting possible overfitting.

Selecting the right model depends on the specific requirements of the task, but based on these metrics, Linear Regression and Random Forest appear to be the most promising candidates.