Hnin Nandar Win
Antoine Teillac
Emilie Schmitter

Python Project - Goodreads Report

I. Approach

Before starting the project, we analyzed and cleaned the dataset provided to optimize the model training subsequently.

We then went through an EDA (Exploratory Data Analysis) to compare how books in the database were reviewed and which factors could influence the ratings.

We decided to compare six factors:

- Distribution of average ratings
- Number of pages vs average ratings
- Rating counts vs average ratings
- Rating per page vs average rating
- Text Reviews to Ratings Ratio vs. Average Rating
- Number of Pages vs. Ratings Count

After comparing all of the above, we decided to train our model using the most relevant ones.

1. Data Cleaning and Transformation

1. **Handling Missing Data**:
    - Rows with missing values in important columns such as authors, language_code, and average_rating were either imputed where possible or removed if they were insufficient for model training.
    - The Unnamed: 12 column was removed entirely as it contained improperly formatted data such as comma separators.

    Converting Data Types:

    - The average_rating and num_pages columns, which were originally stored as strings, were converted to numerical data types (float64 and int32, respectively).
2. **Dropping Irrelevant Columns**:
    - The isbn and isbn13 columns, which serve as unique book identifiers, were dropped as they provide no predictive value for the target variable average_rating.

1.2 Categorical Data Transformation

- Authors, Title, Publisher: These columns contained too many unique values (high cardinality). The authors column was split into three columns: author1_encoded, author2_encoded, and author3+_encoded to account for books with multiple authors. **Target Encoding** was then applied to convert these categorical values into numerical ones based on their relationship with average_rating.

- **Language Code**: The language_code column, containing categories such as eng, fre, and spa, was transformed using **One-Hot Encoding (OHE)**. This created new binary columns (language_code_eng, language_code_fre, etc.) to represent the presence of each language.

1.3 Final Cleaned Data Overview

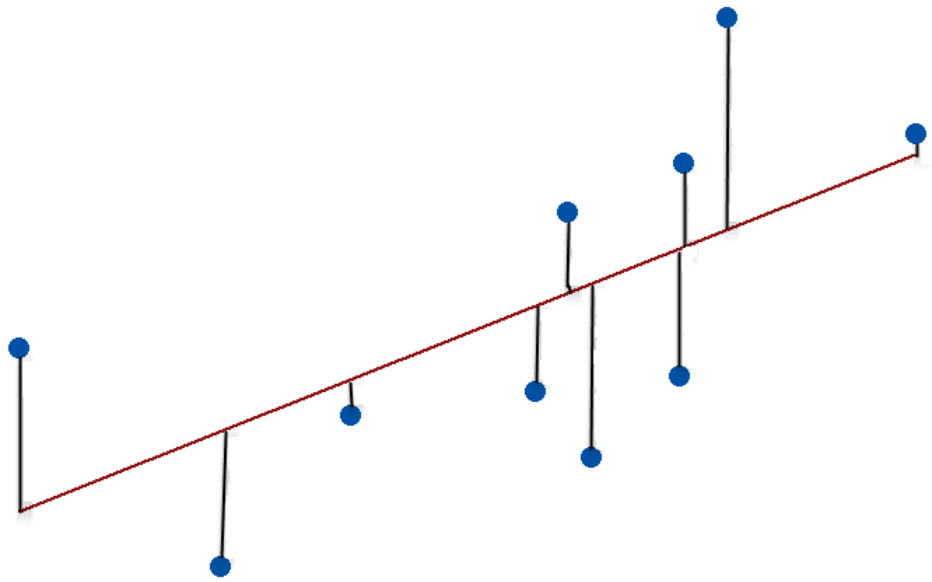
The cleaned dataset contains 17 columns, all in numeric format, ready for use with the Random Forest Regressor:

- **Numerical columns**: num_pages, ratings_count, text_reviews_count, pub_year, pub_month, and pub_day.
- **Encoded categorical features**: author1_encoded, author2_encoded, author3+_encoded, title and publisher_encoded.
- **One-Hot Encoded Language Features**: language_code_en-GB, language_code_en-US, language_code_eng, language_code_fre, and language_code_spa.

All features are now numeric, with appropriate encoding applied to categorical features, ensuring compatibility with the Random Forest Regressor.

II. Results

As a result, we found an MSE (Mean Square Error) of 0.0002878895023041464 (the mean value of the square difference between the predicted results and the "results" we found).



The points are the true results, and the line represents the predicted results.

To find these results, we used the “random forest regressor” method, an estimator that fits a number of decision tree regressors on various sub-samples of the dataset and uses averaging to improve predictive accuracy and control overfitting.

```
[ ]  #séparation pour le train et le test
     X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.9, test_size=.1, random_state=42)
```

```
[ ]  # En utilisant le modèle de regression forêt aléatoire
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.metrics import mean_squared_error


     model = RandomForestRegressor()
     model.fit(X_train, y_train)
     y_pred = model.predict(X_test)

     # Calcul du MSE
     mse = mean_squared_error(y_test, y_pred)
     print(f"MSE: {mse}")
```

```
MSE: 0.0002878895023041464
```