Data Analytics Project 2: Data Analysis and Recommendation System for Automotive Parts(N.04)

Aouina Chadha (aouina.chadha@usi.ch)
Vitaliy Nikitin (vitaliy.nikitin@usi.ch)

2024.04.16

1 Introduction

The goal of this project is to predict the ratings of automotive parts for users based on other users' ratings. The provided dataset contains a heterogeneous set of features about user preferences for automotive parts, such as user, item, rating, and timestamp tuples. The metadata includes descriptions, price, sales-rank, brand information, and co-purchasing links. This project aims to use Python and complementary libraries to explore and analyze the given data and draw meaningful conclusions.

2 Data Cleaning

2.1 Fixing JSON Format

The first step involved fixing the JSON format of the metadata file. The provided JSON file was formatted in a way that required splitting and correcting individual JSON objects. The `fix_json` function was used to read the JSON file, correct the formatting issues, and save it to a new file.

2.2 Formatting JSON Objects

After fixing the JSON format, the next step was to ensure each JSON object was correctly formatted for further processing. This was done using the `format_json_objects` function, which wrote each JSON object to a new line in the output file.

2.3 Loading Data

The cleaned and formatted JSON metadata was loaded into a pandas DataFrame. Similarly, the ratings data was loaded from a CSV file. The two datasets were merged on the item ID to create a comprehensive dataset containing both user ratings and product metadata.

3 Data Exploration

3.1 Merging Datasets

The merged dataset included user IDs, item IDs, ratings, timestamps, and various product metadata such as categories, descriptions, titles, prices, image URLs, brands, related products, and sales ranks.

3.2 Checking for Missing Values

We performed an analysis to identify missing values in the dataset. The results showed significant missing values in several metadata fields:

- **Categories**: 79.23% missing
- **Description**: 79.67% missing

- **Title**: 79.93% missing
- **Price**: 81.20% missing
- **Image URL**: 79.25% missing
- **Brand**: 83.70% missing
- **Related Products**: 80.05% missing
- **Sales Rank**: 97.99% missing

3.3 Descriptive Statistics

Descriptive statistics provided an overview of the central tendency, dispersion, and shape of the dataset's numeric variable distributions.

- **Rating**: Mean rating was 4.21 with a standard deviation of 1.28.
- **Timestamp**: Represents the time of rating in Unix time.
- **Price**: Mean price was $39.45 with a wide range and a maximum price of $862.28.

3.4 Data Visualization

3.4.1 Histograms

Histograms were plotted to visualize the distribution of the 'price' and 'rating' variables. The price distribution showed a right-skewed pattern, while the rating distribution was more uniform with most ratings being 4 or 5.

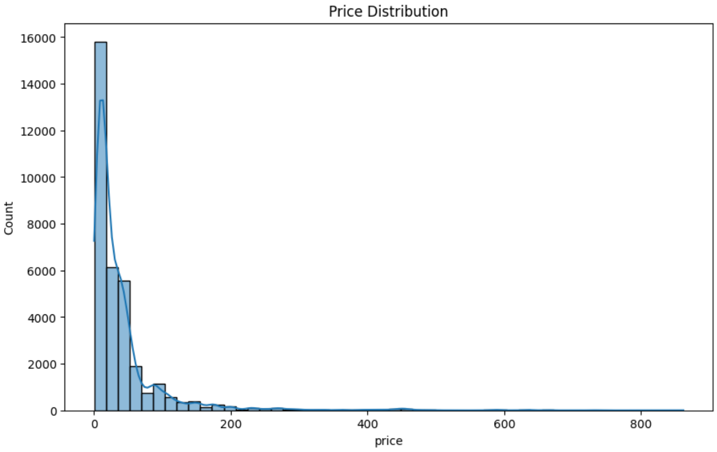

Figure 1: Price Distribution

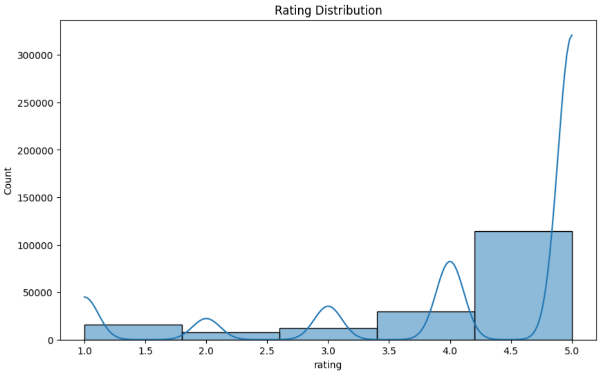

Figure 2: Rating Distribution

3.4.2 Box Plot

A box plot of the 'rating' variable highlighted the presence of outliers and provided a clear visualization of the distribution of ratings.

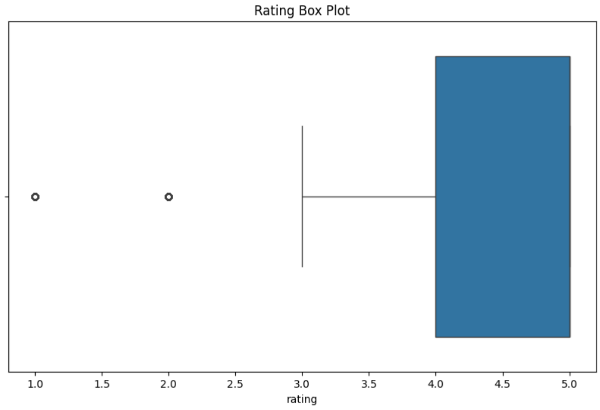

Figure 3: Rating Box Plot

3.4.3 Correlation Matrix

A correlation matrix was calculated and visualized to understand the relationships between numeric variables. The heatmap showed the strength and direction of correlations, which is useful for identifying multicollinearity and interesting variable interactions.

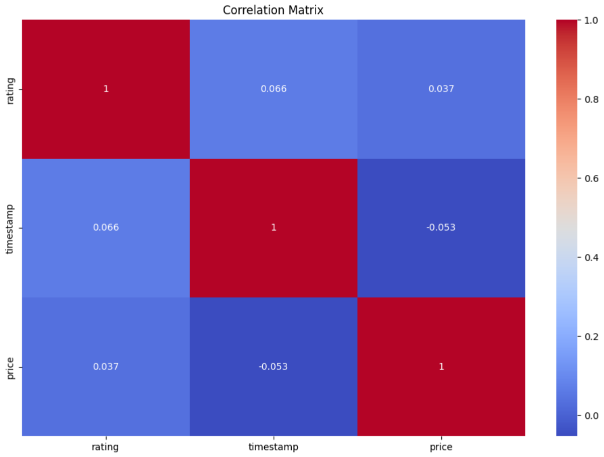

Figure 4: Correlation Matrix

4 Findings

- The dataset contains a significant number of missing values in various metadata fields.
- The ratings are predominantly high, with a mean rating above 4.
- The price variable exhibits a right-skewed distribution.
- The correlation matrix provides insights into potential relationships between numeric variables, although many variables have a low correlation with each other.

5 Pre-processing the data

5.1 Filling Missing Values

Missing values in various columns were handled as follows:

- **Price**: Filled with the mean price.
- **Brand**: Filled with the value 'Unknown'.
- **Description, Title, Image URL, Related**: Filled with appropriate placeholder text such as 'No description', 'No title', 'No image', and 'No related items'.

5.2 Dropping High Missing Value Columns

The `salesRank` column was dropped due to its high percentage of missing values.

5.3 Converting Categories to Strings

The `categories` column, which contained lists of categories, was converted to a single string for each entry. This was done by flattening nested lists and joining them into a space-separated string.

5.4 Encoding Categorical Variables

Categorical variables were encoded as follows:

- **Categories**: Converted to numerical codes.
- **Brand**: Converted to numerical codes.

5.5 Saving Pre-processed Data

The pre-processed data was saved to a new CSV file for future use.

5.6 Check for Missing Values

A final check for missing values was performed to ensure that all missing data had been appropriately handled.

The final dataset was found to be free of missing values in the critical fields necessary for analysis, thus ensuring the dataset was ready for further steps in the project.

6 Prediction Models

In this section, we built two recommendation systems for predicting outdoor automotive parts ratings for users in the test sample. The models include a Random Forest Regressor and a Content-Based Filtering approach.

6.1 Random Forest Regressor

The Random Forest Regressor was used to predict ratings based on a combination of TF-IDF vectorized text features from the title and description of the products.

6.1.1 TF-IDF Vectorization and Dimensionality Reduction

We applied TF-IDF vectorization to the 'title' and 'description' fields separately, each with a maximum of 2500 features. The dimensionality of the resulting TF-IDF matrices was then reduced using Truncated SVD, retaining 50 components for each.

6.1.2 Combining Features and Model Training

The reduced TF-IDF features were combined with the rating data. The dataset was then split into training and testing sets. A Random Forest Regressor with 100 estimators was trained on the training set.

6.1.3 Model Evaluation

The model's performance was evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). The results were:

- **RMSE**: 1.2966
- **MAE**: 1.0001

6.2 Content-Based Filtering

The Content-Based Filtering approach used cosine similarity between items to predict ratings.

6.2.1 Pivot Table and Cosine Similarity

A pivot table was created with items as rows and users as columns, with ratings as values. The table was converted to a sparse matrix, and cosine similarity was calculated between items.

6.2.2 Predicting Ratings

Ratings were predicted based on the similarity scores and the known ratings of similar items. The process was performed in batches to handle large data efficiently.

6.2.3 Model Evaluation

The performance of the Content-Based Filtering model was evaluated using RMSE and MAE. The results were:

- **Content-Based Filtering RMSE**: 0.5241
- **Content-Based Filtering MAE**: 0.2133

7 Conclusion

Both models demonstrated the ability to predict automotive parts ratings with varying degrees of accuracy. The Content-Based Filtering approach showed a lower RMSE and MAE compared to the Random Forest Regressor, indicating better performance for this particular dataset and task. Further optimization and hybrid approaches could be explored to improve prediction accuracy.