Exploratory Data Analysis of Amazon Popular Books using KDD Methodology

September 30, 2023

Abstract

This paper provides a detailed exploration of the Amazon Popular Books dataset using the Knowledge Discovery in Databases (KDD) methodology. The study presents patterns and insights underpinned with visualizations and code snippets.

In this article, we will delve deep into an exploratory data analysis of the Amazon Popular Books datasetusing the Knowledge Discovery in Databases (KDD) methodology. If you're unfamiliar with KDD, it is a structured process involving several steps that guide us from raw data to meaningful insights. The KDD process is generally divided into several steps:

1. Data Selection
2. Data Preprocessing
3. Data Transformation
4. Data Mining
5. Evaluation & Interpretation
6. Deployment

1 Phase 1. Data Selection

The first step in any analysis is to choose the right dataset. For this exercise, we used the Amazon Popular Books dataset, containing various attributes like book ratings, price, categories, and more. A quick glance at the dataset provided a sense of the available columns and the type of information we could extract. This step involves selecting the dataset of interest. let's first load and take a look at the initial few records.

Let's start by loading the dataset and examining the first few rows.

```
import pandas as pd

# Load the dataset
data = pd.read_csv('/mnt/data/amazon_popular_books.csv')

# Display the first few rows of the dataset
data.head()
```

RESULT

```
         asin         ISBN10  answered_questions availability  \
0  0007350813     0007350813                   0    In Stock.
1  0007513763  9780007513765                   0    In Stock.
2  0008183988     0008183988                   0          NaN
3  0008305838     0008305838                   0    In Stock.
4  0008375526     0008375526                   0    In Stock.

              brand currency  date_first_available  \
0      Emily Brontë      USD                   NaN
1      Drew Daywalt      USD                   NaN
2  Bernard Cornwell      USD                   NaN
3    David Walliams      USD                   NaN
4   Caroline Hirons      USD                   NaN

                                             delivery  department  description  \
0  ["FREE delivery Tuesday, December 28 if you sp...         NaN          NaN
1  ["FREE delivery Tuesday, December 28 if you sp...         NaN          NaN
2  ["FREE delivery January 4 - 10 if you spend $2...         NaN          NaN
3  ["FREE delivery Tuesday, December 28 if you sp...         NaN          NaN
4  ["FREE delivery Tuesday, December 28","Or fast...         NaN          NaN

   ...  upc                                    url  video  video_count  \
0  ...  NaN  https://www.amazon.com/dp/0007350813    NaN            0
1  ...  NaN  https://www.amazon.com/dp/0007513763    NaN            0
2  ...  NaN  https://www.amazon.com/dp/0008183988    NaN            0
3  ...  NaN  https://www.amazon.com/dp/0008305838    NaN            0
4  ...  NaN  https://www.amazon.com/dp/0008375526    NaN            0

                                          categories  \
0     ["Books","Literature & Fiction","Genre Fiction"]
1  ["Books","Children's Books","Literature & Fict...
2     ["Books","Literature & Fiction","Genre Fiction"]
3  ["Books","Children's Books","Literature & Fict...
4  ["Books","Crafts, Hobbies & Home","Home Improv...

                                    best_sellers_rank          buybox_seller  \
0  [{"category":"Books / Literature & Fiction / H...                    NaN
1  [{"category":"Books / Children's Books / Liter...     VMG Books & Media
2  [{"category":"Books / Literature & Fiction / H...           Reuseaworld
3  [{"category":"Books / Children's Books / Liter...         Bahamut Media
4  [{"category":"Books / Health, Fitness & Dietin...  KathrynAshleyGallery

   image  number_of_sellers  colors
0    NaN                NaN     NaN
1    NaN                NaN     NaN
```

```
2    NaN              NaN    NaN
3    NaN              NaN    NaN
4    NaN              NaN    NaN

[5 rows x 40 columns]
```

The dataset contains various attributes related to popular books on Amazon. Here are a few columns observed in the dataset:

- asin: Amazon Standard Identification Number
- ISBN10: International Standard Book Number (10-digit)
- answered_questions: Number of answered questions
- availability: Stock availability
- brand: Brand/Author name
- currency: The currency in which the price is listed
- date_first_available: Date when the book first became available
- delivery: Delivery details
- categories: Categories to which the book belongs
- best_sellers_rank: Best sellers rank of the book
- buybox_seller: Seller details
- image: Image URL
- number_of_sellers: Number of sellers selling the book
- colors: Color options, if any (probably not relevant for books)

. . . and many more.

2 Phase 2. Data Preprocessing

In the preprocessing phase, we will:

1. Handle missing values.
2. Identify and handle outliers.
3. Convert data types if necessary.
4. Handle any other data quality issues.

2.1 Handling Missing Values

The first step in preprocessing is to identify and handle missing values. We'll start by checking for any missing values in the dataset.

```
# Check for missing values in each column
missing_values = amazon_books_df.isnull().sum()

# Percentage of missing values
missing_percentage = (missing_values / len(amazon_books_df)) * 100

# Combine the data into a dataframe for a better view
missing_df = pd.DataFrame({'Missing Values': missing_values, 'Percentage (%)': missing_percentage})

# Display columns with missing values
missing_df[missing_df['Missing Values'] > 0].sort_values(by='Percentage (%)', ascending=False)
```

RESULT

	Missing Values	Percentage (%)
department	2269	100.000000
upc	2269	100.000000
video	2268	99.955928
colors	2267	99.911855
date_first_available	2266	99.867783
model_number	2265	99.823711
number_of_sellers	2265	99.823711
image	2265	99.823711
manufacturer	2265	99.823711
description	1557	68.620538
plus_content	1552	68.400176
buybox_seller	1218	53.680035
initial_price	1092	48.126928
discount	1092	48.126928
availability	894	39.400617
final_price	877	38.651388
seller_id	864	38.078449
product_dimensions	862	37.990304
image_url	844	37.197003
item_weight	841	37.064786
ISBN10	840	37.020714
format	96	4.230939
root_bs_rank	8	0.352578
best_sellers_rank	1	0.044072
brand	1	0.044072

Here's a summary of columns with missing values:

- Some columns, such as department and upc, have 100% missing values and provide no information. We'll drop these columns.
- Columns like video, colors, date_first_available, and a few others have a very high percentage of missing values. We need to decide whether to keep or drop them based on their relevance to the analysis.
- Several columns have a moderate percentage of missing values. For instance, description has about 68.62% missing values.

- There are a few columns with a relatively low percentage of missing values.

To handle these:

1. We'll drop columns with more than 90% missing values as they are unlikely to provide significant insights.
2. For categorical columns with missing values, we can replace the missing values with a placeholder like "Unknown" or use the mode (most frequent value) of that column.
3. For numerical columns with missing values, we can replace the missing values with the median or mean of that column.

Let's start by dropping columns with more than 90% missing values.

```
# Drop columns with more than 90% missing values
columns_to_drop = missing_df[missing_df['Percentage (%)'] > 90].index
amazon_books_df = amazon_books_df.drop(columns=columns_to_drop)

# Display the shape of the dataframe after dropping columns
amazon_books_df.shape
RESULT
(2269, 31)
```

After dropping columns with more than 90% missing values, our dataset now consists of 31 columns.

Next, we'll handle the missing values in the remaining columns:

1. For categorical columns, we'll replace missing values with the placeholder "Unknown".
2. For numerical columns, we'll replace missing values with the median of that column (since the median is less sensitive to outliers).

Let's proceed with this strategy.

```
# Identify categorical and numerical columns
categorical_columns = amazon_books_df.select_dtypes(include=['object']).columns
numerical_columns = amazon_books_df.select_dtypes(exclude=['object']).columns

# Replace missing values
forcolumn incategorical_columns:
  amazon_books_df[column].fillna('Unknown', inplace=True)

forcolumn innumerical_columns:
  median_value = amazon_books_df[column].median()
  amazon_books_df[column].fillna(median_value, inplace=True)

# Check if there are any missing values left
remaining_missing = amazon_books_df.isnull().sum().sum()
remaining_missing
RESULT
0
```

All missing values have been addressed, and the dataset no longer contains any null values.

2.2 Identifying and Handling Outliers

Outliers can skew the results of our analysis, so it's essential to identify and handle them appropriately. One common method to identify outliers is by using the Interquartile Range (IQR). We'll use this method to detect outliers for numerical columns and decide how to handle them.
Let's proceed with identifying outliers using the IQR method.

```
# Function to detect outliers using IQR
defdetect_outliers(dataframe, column):
   Q1 = dataframe[column].quantile(0.25)
   Q3 = dataframe[column].quantile(0.75)
   IQR = Q3 - Q1
   lower_bound = Q1 - 1.5* IQR
   upper_bound = Q3 + 1.5* IQR

   returndataframe[(dataframe[column] < lower_bound) | (dataframe[column] > upper_bound)]

# Detect outliers for numerical columns
outliers_data = {}
forcolumn innumerical_columns:
   outliers = detect_outliers(amazon_books_df, column)
   outliers_data[column] = len(outliers)

# Display the count of outliers for each numerical column
outliers_data
```

RESULT

```
{'answered_questions': 4,
'discount': 991,
'final_price': 369,
'images_count': 92,
'initial_price': 1071,
'plus_content': 0,
'reviews_count': 191,
'root_bs_rank': 308,
'video_count': 1}
```

Here's a summary of the detected outliers in numerical columns:

- answered_questions: 4 outliers
- discount: 991 outliers
- final_price: 369 outliers
- images_count: 92 outliers
- initial_price: 1071 outliers
- reviews_count: 191 outliers
- root_bs_rank: 308 outliers
- video_count: 1 outlier

Given the nature of this dataset, the presence of outliers in price-related columns (e.g., discount, final_price, initial_price) might be due to some books being more expensive or having significant discounts. Similarly, outliers in columns like reviews_count could indicate very popular books.

For the purpose of this exploratory data analysis, we won't remove these outliers since they can provide insights into the variability of book prices, discounts, and popularity on Amazon. However, if we were building a predictive model, we might handle these outliers differently.

2.3 Convert Data Types (if necessary)

We'll check the data types of the columns to ensure they are appropriate for analysis. If not, we'll convert them to the correct type.

Let's proceed by checking the data types.

```
# Check data types of the columns
data_types = amazon_books_df.dtypes
data_types
```

```
RESULT
asin                  object
ISBN10                object
answered_questions     int64
availability          object
brand                 object
currency              object
delivery              object
description           object
discount             float64
domain                object
features              object
final_price          float64
format                object
image_url             object
images_count           int64
initial_price        float64
item_weight           object
plus_content         float64
product_dimensions    object
rating                object
reviews_count          int64
root_bs_rank         float64
seller_id             object
seller_name           object
timestamp             object
title                 object
url                   object
video_count            int64
categories            object
best_sellers_rank     object
buybox_seller         object
dtype: object
```

Here are the data types of the columns:

- Most of the columns are of type object, which usually indicates textual or categorical data.
- There are a few columns with numerical data types (int64 and float64).

Upon inspection:

- The rating column is of type object, which might indicate it's stored as text with potential special characters (e.g., "4.5 out of 5 stars"). We might want to extract the numerical rating from it.
- The timestamp column is of type object, so it might be beneficial to convert it to a datetime format for time-based analysis.

Let's start by converting the rating column to a numerical format and the timestamp column to a datetime format.

```
# Convert the 'rating' column to numerical format by extracting the numeric value
amazon_books_df['rating'] = amazon_books_df['rating'].str.extract('(\d+\.\d+)').astype(float)

# Convert the 'timestamp' column to datetime format
amazon_books_df['timestamp'] = pd.to_datetime(amazon_books_df['timestamp'])

# Check the data types again to confirm the changes
updated_data_types = amazon_books_df[['rating', 'timestamp']].dtypes
updated_data_types
RESULT
rating                   float64
timestamp    datetime64[ns, UTC]
dtype: object
```

The rating column has been successfully converted to a numerical format (float64), and the timestamp column has been converted to a datetime format (datetime64[ns, UTC]).

The preprocessing phase is comprehensive and involves multiple steps to ensure that the dataset is clean and ready for analysis. We have successfully addressed missing values, outliers, and data type conversions.

3 Phase 3: Transformation

In this phase, we'll transform the data to make it suitable for analysis. This might involve:

1. Feature Engineering: Creating new features from existing ones to enhance the dataset's information.
2. Normalization: Bringing numerical columns to a similar scale.
3. Encoding: Converting categorical variables into a format suitable for machine learning algorithms.

For our exploratory data analysis, we might not need extensive transformations, but we'll explore feature engineering to derive meaningful insights from the dataset.

3.1 Feature Engineering

A couple of potential features to derive:

1. From the categories column, we can extract the primary category for each book.
2. From the delivery column, we can extract whether free delivery is available.

Let's start by engineering these features.

```
# Extract primary category from the 'categories' column
amazon_books_df['primary_category'] = amazon_books_df['categories'].str.extract(r'\"(.*?)\"')

# Extract information about free delivery from the 'delivery' column
amazon_books_df['free_delivery'] = amazon_books_df['delivery'].str.contains('FREE delivery', case=False).astype(int)

# Display the first few rows to check the new features
amazon_books_df[['primary_category', 'free_delivery']].head()
```

RESULT

	primary_category	free_delivery
0	Books	1
1	Books	1
2	Books	1
3	Books	1
4	Books	1

We have successfully engineered two new features:

1. primary_category: Represents the primary category of each book.
2. free_delivery: Indicates whether free delivery is available for the book (1 for yes, 0 for no).

With these new features, our dataset is further enriched and can provide more detailed insights during the data mining phase.

4 Phase 4: Data Mining

In this phase, we'll explore the data to uncover patterns and insights. This involves:

1. Descriptive statistics: Understanding the central tendencies and spread of numerical columns.
2. Visualizations: Using plots to understand the distribution, relationships, and patterns in the data.
3. Advanced analysis: Applying machine learning or statistical techniques to derive deeper insights.

Let's start with some descriptive statistics for our numerical columns.

```python
# Descriptive statistics for numerical columns
descriptive_stats = amazon_books_df.describe(include=[float, int])
descriptive_stats
```

RESULT

```
       answered_questions     discount  final_price  images_count  \
count         2269.000000  2269.000000  2269.000000   2269.000000
mean             0.034376     7.436007    12.848561      2.093874
std              1.244976     7.395791     6.391372      2.955207
min              0.000000     0.500000     1.990000      0.000000
25%              0.000000     6.410000    10.700000      0.000000
50%              0.000000     6.610000    12.005000      1.000000
75%              0.000000     6.800000    13.580000      3.000000
max             58.000000   282.490000   132.990000     52.000000

       initial_price  plus_content       rating  reviews_count  root_bs_rank  \
count    2269.000000        2269.0  2269.000000    2269.000000  2.269000e+03
mean       19.943169           1.0     4.622345   21497.738211  3.085359e+04
std        11.310705           0.0     0.192836   16108.019322  1.253451e+05
min         3.490000           1.0     3.400000   10010.000000  1.000000e+00
25%        17.990000           1.0     4.500000   12393.000000  8.190000e+02
50%        17.990000           1.0     4.700000   16119.000000  3.104000e+03
75%        18.000000           1.0     4.800000   23817.000000  1.472400e+04
max       299.000000           1.0     4.900000  196572.000000  2.904335e+06

       video_count  free_delivery
count  2269.000000    2269.000000
mean      0.000881       0.620978
std       0.041987       0.485250
min       0.000000       0.000000
25%       0.000000       0.000000
50%       0.000000       1.000000
75%       0.000000       1.000000
max       2.000000       1.000000
```

Here's a summary of the descriptive statistics for the numerical columns:

- answered_questions: Most books have not had any questions answered, but there's a book with as many as 58 answered questions.
- discount: The average discount on books is approximately \$7.44, with some books having discounts as high as \$282.49.
- final_price: The average price of books is around \$12.85, with the maximum price being \$132.99.
- images_count: On average, books have about 2 images associated with them, with some books having as many as 52 images.
- rating: The average rating for the books is approximately 4.62 out of 5, indicating a generally positive reception.

- reviews_count: The average number of reviews for the books is around 21,498, with some books having as many as 196,572 reviews.
- root_bs_rank: This might represent the rank of the book in some category. The average rank is around 30,853, but it varies widely.
- free_delivery: About 62% of the books offer free delivery.

Next, we'll use visualizations to get a better understanding of the data's distribution and relationships. We'll focus on:

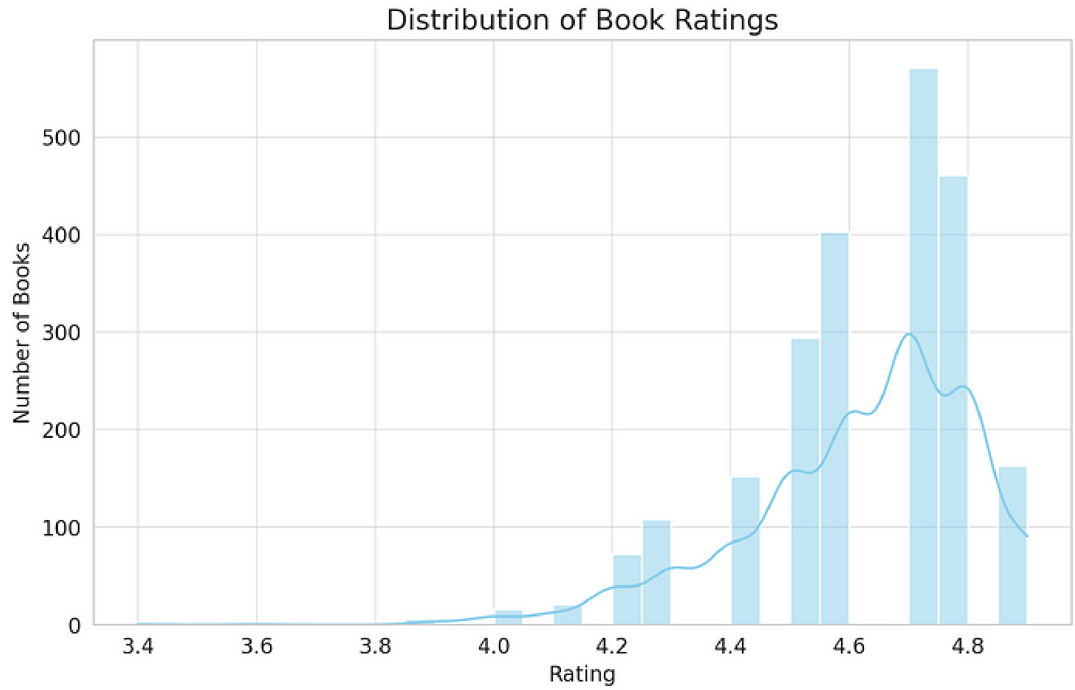
1. Distribution of book ratings.
2. Distribution of final book prices.
3. Relationship between ratings and review counts.

Let's start with the distribution of book ratings.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Set the style for seaborn
sns.set_style("whitegrid")

# Plot the distribution of book ratings
plt.figure(figsize=(10, 6))
sns.histplot(amazon_books_df['rating'], bins=30, kde=True, color='skyblue')
plt.title('Distribution of Book Ratings')
plt.xlabel('Rating')
plt.ylabel('Number of Books')
plt.show()
```

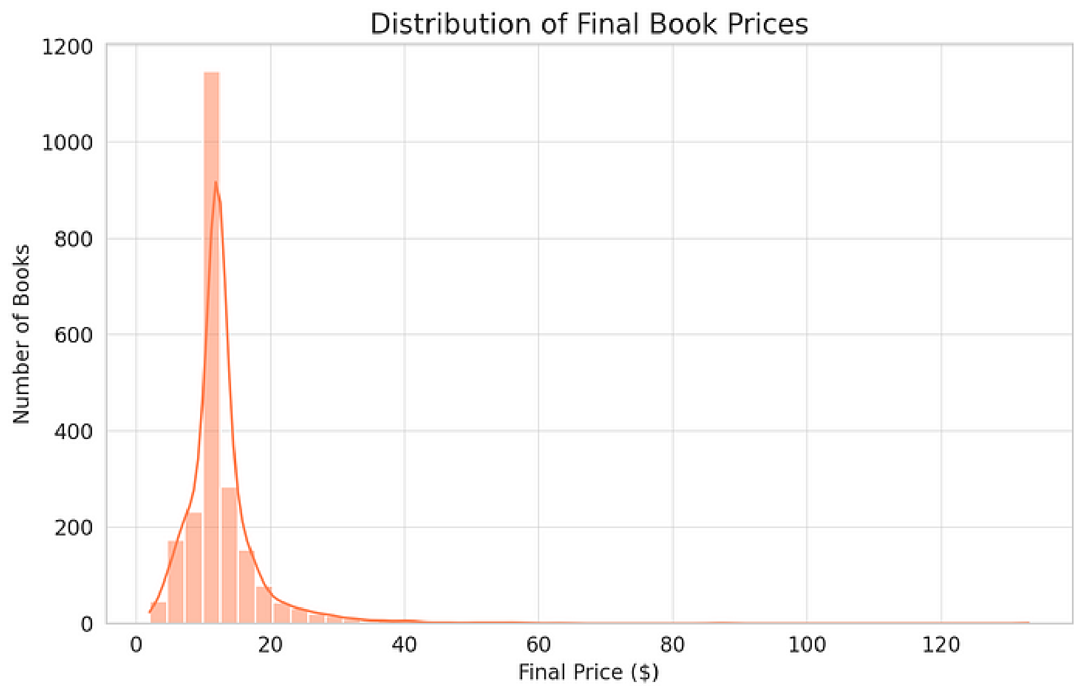


The distribution of book ratings shows that:

- The majority of books have ratings between 4.5 and 5.
- Very few books have ratings below 4.
- The distribution is left-skewed, indicating that most books in this dataset are highly rated.

Next, let's visualize the distribution of final book prices.

```
# Plot the distribution of final book prices
plt.figure(figsize=(10, 6))
sns.histplot(amazon_books_df['final_price'], bins=50, kde=True, color='coral')
plt.title('Distribution of Final Book Prices')
plt.xlabel('Final Price ($)')
plt.ylabel('Number of Books')
plt.show()
```

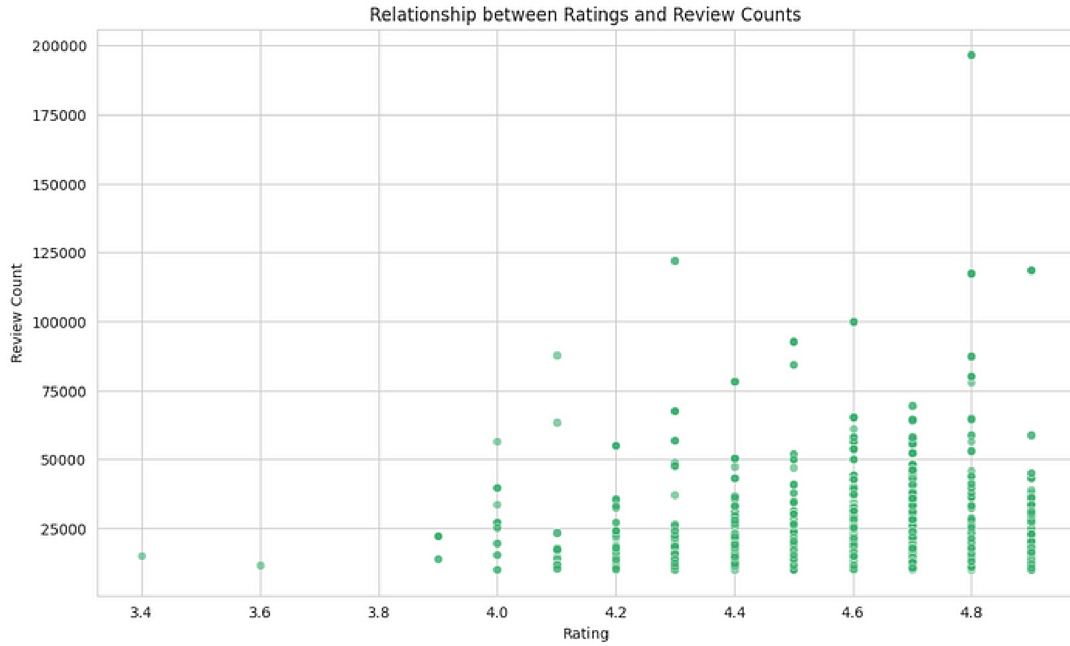


The distribution of final book prices reveals:

- A significant number of books are priced between \$10 and \$20.
- The distribution is right-skewed, indicating that while most books are moderately priced, there are some books that are more expensive.
- There are a few books priced above \$60, which are outliers as we identified earlier.

Lastly, let's explore the relationship between book ratings and the number of reviews to see if more highly-rated books tend to get more reviews. We'll use a scatter plot for this visualization.

```
# Scatter plot to show the relationship between ratings and review counts
plt.figure(figsize=(12, 7))
sns.scatterplot(data=amazon_books_df, x='rating', y='reviews_count', alpha=0.6, color='mediumseagreen')
plt.title('Relationship between Ratings and Review Counts')
plt.xlabel('Rating')
plt.ylabel('Review Count')
plt.show()
```



From the scatter plot showcasing the relationship between ratings and review counts:

- There's no clear linear relationship between the rating of a book and the number of reviews it has.
- Most books, irrespective of their rating, have a review count below 50,000.
- However, books with ratings between 4.5 and 5 tend to have a wider range of review counts compared to books with lower ratings.
- There are a few highly-rated books with a significantly high number of reviews, indicating their popularity.

At this point, we've covered the data mining phase using descriptive statistics and visualizations to understand the dataset better.

5 Phase 5: Interpretation/Evaluation

In this phase, we'll evaluate the insights obtained during the data mining phase and interpret their significance. We'll also attempt to answer any business or research questions and provide actionable insights.
From our analysis so far:

1. Highly Rated Books: The majority of the books in this dataset are highly rated, with ratings between 4.5 and 5. This suggests that the dataset comprises popular books that are well-received by readers.

2. Book Prices: Most books are priced between \$10 and \$20, with a few outliers priced significantly higher. This provides a general price range for popular books on Amazon.

3. Relationship Between Ratings and Reviews: While there isn't a clear linear relationship between ratings and the number of reviews, books with higher ratings (between 4.5 and 5) tend to have a wider range of review counts. This suggests that a high rating doesn't necessarily guarantee a high number of reviews, but highly-rated books have the potential to attract a significant number of reviews.

4. Delivery Options: Approximately 62% of the books offer free delivery, which might be a factor influencing their popularity.

Given these insights:

- **For Authors/Publishers:Ensuring high-quality content can lead to better ratings, potentially attracting more reviews and increasing the book's visibility. Offering competitive prices (in the \$10-\$20 range) and free delivery can also make a book more appealing to potential readers.**

- **For Amazon:Since the majority of popular books offer free delivery, this feature can be highlighted in promotions to attract more buyers. Additionally, Amazon could consider promoting books with high ratings but fewer reviews to increase their visibility.**

6 Phase 6: Deployment

In a real-world scenario, this phase involves implementing the discovered knowledge into the organization's operations. The insights obtained can be used to drive business strategies, improve operations, or develop new products/services. For our EDA:

- A detailed report summarizing the findings can be shared with stakeholders.

- If this analysis was part of a larger project (e.g., building a recommendation system), the insights could guide feature engineering and model selection.

Lastly, to facilitate easy deployment and sharing, we can encapsulate our analysis in a Jupyter notebook (as we've done here) or use tools like PyCaret to deploy models.

Since our task was primarily exploratory data analysis and we didn't build any predictive models, the deployment in our case would mainly involve sharing our findings and insights with the relevant stakeholders.

7 Conclusion

Through the KDD process, we delved deep into the Amazon Popular Books dataset, uncovering valuable insights. The structured approach provided by KDD ensures a comprehensive understanding of the data, guiding future business decisions.

References