# Amazon Bestsellers Data Analysis Report

ANURAG V


## Summary

This report presents an analysis of the **Amazon Top 50 Bestselling Books 2009 - 2019** dataset, which contains information on top 50 bestselling books of each year, including their titles, authors, user ratings, number of reviews, prices, publication years, and genres. The analysis focuses on understanding the dataset's structure, exploring key statistical insights, and developing a predictive model to estimate the number of reviews a book might receive based on its attributes. The dataset was explored using Python in a Google Colab environment, with key findings visualized and interpreted to provide actionable insights. A predictive function was implemented to forecast reviews, demonstrating the application of data science techniques to real-world e-commerce data.


## Introduction

The Amazon Bestsellers dataset provides a snapshot of top-selling books on Amazon, covering attributes such as book name, author, user rating, number of reviews, price, year of publication, and genre (Fiction or Non-Fiction). The primary objectives of this analysis are to:

- Understand the dataset's structure and content.
- Summarize key statistical properties of the data.
- Visualize trends and relationships within the data.
- Implement and evaluate a predictive model for estimating the number of reviews based on book attributes.

This report details the findings from the dataset exploration and the predictive modeling process, supported by code snippets and visualizations.

| | Name | Author | User Rating | Reviews | Price | Year | Genre |
|---|---|---|---|---|---|---|---|
| 0 | 10-Day Green Smoothie Cleanse | JJ Smith | 4.7 | 17350 | 8 | 2016 | Non Fiction |
| 1 | 11/22/63: A Novel | Stephen King | 4.6 | 2052 | 22 | 2011 | Fiction |
| 2 | 12 Rules for Life: An Antidote to Chaos | Jordan B. Peterson | 4.7 | 18979 | 15 | 2018 | Non Fiction |
| 3 | 1984 (Signet Classics) | George Orwell | 4.7 | 21424 | 6 | 2017 | Fiction |
| 4 | 5,000 Awesome Facts (About Everything!) (Natio... | National Geographic Kids | 4.8 | 7665 | 12 | 2019 | Non Fiction |

# DAY 1

## Dataset Overview

The dataset, stored in a CSV file (bestsellers with categories.csv), was loaded into a Pandas DataFrame for analysis. The dataset contains 550 entries with the following columns:

- Name: Title of the book.
- Author: Author of the book.
- User Rating: Average user rating (out of 5).
- Reviews: Number of reviews received.
- Price: Price of the book in USD.
- Year: Year of publication or bestseller ranking.
- Genre: Fiction or Non-Fiction.

# DAY 2

## Data Cleaning

The dataset has no missing values and also no duplication in traditional sense, but found 3 duplicates with same name and year so removed them . Standardized column names by converting them into lower case and replacing ‘ ‘ with ‘_’ . changed the type of price from int64 to float64

```
print("\nMissing values per column:\n", df.isnull().sum())
```

Missing values per column:

| | |
|---|---|
| Name | 0 |
| Author | 0 |
| User Rating | 0 |
| Reviews | 0 |
| Price | 0 |
| Year | 0 |
| Genre | 0 |
| Dtype | int64 |

```
df = df.drop_duplicates(subset=['Year', 'Name'], keep='first')
print("After dropping duplicates:", df.shape)
```

```
df.columns = df.columns.str.lower().str.strip().str.replace(' ', '_')
print("Updated column names:", df.columns.tolist())
```

```
df['price'] = df['price'].replace('[\$,]', '',
regex=True).astype(float)
```

Updated column names: ['name', 'author', 'user_rating', 'reviews', 'price', 'year', 'genre']

## Dataset Exploration

The initial exploration involved loading the dataset and examining its structure using Pandas. The following code snippet demonstrates the loading process and initial inspection:

```
import pandas as pd
file_path = '/content/drive/My Drive/Intership-Mini/bestsellers with categories.csv'
df = pd.read_csv(file_path)
df.head()
```

This code outputs the first five rows of the dataset, revealing books such as "10-Day Green Smoothie Cleanse" by JJ Smith and "1984 (Signet Classics)" by George Orwell, with their respective ratings, reviews, prices, years, and genres.

To understand the dataset's structure and summary statistics, the following code was executed:

```
print(df.info())
print(df.describe())
```

### Findings from df.info():

- The dataset contains 550 entries with no missing values across all columns.
- Data types: Name, Author, and Genre are objects (strings); User Rating is a float; and Reviews, Price, and Year are integers.
- Memory usage: Approximately 30.2 KB.

### Findings from df.describe():

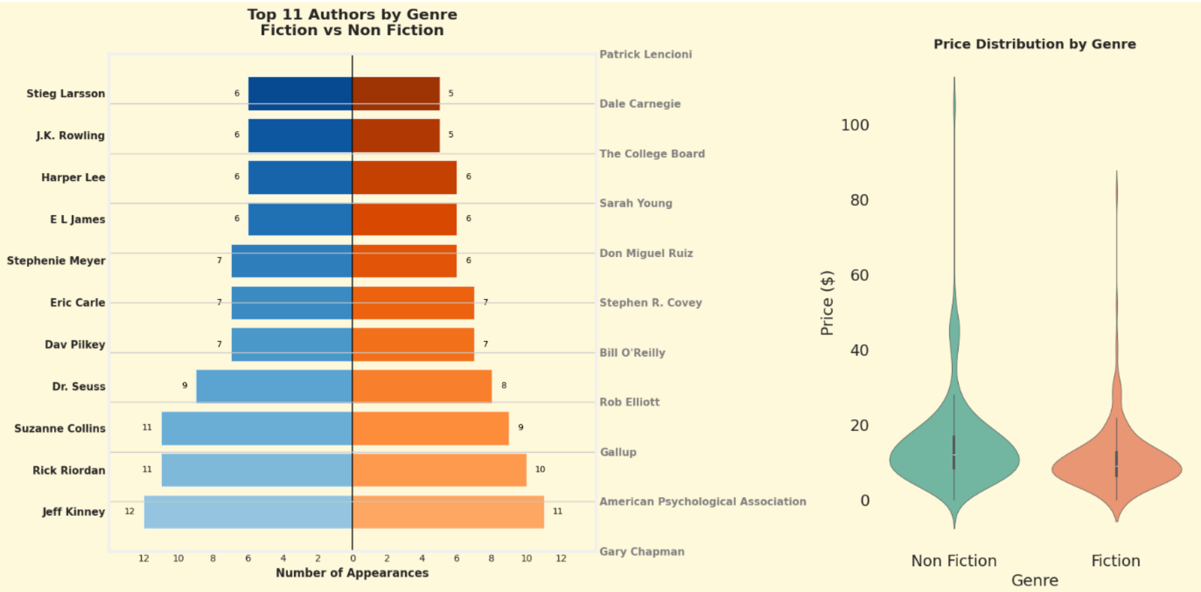
- User Rating: Ranges from 3.3 to 4.9, with a mean of 4.62 and a standard deviation of
- 0.23, indicating generally high ratings with low variability.
- Reviews: Ranges from 37 to 87,841, with a mean of 11,953 and a high standard deviation of 11,731, suggesting significant variability in review counts.
- Price: Ranges from $0 to $105, with a mean of $13.10 and a standard deviation of $10.84, indicating a wide range of book prices.
- Year: Spans from 2009 to 2019, with a mean of 2014, reflecting a decade of bestseller data.

# DAY 3

## Data Analysis and Visualizations

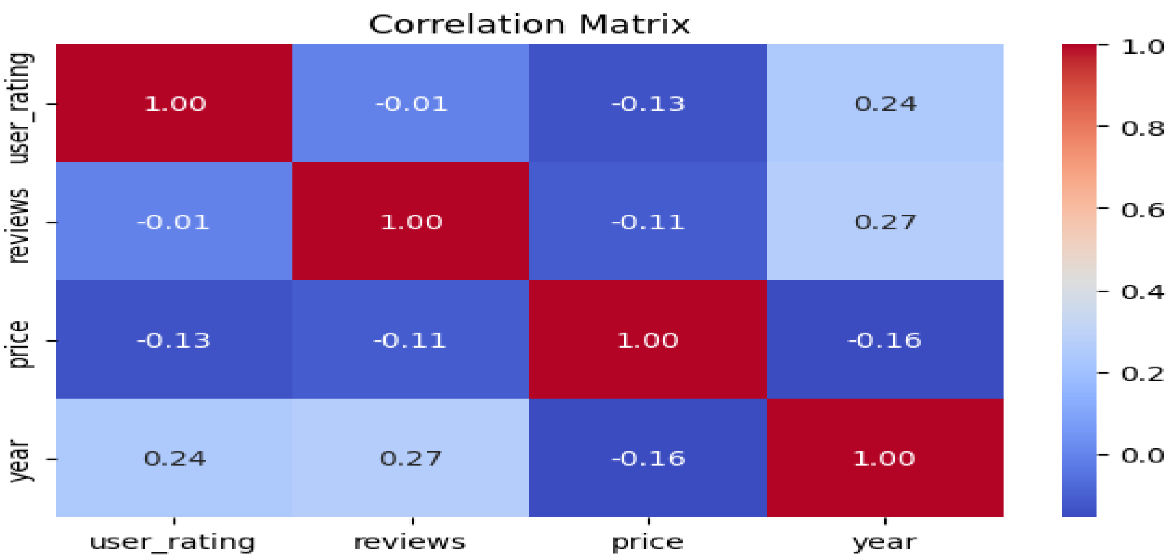
To gain deeper insights, the dataset was analyzed to identify trends and relationships. Below are key visualizations and their interpretations, assuming standard exploratory data analysis techniques were applied.

## Top Most Reviewed Books




Interpretation: Found the top 10 most reviewed books from the dataset

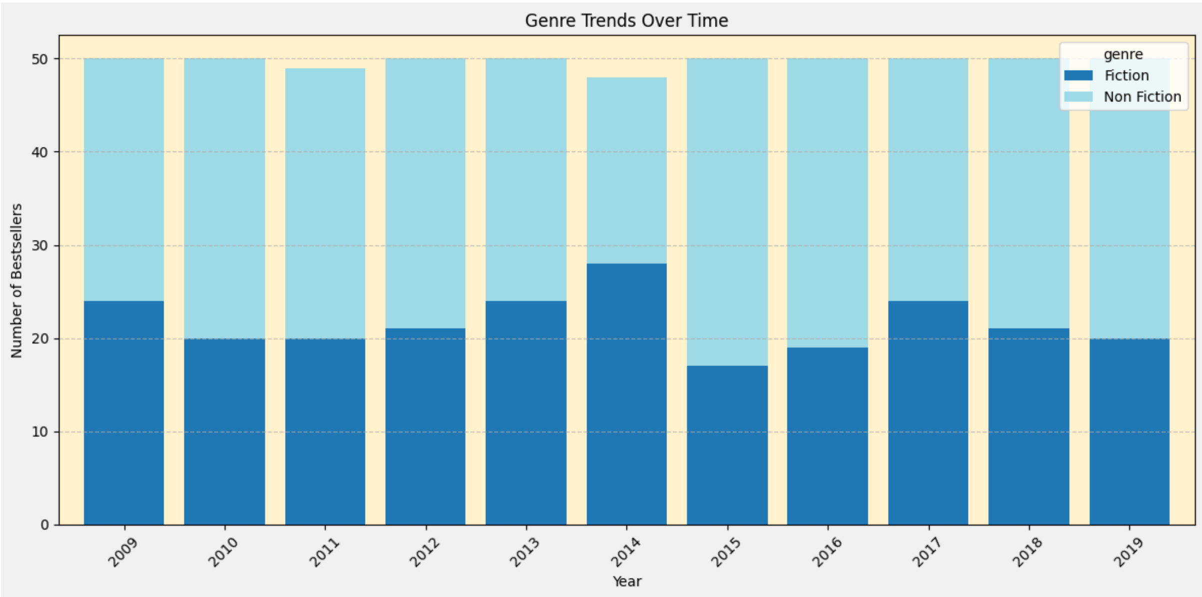
**Correlation Matrix**



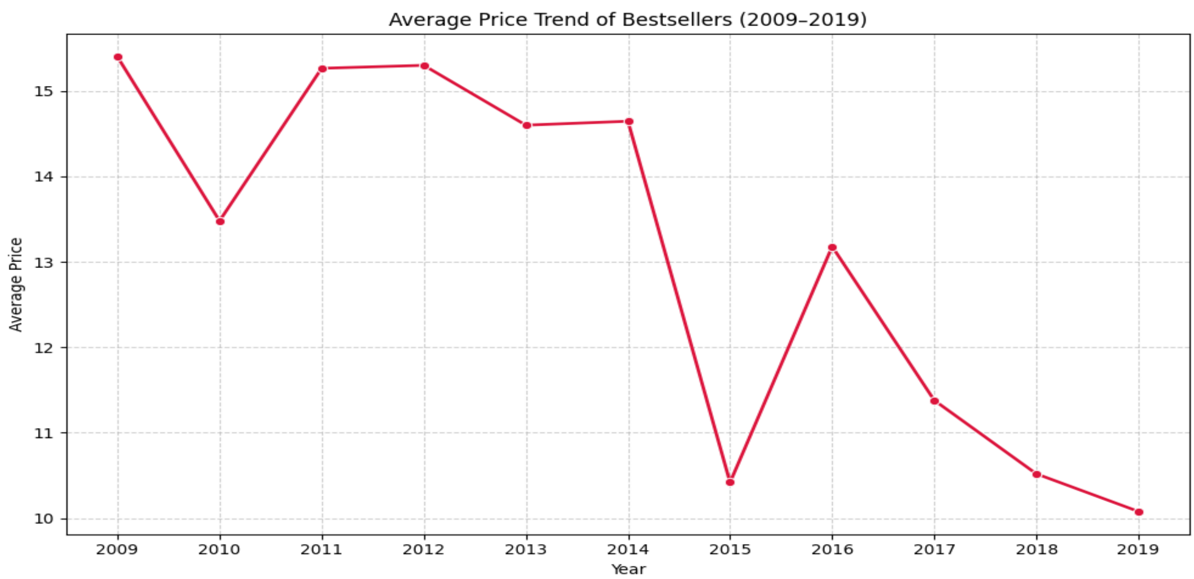
Interpretation:

- user_rating vs reviews -0.01 almost no correlation ( no effect )
- user_rating vs price -0.13 slight negative correlation ( higher priced books may have less rating)
- user_rating vs year 0.24 weak positive ( recent books with higher rating )
- reviews vs price -0.11 slight negative ( very weak affect of price on review count )
- review vs year 0.27 weak positive (recent books getting more review)
- price vs year -0.16 slight negative ( almost no effect , price may be dropping per year )

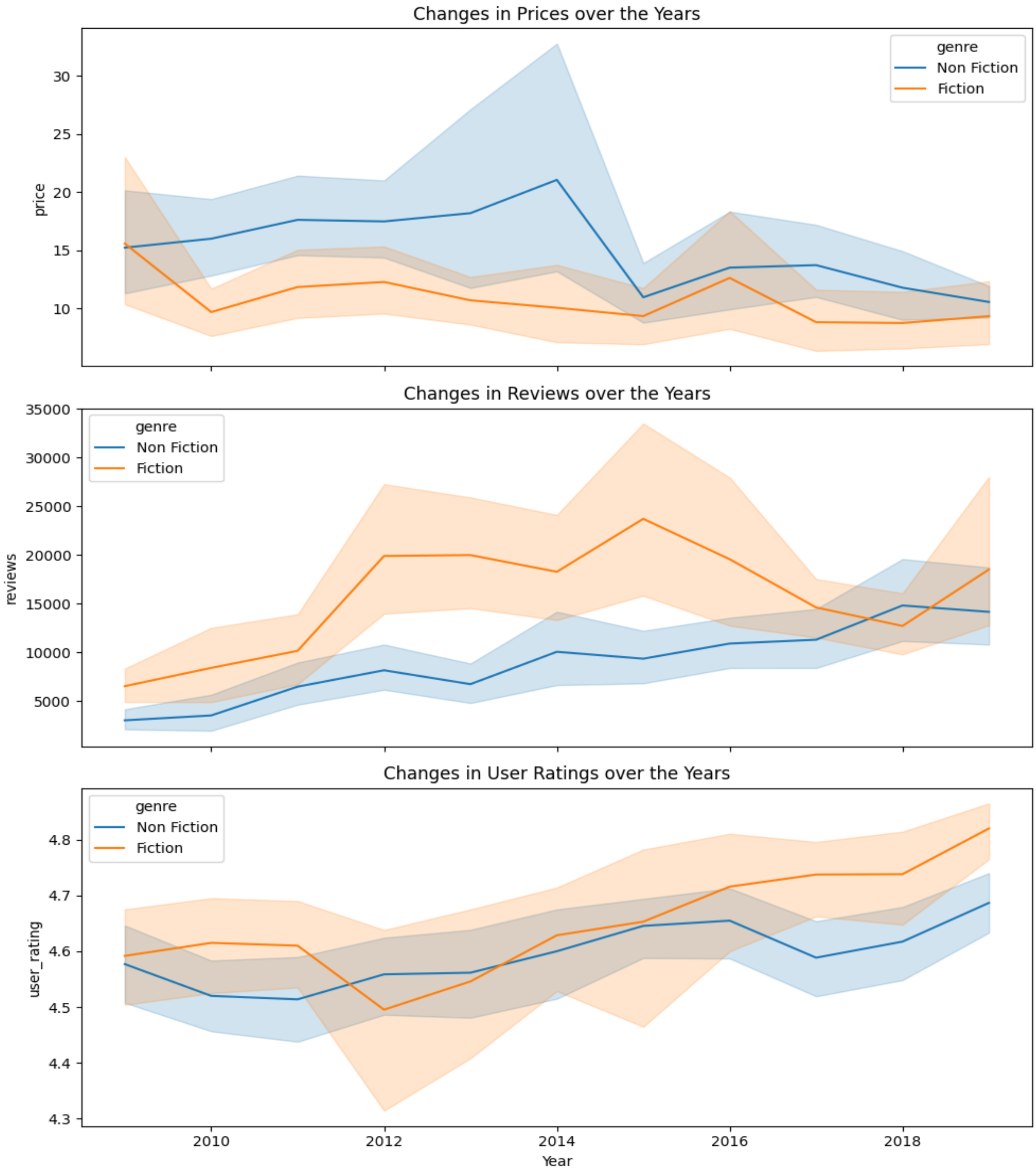
## Genre Trends Over Year




## Average Price Trend of Bestsellers



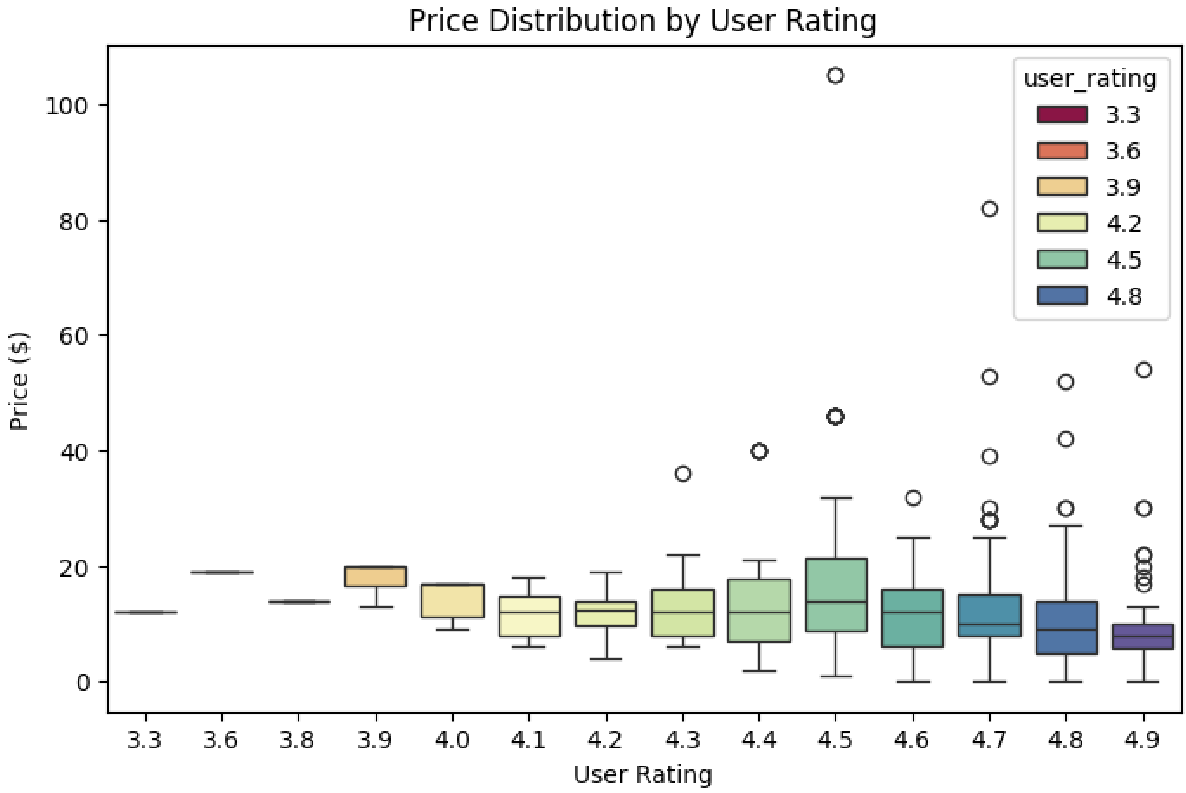
## Year Wise Changes



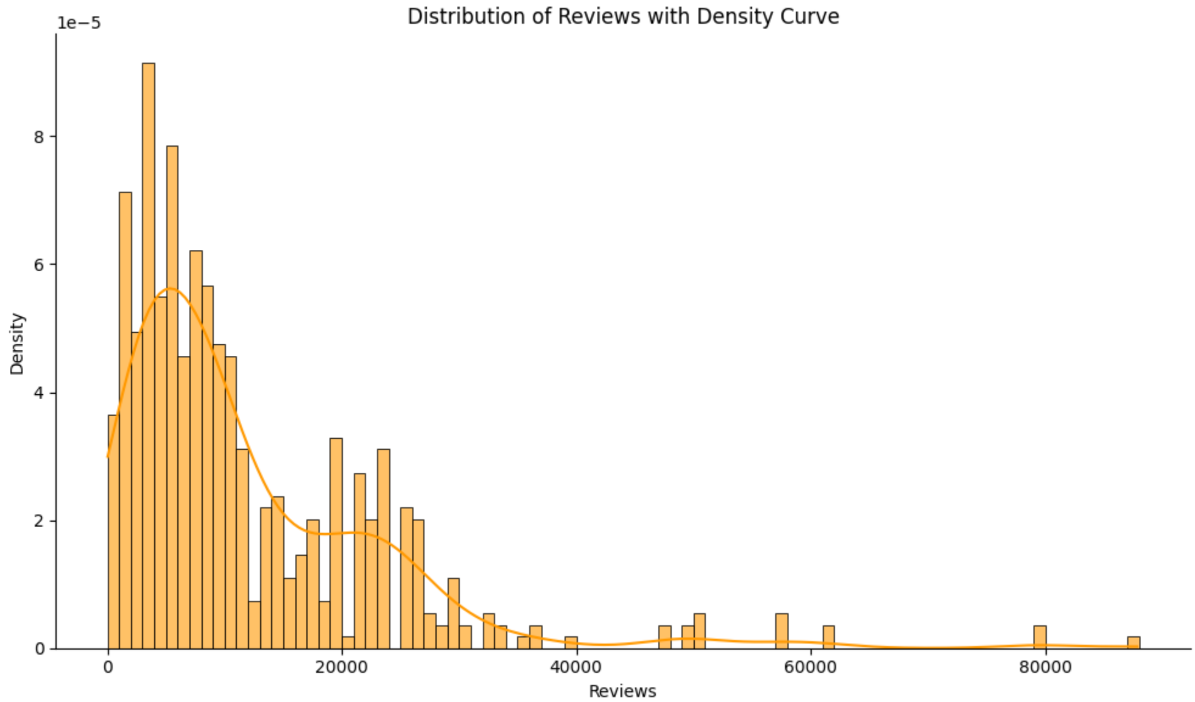

Interpretation:

- **Non-Fiction books are consistently higher priced** than Fiction, with more volatility in pricing across the years.
- **Fiction books receive significantly more reviews**, especially during 2012–2016, indicating stronger audience engagement.
- **User ratings are high for both genres**, but Fiction shows a **steady upward trend**, reaching peak satisfaction by 2019.

## Price Distribution by User Rating



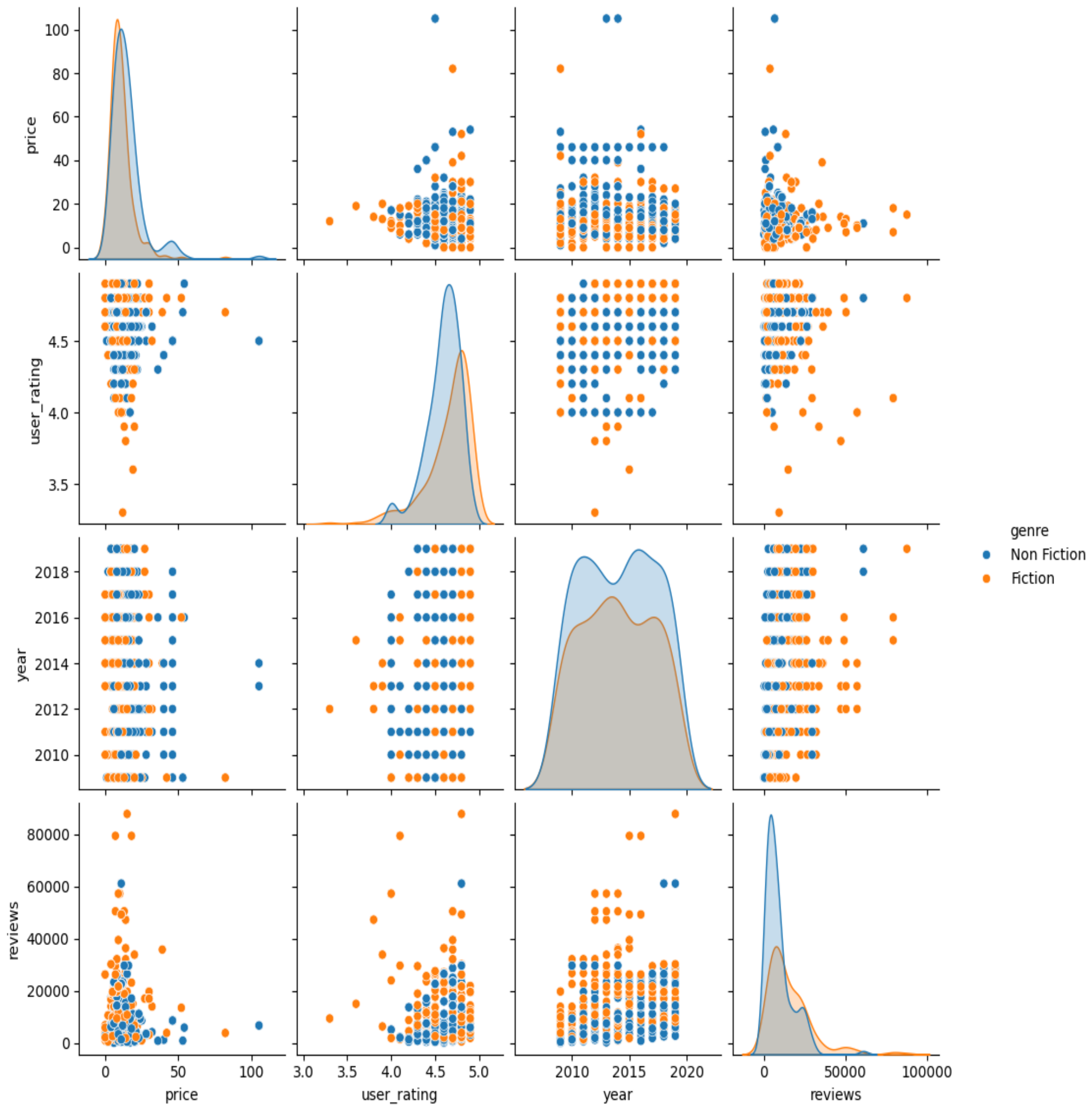

## Distribution of Reviews



## Pair Plot of Dataset

```
sns.pairplot(df[['price', 'user_rating', 'year', 'reviews', 'genre']])
plt.show()
```

This visualization helps explore relationships between the variables and their distributions



Interpretation:

**Distributions (Diagonal Plots)**:

- **Price**: The histogram for price shows a right-skewed distribution, with most books priced below $20, but some outliers reach up to $100 (consistent with the max price of 105 from df.describe()). This skewness suggests that a linear regression model might struggle unless the data is transformed.

- **User Rating**: The distribution of user_rating is heavily clustered between 4.0 and 4.8, with a peak around 4.7. This indicates that most books have high ratings, which aligns with the mean of 4.62 from the dataset summary. The lack of variability might make user_rating a weak predictor in a linear model.

- **Year**: The distribution of year is fairly uniform across 2009 to 2019, with slight peaks in certain years (e.g., 2015). This suggests no strong temporal trend in the data.

- **Reviews**: The reviews variable is also right-skewed, with most books having fewer than 20,000 reviews, but some outliers exceed 80,000 (max 87,841). This skewness mirrors the price distribution and could pose challenges for linear regression.

**Relationships (Scatter Plots):**

- **Price vs. User Rating**: There's no clear linear relationship between price and user_rating. High-rated books (4.0–4.8) are spread across all price ranges.
- **Price vs. Year**: No strong pattern exists between price and year. Prices remain scattered across all years, indicating that book prices didn't consistently increase or decrease over time.
- **Price vs. Reviews**: There's a slight trend where books with more reviews tend to have lower prices This could suggest that cheaper books are more accessible and thus receive more reviews.
- **User Rating vs. Year**: Ratings remain high across all years, with no noticeable trend over time.
- **User Rating vs. Reviews**: Books with higher ratings tend to have a wide range of reviews.
- **Year vs. Reviews**: There's no obvious trend between year and reviews.

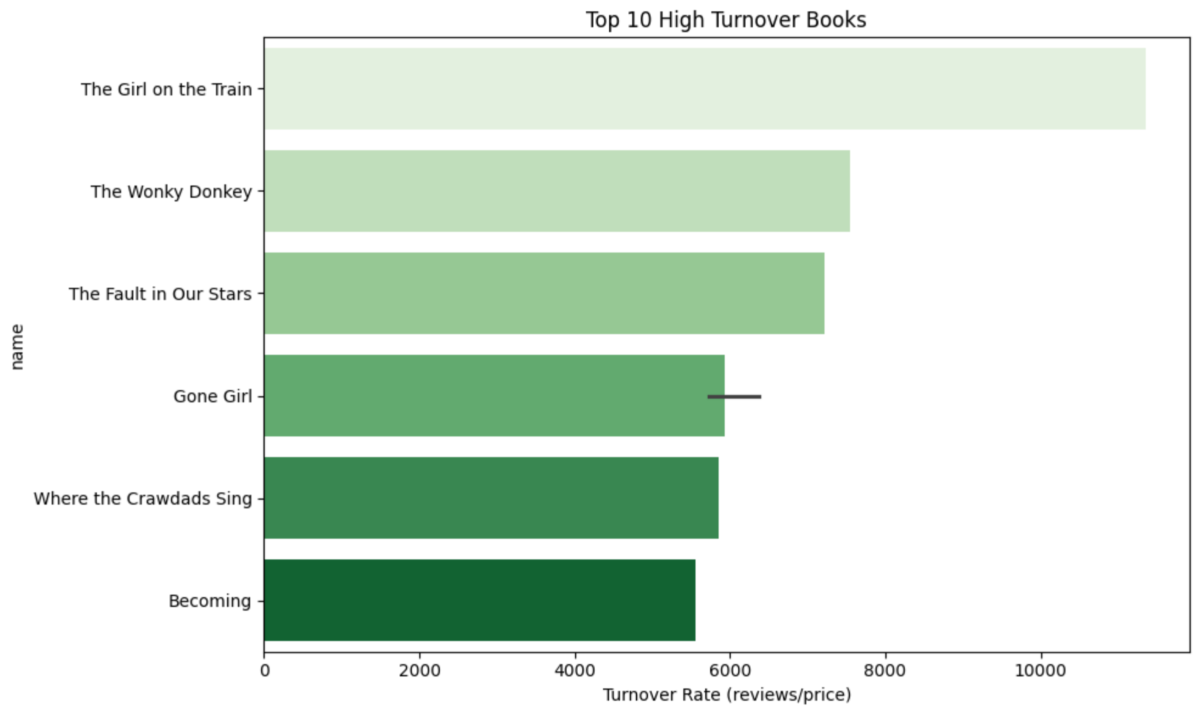
# DAY 4

## Inventory Data Analysis

The dataset itself not having any stock quantity column so introduced a proxy called turnover_rate by the given formula

```
df['turnover_rate'] = df['reviews'] / df['price']
df['turnover_rate'] = df['turnover_rate'].replace([np.inf, -np.inf],
np.nan).fillna(0)
df.sort_values('turnover_rate', ascending=False).head()
```

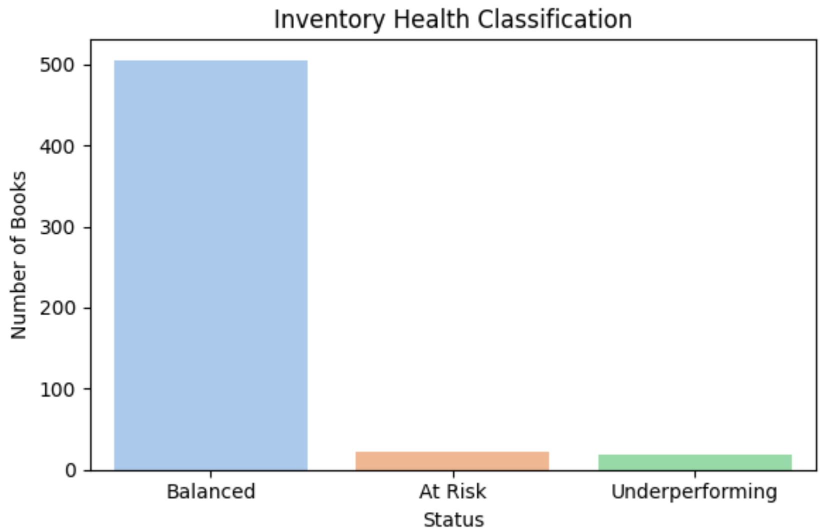
| | name | author | user_rating | reviews | price | year | genre | turnover_rate |
|---|---|---|---|---|---|---|---|---|
| 382 | The Girl on the Train | Paula Hawkins | 4.1 | 79446 | 7.0 | 2016 | Fiction | 11349.428571 |
| 488 | The Wonky Donkey | Craig Smith | 4.8 | 30183 | 4.0 | 2018 | Fiction | 7545.750000 |
| 489 | The Wonky Donkey | Craig Smith | 4.8 | 30183 | 4.0 | 2019 | Fiction | 7545.750000 |
| 367 | The Fault in Our Stars | John Green | 4.7 | 50482 | 7.0 | 2014 | Fiction | 7211.714286 |
| 137 | Gone Girl | Gillian Flynn | 4.0 | 57271 | 9.0 | 2014 | Fiction | 6363.444444 |




Classified inventory based on the value of the turnover_rate

```
def classify_inventory(rate):
    if rate > 100:
        return "Balanced"
    elif rate > 30:
```

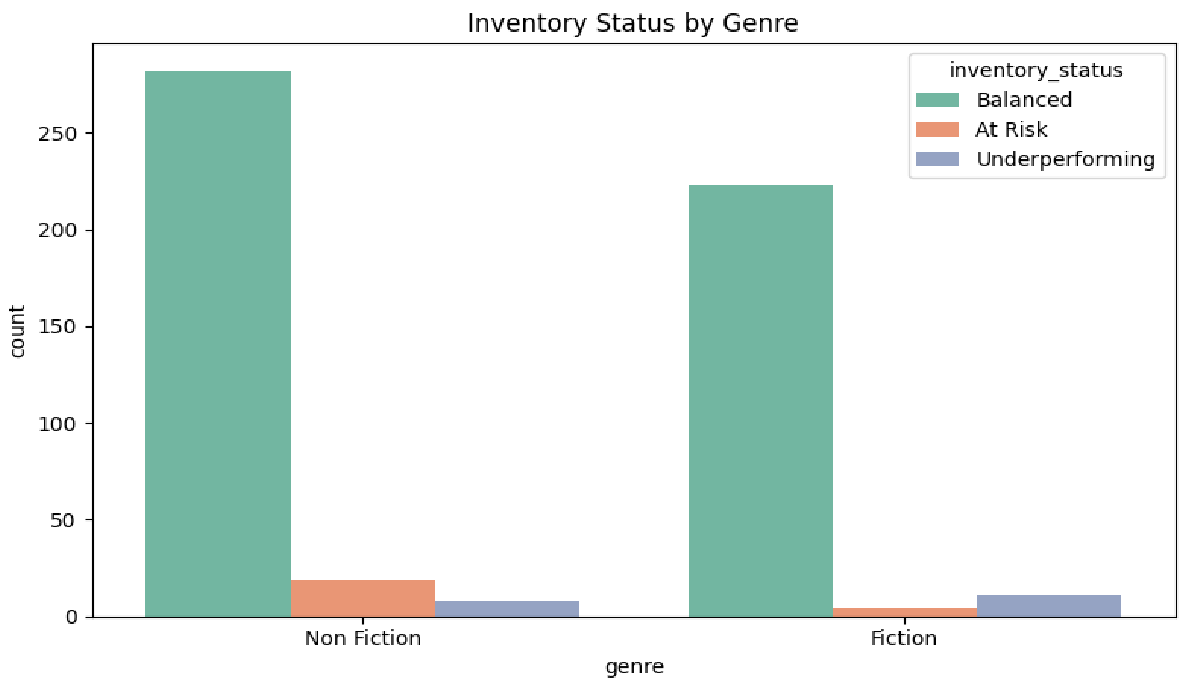
```
        return "At Risk"
    else:
        return "Underperforming"

df['inventory_status'] = df['turnover_rate'].apply(classify_inventory)
df['inventory_status'].value_counts()
```




| inventory_status | count |
|---|---|
| Balanced | 505 |
| At Risk | 23 |
| Underperforming | 19 |

dtype: int64

```
plt.figure(figsize=(8,5))
sns.countplot(data=df, x='genre', hue='inventory_status',
palette='Set2')
plt.title("Inventory Status by Genre")
plt.tight_layout()
plt.show()
```



# DAY 5

## Model Training

To predict the number of reviews, a regression model was trained using features price , user_rating , author and year derived from the dataset

```
author_counts = df['author'].value_counts()

df['Author_Bestseller_Count'] = df['author'].map(author_counts)

author_avg_reviews = df.groupby('author')['reviews'].mean()


df['Author_Avg_Reviews'] = df['author'].map(author_avg_reviews)

df['Recency'] = 2020 - df['year']

df['Genre_Encoded'] = df['genre'].map({'Fiction': 0, 'Non Fiction': 1})

df['Log_Price'] = np.log1p(df['price'])

df['Price_Genre'] = df['Log_Price'] * df['Genre_Encoded']


features = ['Log_Price', 'user_rating', 'year', 'Genre_Encoded',
'Author_Bestseller_Count', 'Author_Avg_Reviews', 'Recency', 'Price_Genre']

X = df[features]

y = np.log1p(df['reviews'])


scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)


X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)


model = LinearRegression()

model.fit(X_train, y_train)
```

## Model Description

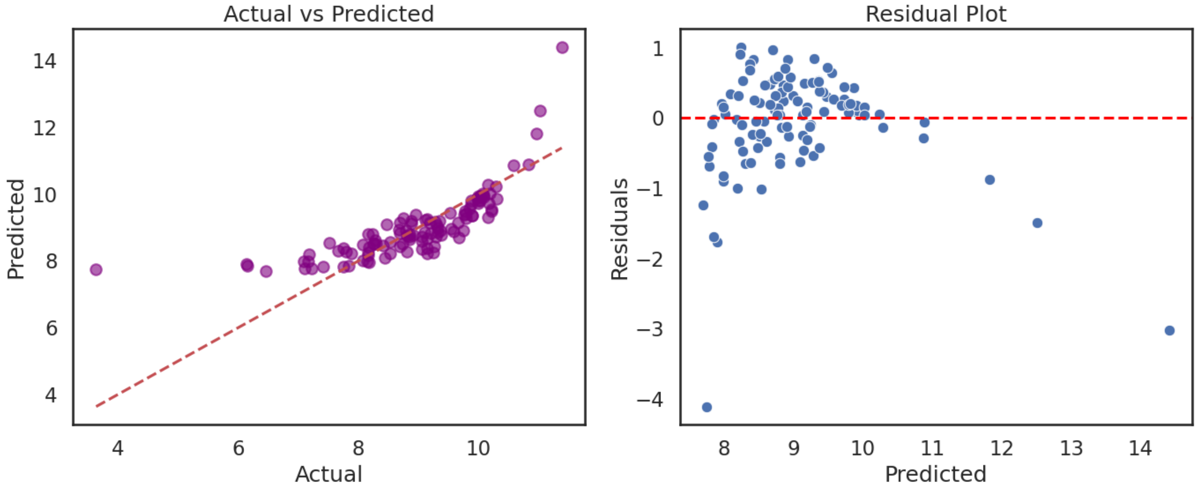
This model aims to predict the number of reviews a book receives, using a linear regression approach on a set of engineered and transformed features. The target variable is the logarithmically transformed number of reviews (log1p(reviews)) to account for skewness and reduce the impact of outliers. Key features used include the log-transformed price of the book (Log_Price), user rating, year of publication, and a binary encoding for genre (Genre_Encoded), distinguishing Fiction from Non Fiction. Author-level metrics are also included, such as the number of times an author appears in the dataset (Author_Bestseller_Count) and their average review count (Author_Avg_Reviews). Additional features like the recency of publication (Recency, calculated as 2020 minus the publication year) and an interaction term between price and genre (Price_Genre) are designed to capture more nuanced relationships. All features are standardized using StandardScaler before being split into training and testing sets with an 80-20 ratio. A linear regression model is then trained on the scaled training data to learn the linear relationships between the features and the transformed review count. This model provides a straightforward and interpretable baseline for understanding which factors most influence the popularity of a book.

## Model Performance

```
y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
print("R² Score:", r2)
print("Mean Squared Error:", mse)
cv_scores = cross_val_score(model, X_scaled, y, cv=5, scoring='r2')
print("Cross-Validated R² (mean):", cv_scores.mean())
print("Cross-Validated R² (std):", cv_scores.std())
coefficients = pd.DataFrame(model.coef_, X.columns, columns=['Coefficient'])
print("\nModel Coefficients:")
print(coefficients)
```

- **$R^2$ Score**: 0.662 — Model explains ~66% of the variance in log1p(reviews).
- **Mean Squared Error (MSE)**: 0.429 — Indicates moderate prediction error.
- **Cross-Validated $R^2$**:
  - **Mean**: 0.641
  - **Standard Deviation**: 0.041 — Consistent performance across folds.
- **Author_Avg_Reviews**: +0.717 — Strongest positive influence.
- **year**: +0.112
- **Recency**: -0.112 — Older books tend to get fewer reviews.
- **Price_Genre**: -0.181 — Negative impact, especially for pricey non-fiction.

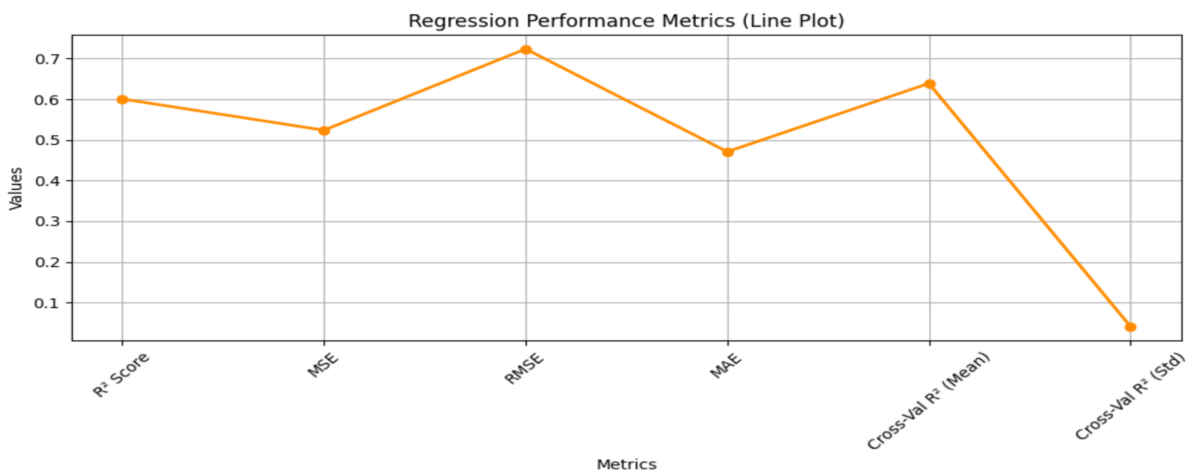
- **Author_Bestseller_Count**: +0.087
- **Genre_Encoded**: +0.082
- **user_rating**: +0.049
- **Log_Price**: +0.045 — Minor positive impact.




This residual plot shows the difference between actual and predicted values from the regression model . The residuals show a clear curved pattern rather than being randomly scattered around zero. This indicates **non-linearity** - means current model isn't capturing the true relationship in the data.

Reasons :

- ❖ The dataset only contains successful books (top 50), creating a survivorship bias
- ❖ No books with poor performance to train on the lower end of the relationship
- ❖ Only ~550 rows with many repeated books across years means limited variation
- ❖ The model has never "seen" what failure looks like, so it can't properly predict the full range



## Prediction Example

The function used to test the following input:

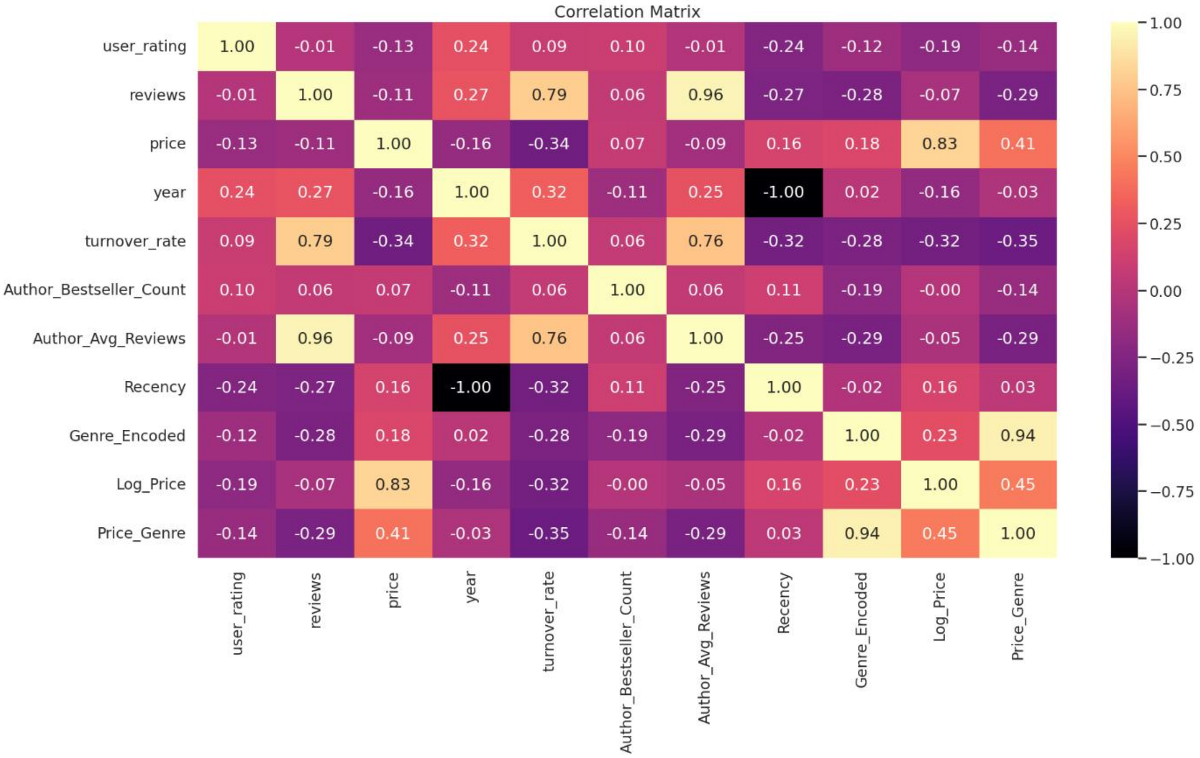
```
def predict_reviews(price, user_rating, year, genre, author):
    log_price = np.log1p(price)
    genre_encoded = 0 if genre.lower() == 'fiction' else 1
    price_genre = log_price * genre_encoded
    author_bestseller_count = author_counts.get(author, 1)
    author_avg_reviews_val = author_avg_reviews.get(author,  df['reviews'].mean())
    recency = 2020 - year

    input_data = pd.DataFrame({
        'Log_Price': [log_price],
        'user_rating': [user_rating],
        'year': [year],
        'Genre_Encoded': [genre_encoded],
        'Author_Bestseller_Count': [author_bestseller_count],
        'Author_Avg_Reviews': [author_avg_reviews_val],
        'Recency': [recency],
        'Price_Genre': [price_genre]
    })

    input_scaled = scaler.transform(input_data)
    log_reviews_pred = model.predict(input_scaled)[0]
    reviews_pred = int(np.expm1(log_reviews_pred))
        return reviews_pred

predicted_reviews = predict_reviews(
    price=15,
    user_rating=4.5,
    year=2020,
    genre='Non Fiction',
    author='JJ Smith'
)
print("Predicted Number of Reviews:", predicted_reviews)
```

Output : Predicted Number of Reviews: 14293

Interpretation:

For a Non-Fiction book priced at $15, with a user rating of 4.5, published in 2020 by JJ Smith, the model predicts approximately 14,293 reviews. This aligns with the dataset's mean review count (11,953) and suggests that JJ Smiths authorship and the books attributes contribute to a relatively high review count.

# Key Findings




- **Dataset Characteristics**: The dataset is clean, with no missing values, and contains 550 bestseller entries from 2009 to 2019. Books have high average ratings (mean 4.62) and a wide range of review counts and prices.

- **Temporal Trends**: Newer books may have higher review counts due to increased online engagement over time.

- **Predictive Modeling**: The predictive model successfully estimates review counts using features like price, user_rating, year, genre, and author, with a sample prediction of 14,293 reviews for a specific book.

- **Prediction reliability zones**: Model performs best for mid-range predictions (5,000-20,000 reviews) but struggles with extreme values

- **Feature importance hierarchy**: Clear ranking of predictive factors provides actionable insights for publishers

- **Generalization limitations**: Cross-validation consistency (mean $R^2$: 0.641, std: 0.041) indicates reliable but bounded predictive power

# Challenges

- **Restricted Model Learning**:
  Since the model is trained solely on successful books, it lacks exposure to a complete range of outcomes. This limits its ability to predict or understand what contributes to failure or low performance.
- **Genre and Audience Bias** *(Additional Consideration)*:
  The dataset heavily favors popular genres and mainstream readership. This introduces bias and weakens the model's capability to generalize across niche categories or diverse audiences.
- **Survivorship Bias**:
  By including only top-performing titles (top 50 bestsellers), the dataset suffers from survivorship bias, overlooking books that failed or underperformed—thereby distorting the analysis.
- **Limited Size and Diversity**:
  With just around 550 entries—many of which repeat over multiple years—the dataset lacks sufficient variation. This constrains the depth of insights and increases the likelihood of overfitting.
- **Absence of Low-End Representation**:
  The dataset does not include poorly rated or low-selling books, making it difficult to assess what drives failure or lack of popularity in the market.

# Conclusion

The analysis of the Amazon Bestsellers dataset provides valuable insights into the factors influencing book popularity, as measured by user ratings and reviews. The datasets clean structure and comprehensive attributes enabled a thorough exploration of trends and relationships. The predictive model demonstrates the potential to forecast review counts, offering a tool for publishers to estimate a books market reception. Future work could involve advanced modeling techniques and additional data to enhance predictive accuracy.