Wil (Wilson) Sheffield

6 May 2025

Sentiment Analysis vs. Star Ratings in User Reviews Across Media, Apps, and Products

I. Abstract

This project examines the correlation of user-generated star ratings and sentiment in written reviews of three consumer markets: film, mobile applications, and e-commerce items. In order to include a variety of opinions, data was obtained from three differing sources: web-scraped reviews at Letterboxd (films), API-retrieved reviews at the Google Play Store (applications), and CSV-imported product reviews from the Amazon Sales Dataset at Kaggle. Each data set was cleaned, normalized, and structured into one common format for the purpose of investigation. A sentiment score was given to each particular review using a keyword method, and sentiment polarity (positive, neutral, negative) was calculated from the sentiment scores.

The cleaned and aggregated data set, comprising more than 4,000 reviews, was analyzed to detect and quantify mismatches, defined as instances where sentiment and rating significantly parted. Visualizations and statistical tests (Chi-square, ANOVA, T-test, correlation) confirmed, via the data, that although there is moderate correlation between sentiment and overall rating, mismatches are present and they differ by platform. Mismatches tended to have even more extreme sentiment than matches, indicating some users have strong opinions even when they have low or high numerical ratings. This supports the hypothesis that using star ratings to get at user opinion might disguise the full subtlety of user sentiment, and platform context is involved in what is conveyed through sentiment.

II. Motivation

I wanted to research a topic both relevant to the tools I have learned about and practiced in class as well as to my desired career goal of working in data science within business tech (FAANG). With all of that said, after doing some research, I chose to examine the mismatch that often occurs between user-posted product reviews and their accompanying star ratings. In many cases, users will award some product with high marks (i.e. 4 or 5 stars), but then go on to write critical language about it in its review, or vice versa. On the business side, this discrepancy can lead to distorted signals in product recommendation systems and customer satisfaction metrics. To have a better understanding of this phenomenon, I wanted to examine this "gap" with an interpretable analysis using text-level features of product reviews and structured product metadata. One hypothesis examined is whether some platforms have more mismatches between text sentiment and ratings than others, and if mismatched reviews have more extreme sentiment.

III. Datasets

1) Letterboxd Movie Review Data (Scraped Source):

To illustrate the movie domain, I scraped user reviews from Letterboxd.com, which is a social film review resource. From 45 movies in different genres and decades, I scraped up to three pages of reviews from each. Each record contains the title of the movie, the full text of the review, and star rating, which were initially presented as icons (i.e.. ★★★½) and later converted into float values scaled to 1–5. Reviews were extracted from structured HTML via the requests and BeautifulSoup libraries. The outcome was well over 1,500 film reviews gathered from direct user inputs. This dataset offered information about viewer sentiment toward films and was the project's scraping portion. In my submission, I have included the cleaned, isolated dataset as "movie_reviews.csv".

2) Google Play Store App Review Data (API-Fetched Source):

For the mobile application space, I leveraged the Google Play Store API through RapidAPI to pull review data for seven popular applications, such as Facebook, Spotify, TikTok, and Zoom. Each returned review contained the app name, text of the review, and user-assigned numerical star rating. I pulled 200 English-language U.S. reviews for each of the apps for a total of 1,400+ records. Reviews were then parsed into one standardized Pandas DataFrame and output as CSV. This API source provided a programmatic view and regularly refreshed window into how users view functionality and experience on the mobile space. Noteworthy is the fact the RapidAPI source needed to incorporate a key provided through the RapidAPI website, providing limitations on the volume of requests per month. That key is embedded within my notebook and available to subsequent users. The cleaned, individual dataset is included as part of my submission as “app_reviews.csv”.

3) Amazon Product Review Data (CSV Source):

For representing consumer goods, I utilized a structured dataset obtained from Kaggle entitled “Amazon Sales Dataset.” It was composed of thousands of pre-scraped reviews from which I extracted and cleaned the relevant subset including product names, review headlines, review text, and ratings from 1–5 stars. I only considered rows with both the review text and valid numerical ratings. Following filtering and formatting, the data set yielded well above 1,500 clean, usable product reviews from across different consumer categories. This CSV data set formed the triad of data components by offering a stable, structured foundation for the sentiment and ratings from consumers. The cleaned, isolated data set is provided in the submission in the form of “product_reviews.csv”.

4) Final Combined Review Data (Integrated Across All Sources):

By cleaning and formatting the three separate datasets of Letterboxd (films), Google Play Store (applications), and Amazon (products), I unified them in one Pandas DataFrame for homogeneous analysis. This resulting aggregated dataset has more than 4,400 user reviews, each with its source, review text content, normalized star rating, and calculated sentiment score. This integrated format provided direct cross-domain comparisons alongside homogenous analysis through keyword-based sentiment scoring, polarity classification, mismatch detection, and statistical testing. Through standardized fields like product_name, rating, and review_body, the aggregated dataset formed the base for all downstream visualizations, tests, and inferences made from this endeavor. My submission includes the filtered, aggregated dataset as "combined_reviews.csv".

IV. Technical Solution

To examine sentiment-rating alignment across the various platforms, I created one integrated pipeline to gather, normalize, and compare user-review data from three data sources: Letterboxd (web scraping), Google Play API, and Kaggle Amazon reviews in CSV. All three data sets were processed within Python by following structured steps using similar formatting to allow cross-platform comparison.

First, I scraped reviews from Letterboxd by creating custom requests to review pages with requests and BeautifulSoup. I looped through multiple pages for each film to collect sufficient data and extracted both text reviews and star icons, which were passed through a custom parser to convert to float ratings. For mobile apps, I accessed the RapidAPI-hosted Google Play Store API, which provided structured JSON. I extracted text reviews and integer

star ratings, pre-formatted the results, and included source labels. The third dataset was loaded directly from a Kaggle CSV of Amazon product reviews and normalised to pull similar fields.

All the datasets were converted to Pandas DataFrame with the following standardized columns: product_name, review_body, rating, and source. Following the achievement of uniformity and the removal of rows with missing or incorrect data, the datasets were combined into one DataFrame for the analysis. A keyword-based sentiment scoring function was developed where integer values are given according to the occurrence of predefined positive and negative keywords, and sentiment polarity was classified into three levels: positive, neutral, and negative.

I calculated misalignment of sentiment and rating by creating a mismatch flag (e.g., positive sentiment with low rating). Trends were visualized using matplotlib and seaborn, including scatter plots, bar plots, pie plots, and box plots. Statistical significance of trends was tested using chi-square, ANOVA, and t-tests from the scipy.stats library.

One of the main challenges was to find the appropriate site to scrape product reviews from. Both Amazon.com and Walmart.com became significant hurdles: both sites' HTML was extremely dynamic and broken into a lot of pieces, making it hard to consistently pull user reviews from. On top of that, I was repeatedly blocked by both sites after attempting to automate queries with modified headers and sleep frequencies. This made me switch to using pre-scraped Amazon reviews from the Kaggle database to allow me to have consistency in reviewing format and go ahead with the analysis. Another challenge was to transform inconsistent star-based representations (such as “★★★★½”) into floats and to fine-tune the sentiment score threshold without using any outside sentiment libraries.

In order to make it both maintainable and extendable, all the processes in the pipeline were made module-based, well-documented, and recyclable, allowing the same workflow to be easily reused for any other review source and domain in the future.

V. Visualizations

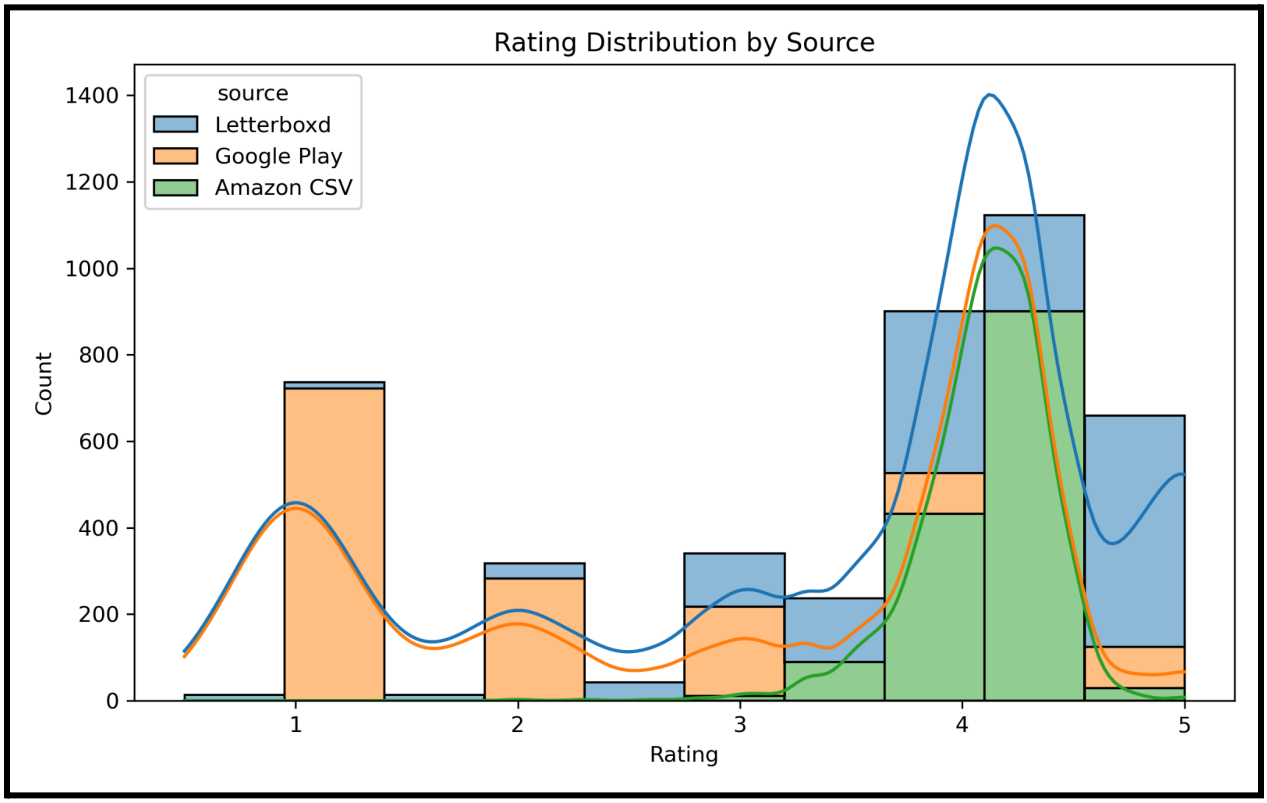

Figure 1: Rating Distribution by Source

This histogram illustrates the distribution of ratings across the three platforms. Amazon and Letterboxd have positively skewed distributions, whereas Google Play exhibits bimodality, with strong focus at both low levels (1-star) and top levels (5-stars). This bimodality may account for the increased mismatch frequency seen in Google Play data.

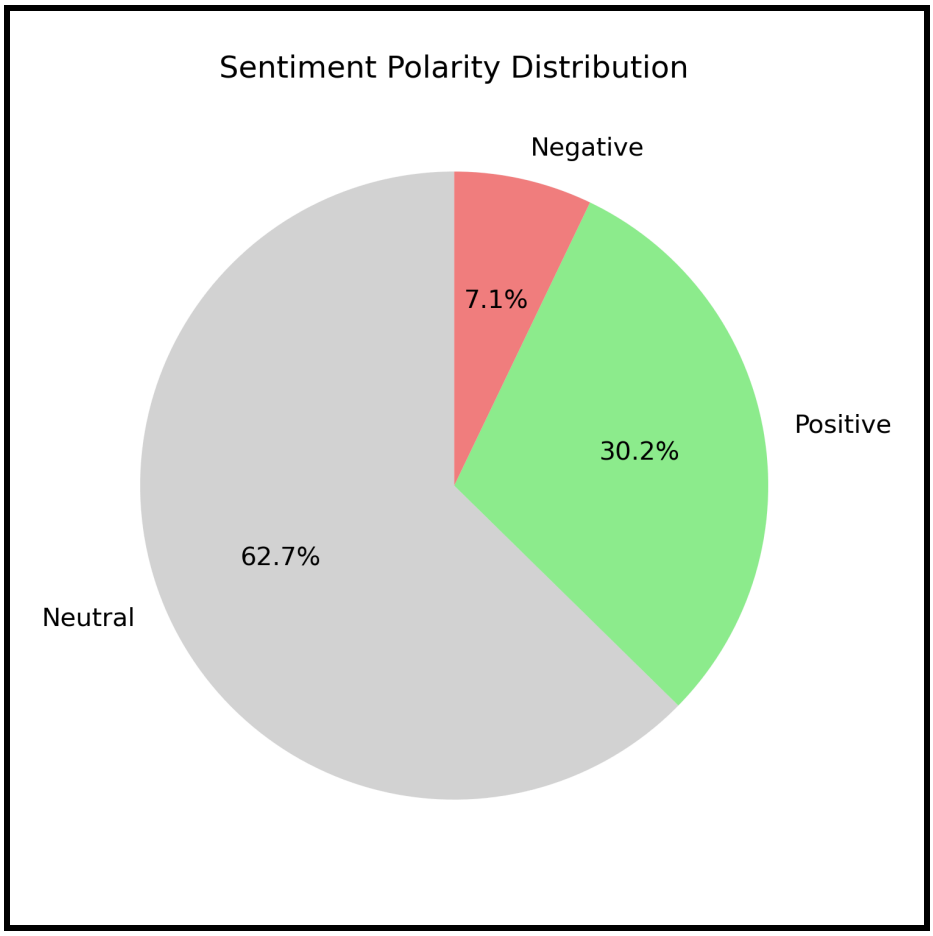

Figure 2: Sentiment Polarity Distribution

This pie chart illustrates the overall polarity of reviews according to keyword scoring. More reviews are categorized as neutral (62.7%), followed by positive reviews (30.2%), and then there is a small percentage of negative reviews (7.1%). This distribution implies reviews tend to lean neither to one extreme nor the other of the sentiment spectrum.

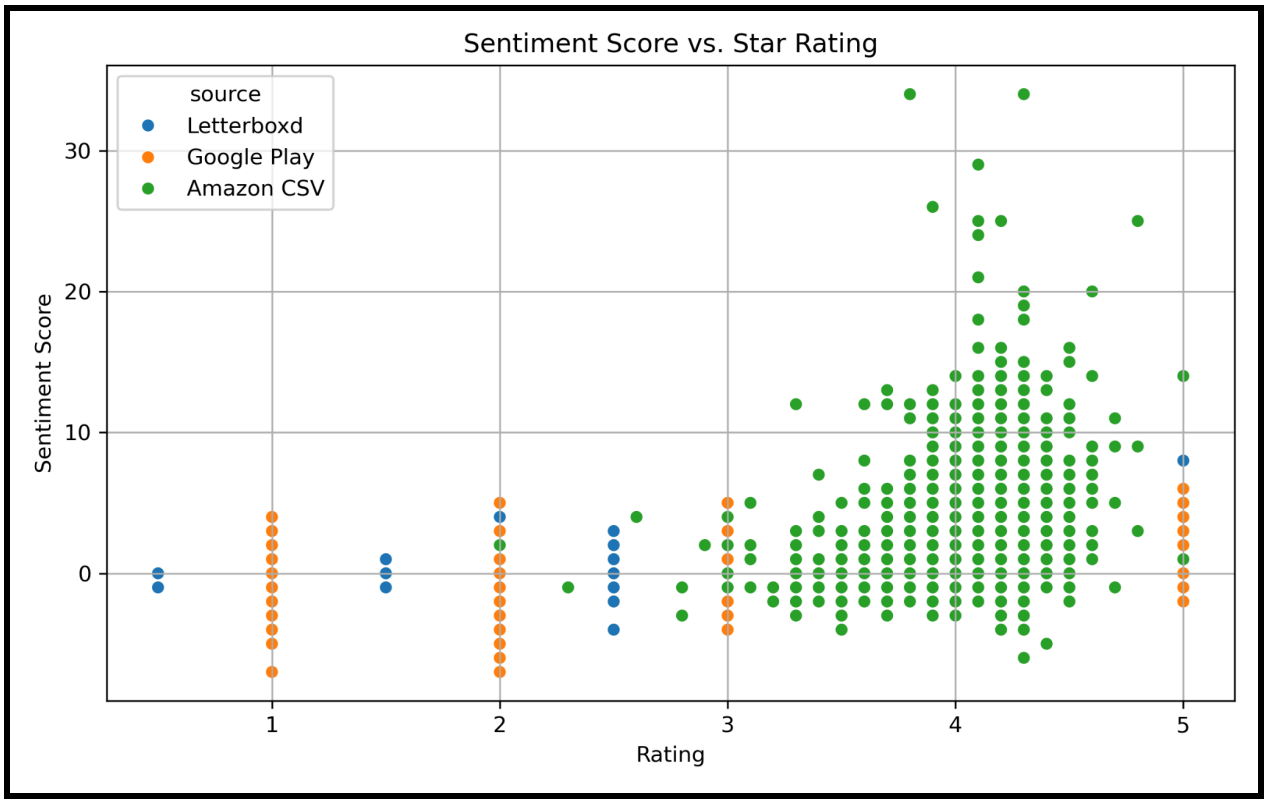

Figure 3: Sentiment Score vs. Star Rating

This scatterplot illustrates the relationship between numerical star ratings and calculated sentiment scores. A weak-to-moderate positive correlation is readily apparent, with Amazon reviews having the most density of high-rated reviews with rich sentiment. Outliers are also apparent, indicating where ratings likely do not represent user sentiment.

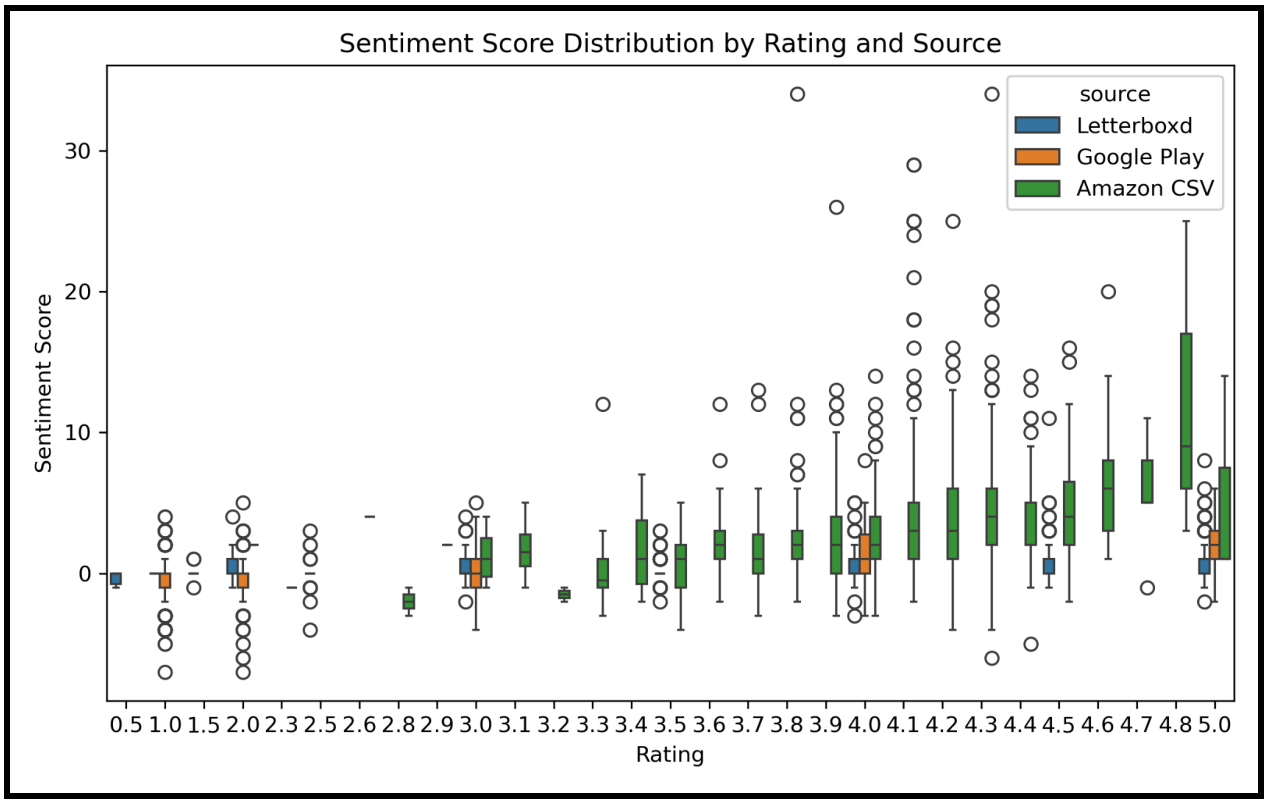

Figure 4: Sentiment Score Distribution by Rating and Source

This box plot illustrates the relative difference in sentiment scores as a function of different values of ratings along each source of reviews. In the plot, we see that higher ratings tend to have higher sentiment scores, but variability and outliers will be higher on some sites such as Amazon. This graph confirms our hypothesis of sentiment-text alignment differences in strength across levels of ratings and sites.

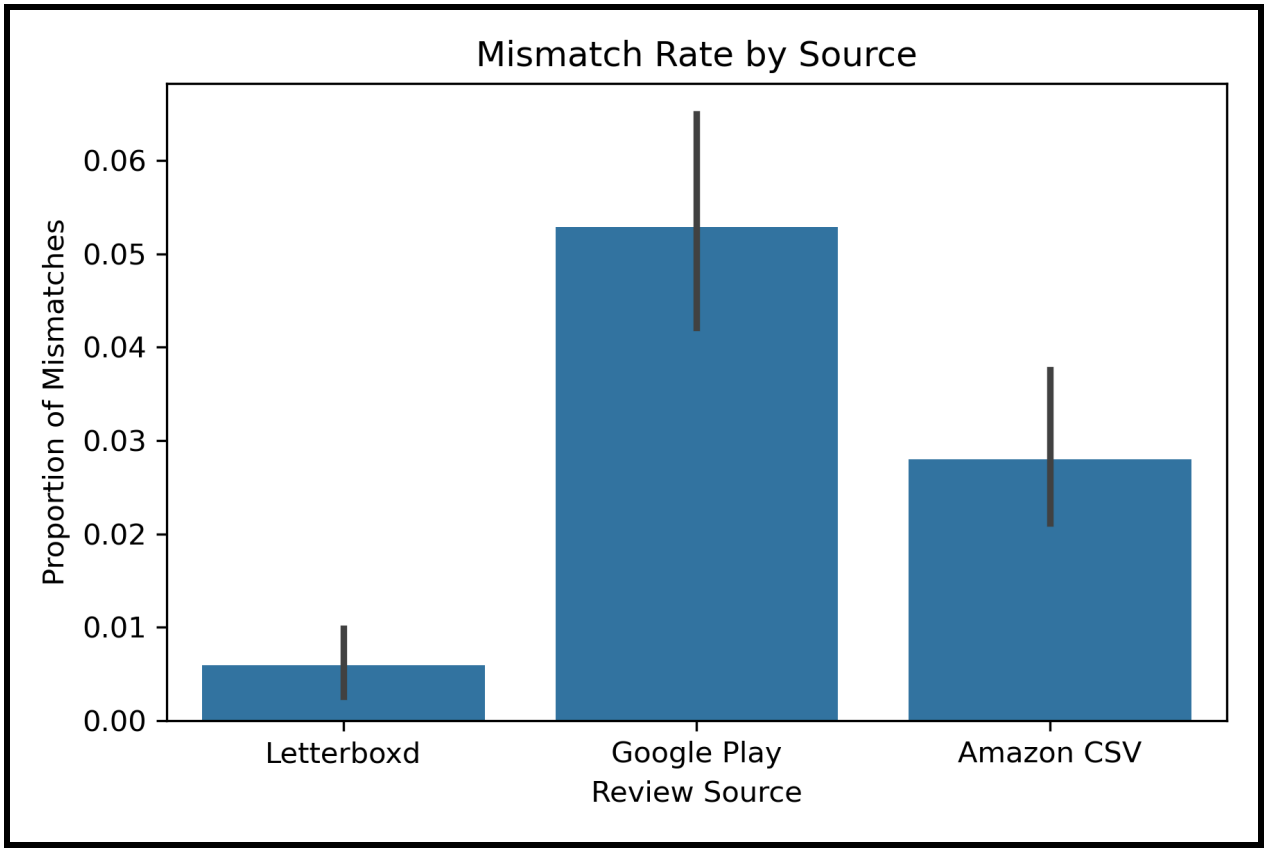

Figure 5: Mismatch Rate by Source

This bar plot represents the percentage of reviews where there is large disparity between the text sentiment and the numeric rating. Google Play has the highest mismatch percentage, followed by Amazon CSV, and the smallest is for Letterboxd. This discrepancy implies platform-specific conventions or difference in user behavior in assigning ratings.

VI. Analysis, Insights, & Conclusions

This project provided insightful findings concerning the correlation of numerical ratings and text sentiment in user reviews, identifying inconsistencies across various sites and content types. In what might have been expected to be the most straightforward correlation of all, higher ratings equaling positive sentiment and vice versa, this correlation did not always hold.

There was a moderate positive correlation of star ratings and overall sentiment scores, indicating some, but not perfect, alignment. A boxplot visual confirmed even at the same star rating level (i.e. 4 stars), there was great variability in the text sentiment, most prominently in Amazon reviews, reflecting platform-specific behavior and expectations of ratings.

The mismatch rate analysis validated such inconsistencies more explicitly. Whereas reviews at Letterboxd had the most consistent correspondence of ratings and sentiments (lowest mismatch percentage), Google Play contained the highest percentage of mismatches, followed by Amazon. This result was further validated by applying a chi-square test, and it reflected a significant statistical difference in mismatch proportions across the three sites. Platform-specific rating culture, differences in moderation or user interfaces, or user intentions might explain such differences. For instance, users might write bad ratings for an app due to bugs but positive text for the overall idea.

An ANOVA test revealed that sentiment scores themselves differ significantly by platform, supporting the conclusion that word and tone of reviews differ by domain and platform. The t-test for difference in sentiment score between mismatched and aligned reviews was also significant: mismatched reviews tended to have stronger (more extreme) sentiment than aligned reviews, and we conclude that users deviate from convention in ratings because when they do, they feel intensely, positively or negatively.

VII. Limitations & Future Work

This task depended on keyword-based sentiment scoring, which is simple to interpret but unable to pick up nuances like sarcasm, negation ("not good"), and domain-specific expressions. Additionally, due to the fact that Letterboxd and Amazon reviews were sampled from availability of CSV and scraping, sampling bias might have impacted the distributions.

The information was further restricted to U.S. reviews in English, which may have precluded international viewpoints or cross-cultural rating patterns. Anti-scraping protections via HTML formatting and bot-blocking technologies on high-traffic sites such as Amazon and Walmart presented obstacles to scraping that necessitated fallback approaches (that is, using a Kaggle dataset).

Future research could utilize a machine learning sentiment model (such as VADER, or transformer-based ones) to provide even richer results. It might also prove useful to examine temporal trends (e.g. how sentiment evolves over time) or topic modeling to classify the kinds of attributes or problems users discuss in incompatible reviews.