Book Sentiment Lookup Tool

Compare any book's sentiment to its genre average

This tool analyzes reader sentiment for any book in the Amazon Books dataset and compares it to the genre baseline.

1. Setup (Run Once)

In [1]:
```
import pandas as pd
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import warnings
warnings.filterwarnings('ignore')

# Initialize VADER
vader = SentimentIntensityAnalyzer()

def get_sentiment(text):
    try:
        return vader.polarity_scores(str(text))['compound']
    except:
        return 0

print("Loading data...")
books = pd.read_csv('books_data.csv')
ratings = pd.read_csv('Books_rating.csv')

# Merge
df = ratings.merge(books[['Title', 'categories']], on='Title', how='inner')
df = df.dropna(subset=['review/text', 'categories'])

# Extract genre
def extract_genre(cat_str):
    if pd.isna(cat_str):
        return None
    cat_str = str(cat_str).replace("['", "").replace("']", "")
    return cat_str.split("',")[0].strip()

df['genre'] = df['categories'].apply(extract_genre)
print(f"Loaded {len(df):,} reviews across {df['genre'].nunique()} genres")
```

```
Loading data...
Loaded 2,448,496 reviews across 10883 genres
```

In [2]:
```
# Compute genre baselines (takes ~1-2 minutes)
print("Computing genre baselines...")

genre_stats = {}
for genre in df['genre'].unique():
    if pd.isna(genre):
        continue
    genre_reviews = df[df['genre'] == genre]

    # Sample if too many
    if len(genre_reviews) > 5000:
        genre_reviews = genre_reviews.sample(n=5000, random_state=42)

    if len(genre_reviews) < 10:
        continue

    sentiments = genre_reviews['review/text'].apply(get_sentiment)
    genre_stats[genre] = {
        'mean': sentiments.mean(),
        'std': sentiments.std(),
        'count': len(df[df['genre'] == genre])
    }

print(f"Done! Baselines computed for {len(genre_stats)} genres.")
print("\nReady to look up books!")
```

```
Computing genre baselines...
Done! Baselines computed for 3121 genres.

Ready to look up books!
```

2. Book Lookup

Change the title below and run the cell to analyze any book:

In [33]:
```
#############################################
# CHANGE THIS TITLE TO LOOK UP ANY BOOK
#############################################

BOOK_TITLE = "Malazan"
#############################################

# Check if book exists in dataset
matches = df[df['Title'].str.contains(BOOK_TITLE, case=False, na=False)]
if len(matches) == 0:
    print(f"NOT FOUND: No books matching '{BOOK_TITLE}' in the dataset.")
    print("Try a different title or check your spelling.")
else:
    unique_titles = matches['Title'].value_counts()
    print(f"FOUND: {len(unique_titles)} book(s) matching '{BOOK_TITLE}'")
    print(f"Top match: \"{unique_titles.index[0]}\" ({unique_titles.iloc[0]:,} revi
    print("\nRun the next cell for full analysis.")
```

```
FOUND: 2 book(s) matching 'Malazan'
Top match: "Deadhouse Gates : A Tale of Malazan Book of the Fallen" (132 reviews)

Run the next cell for full analysis.
```

In [34]:
```
# Find matching books
matches = df[df['Title'].str.contains(BOOK_TITLE, case=False, na=False)]
```

```
if len(matches) == 0:
    print(f"No books found matching '{BOOK_TITLE}'")
else:
    # Show matches
    unique_titles = matches['Title'].value_counts()
    print(f"Found {len(unique_titles)} matching titles:")
    for title, count in unique_titles.head(10).items():
        print(f"  - {title} ({count} reviews)")

    # Analyze top match
    title = unique_titles.index[0]
    book_reviews = matches[matches['Title'] == title].copy()
    genre = book_reviews['genre'].iloc[0]

    print(f"\n{'='*60}")
    print(f"ANALYZING: {title}")
    print(f"{'='*60}")
    print(f"Genre: {genre}")
    print(f"Reviews: {len(book_reviews):,}")

    # Calculate sentiment
    book_reviews['sentiment'] = book_reviews['review/text'].apply(get_sentiment)
    book_mean = book_reviews['sentiment'].mean()
    avg_stars = book_reviews['review/score'].mean()

    print(f"\nAverage stars: {avg_stars:.1f} / 5.0")
    print(f"Sentiment score: {book_mean:.3f}")

    # Genre comparison
    if genre in genre_stats:
        genre_mean = genre_stats[genre]['mean']
        genre_count = genre_stats[genre]['count']
        diff = book_mean - genre_mean

        print(f"\n--- GENRE COMPARISON ---")
        print(f"Genre '{genre}' has {genre_count:,} reviews")
        print(f"Book sentiment:  {book_mean:+.3f}")
        print(f"Genre average:   {genre_mean:+.3f}")
        print(f"Difference:      {diff:+.3f}")

        if diff > 0.1:
            print(f"\n>>> OUTPERFORMING genre by {diff:.2f} points")
        elif diff < -0.1:
            print(f"\n>>> UNDERPERFORMING genre by {abs(diff):.2f} points")
        else:
            print(f"\n>>> AVERAGE for genre")

    # Breakdown
    pos = (book_reviews['sentiment'] > 0.1).mean() * 100
    neg = (book_reviews['sentiment'] < -0.1).mean() * 100
    neu = 100 - pos - neg
    print(f"\n--- BREAKDOWN ---")
    print(f"Positive: {pos:.1f}%")
    print(f"Neutral:  {neu:.1f}%")
    print(f"Negative: {neg:.1f}%")
```

```
# Show most positive and negative reviews for the book
if len(matches) > 0 and 'sentiment' in book_reviews.columns:
    print("MOST POSITIVE REVIEW:")
    top_pos = book_reviews.nlargest(1, 'sentiment')
    print(f"Sentiment: {top_pos['sentiment'].iloc[0]:.2f}")
    print(f"Stars: {top_pos['review/score'].iloc[0]}")
    print(f"\n\"{top_pos['review/text'].iloc[0][:500]}...\"")

    print("\n" + "="*60)

    print("\nMOST NEGATIVE REVIEW:")
    top_neg = book_reviews.nsmallest(1, 'sentiment')
    print(f"Sentiment: {top_neg['sentiment'].iloc[0]:.2f}")
    print(f"Stars: {top_neg['review/score'].iloc[0]}")
    print(f"\n\"{top_neg['review/text'].iloc[0][:500]}...\"")
```

```
Found 2 matching titles:
  - Deadhouse Gates : A Tale of Malazan Book of the Fallen (132 reviews)
  - Reaper's Gale: Malazan Book of the Fallen #7: (UK Hardcover Steven Erikson) (16
reviews)

============================================================
ANALYZING: Deadhouse Gates : A Tale of Malazan Book of the Fallen
============================================================
Genre: Fiction
Reviews: 132

Average stars: 4.3 / 5.0
Sentiment score: 0.535

--- GENRE COMPARISON ---
Genre 'Fiction' has 824,437 reviews
Book sentiment:  +0.535
Genre average:   +0.566
Difference:      -0.031

>>> AVERAGE for genre

--- BREAKDOWN ---
Positive: 81.1%
Neutral:  1.5%
Negative: 17.4%
MOST POSITIVE REVIEW:
Sentiment: 1.00
Stars: 5.0

"While Gardens of the Moon was, though a great book in its own right, the worst book
in this series, Deadhouse Gates is one of the best. However, Deadhouse Gates is also
one of the two most difficult reads of the series. Not only is it long, 943 pages, b
ut it is extremely heavy reading. I won't give you any details, don't want to spoil
the book, but the story follows a small army of refugees known as the Chain of Dogs,
as they attempt to make their way across a continent in the midst of rebellion..."

============================================================

MOST NEGATIVE REVIEW:
Sentiment: -1.00
Stars: 1.0

"Though Gardens of the Moon was tripe, oftentimes a sequel can redeem a failure's le
gacy. Unfortunately things get worse. Gardens of the Moon was childish in how much m
agic there was and every main character was important based on their power levels an
d their affinity with Flat Random God A. But that is Shakespearean compared to this
moronic excuse of a story.The story involves the Seven Cities Rebellion, known as th
e Whirlwind, announced at the end of Gardens of the Moon which is rising to chal
l..."

Average stars: 4.3 / 5.0
Sentiment score: 0.535

--- GENRE COMPARISON ---
Genre 'Fiction' has 824,437 reviews
```

```
Book sentiment:  +0.535
Genre average:   +0.566
Difference:      -0.031

>>> AVERAGE for genre

--- BREAKDOWN ---
Positive: 81.1%
Neutral:  1.5%
Negative: 17.4%
MOST POSITIVE REVIEW:
Sentiment: 1.00
Stars: 5.0

"While Gardens of the Moon was, though a great book in its own right, the worst book
in this series, Deadhouse Gates is one of the best. However, Deadhouse Gates is also
one of the two most difficult reads of the series. Not only is it long, 943 pages, b
ut it is extremely heavy reading. I won't give you any details, don't want to spoil
the book, but the story follows a small army of refugees known as the Chain of Dogs,
as they attempt to make their way across a continent in the midst of rebellion..."

============================================================

MOST NEGATIVE REVIEW:
Sentiment: -1.00
Stars: 1.0

"Though Gardens of the Moon was tripe, oftentimes a sequel can redeem a failure's le
gacy. Unfortunately things get worse. Gardens of the Moon was childish in how much m
agic there was and every main character was important based on their power levels an
d their affinity with Flat Random God A. But that is Shakespearean compared to this
moronic excuse of a story.The story involves the Seven Cities Rebellion, known as th
e Whirlwind, announced at the end of Gardens of the Moon which is rising to chal
l..."
```