

---

# **SI 201**

# **Final Presentation**

**Presented by :**

**I Want To Go Home: Chih-Hsiang Chang & Xiwen Mark**

# Today's Agenda

- 01** Introduction
- 02** Key Questions
- 03** APIs and Websites Sources
- 04** Gathering Data and Database
- 05** Calculation and Visualization
- 06** Conclusion
- 07** Challenges
- 08** Resources

A close-up, low-angle shot of a vintage-style film projector. Two large, dark metal reels are visible on the left, with a thick black cable running between them. The projector body is made of dark wood or metal with various knobs, screws, and a small red button. A bright beam of light is projected from a small circular opening on the right side. The background is a soft-focus blue.

# Introduction

This project is aim to gather data from one or more APIs and a website (with Beautiful Soup) to answer questions about the preferences and correlations between books and their movie adaptations.

# Goals

---

**Initial:**

**Identify the preferences and correlations between books and their movie adaptations using two APIs.**

**Final:**

**Added an additional website to scrape a list of books that have been made into movies**

# Key Questions



## PROBLEM 1

### Book vs Movie Preferences

Do people like the movie adaptions more?

Or do people love their original book compositions...?



## PROBLEM 2

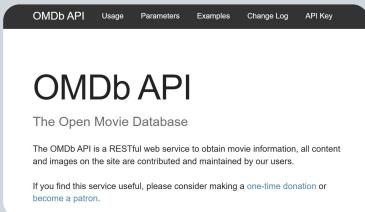
### Book vs Movie Ratings Correlation

Do highly rated books lead to highly rated movies?

# APIs & Web

Source 02

Source 03



As seen in the homeworks...

**-The almighty OMDB API!**

Google Books APIs

goodreads

Google Books API

Goodreads Website

# Gathering Data and Database

- Final Database in DB Browser

title_id	title
289	290 Becoming Jane Austen
290	291 Fantastic Voyage
291	292 Birdsong
292	293 Rocket Boys
293	294 Neverwhere
294	295 Blue Dog
295	296 Wonder Boys
296	297 Somewhere In Time
297	298 A Monster Calls
298	299 A Child's Christmas in Wales
299	300 Puberty Blues

Titles

book_id	title_id	book_rating	ratings_count
239	239	289	NULL
240	240	291	4.0
241	241	292	NULL
242	242	293	2.5
243	243	294	4.0
244	244	295	5.0
245	245	296	NULL
246	246	297	NULL
247	247	298	5.0
248	248	299	4.0
249	249	300	NULL

Books

movie_id	title_id	movie_rating	movie_count
239	239	289	7.3
240	240	291	6.8
241	241	292	7.3
242	242	293	8.8
243	243	294	7.2
244	244	295	6.1
245	245	296	7.2
246	246	297	7.2
247	247	298	7.4
248	248	299	7.7
249	249	300	8.2

Movies

failed_id	title_id
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50

Failed Titles

# Gathering Data and Database

- Preventing Duplicates

```
if gb_data is None:  
    print("  Skipping: no valid Google Books data.")  
    cur = conn.cursor()  
    cur.execute(  
        "INSERT OR IGNORE INTO FailedTitles (title_id) VALUES (?)",  
        (title_id,)  
    )  
  
    conn.commit()  
    continue
```

```
if omdb_data is None:  
    print("  Skipping: no valid OMDb data.")  
    cur = conn.cursor()  
    cur.execute(  
        "INSERT OR IGNORE INTO FailedTitles (title_id) VALUES (?)",  
        (title_id,)  
    )  
  
    conn.commit()  
    continue
```

Table: FailedTitles

	failed_id	title_id
	Filter	Filter
40	40	225
41	41	231
42	42	235
43	43	238
44	44	247
45	45	258
46	46	261
47	47	270
48	48	280
49	49	287
50	50	290

40 - 50 of 50

# Gathering Data and Database

- Integer Key (title\_id)

Table: Titles

	title_id	title
289	290	Becoming Jane Austen
290	291	Fantastic Voyage
291	292	Birdsong
292	293	Rocket Boys
293	294	Neverwhere
294	295	Blue Dog
295	296	Wonder Boys
296	297	Somewhere In Time
297	298	A Monster Calls
298	299	A Child's Christmas in Wales
299	300	Puberty Blues

◀ ▶ 289 - 299 of 299 ▶ ▷

Titles

Table: Books

	book_id	title_id	book_rating	ratings_count
239	239	289	NULL	NULL
240	240	291	4.0	8
241	241	292	NULL	NULL
242	242	293	2.5	2
243	243	294	4.0	1
244	244	295	5.0	1
245	245	296	NULL	NULL
246	246	297	NULL	NULL
247	247	298	5.0	1
248	248	299	4.0	1
249	249	300	NULL	NULL

◀ ▶ 239 - 249 of 249 ▶ ▷

Books

Table: Movies

	movie_id	title_id	movie_rating	movie_count
239	239	289	7.3	35895
240	240	291	6.8	21970
241	241	292	7.3	6073
242	242	293	8.8	19708
243	243	294	7.2	3178
244	244	295	6.1	121
245	245	296	7.2	67814
246	246	297	7.2	35584
247	247	298	7.4	97272
248	248	299	7.7	661
249	249	300	8.2	2295

◀ ▶ 239 - 249 of 249 ▶ ▷

Movies

# Gathering Data and Database

- Limits to 25 data stored

```
def load_batch(conn, max_new=25):
    """
    Load up to max_new NEW adaptations in one run.
    This enforces the 25-items-per-run rule from the project spec.
    """

    candidate_rows = get_pending_titles(conn, max_new)

    inserted = 0
    for title_id, title in candidate_rows:
        if inserted >= max_new:
            break

        print(f"Processing '{title}' (title_id={title_id}) ...")
```

# Select Data & JOIN

---

JOIN if Book and Movie has the same title\_id

```
def fetch_joined_data(conn):
    """
    Get joined rows: one per book-movie pair.
    """
    cur = conn.cursor()
    cur.execute("""
        SELECT
            T.title,
            B.book_rating,
            B.ratings_count,
            M.movie_rating,
            M.movie_count
        FROM Titles AS T
        JOIN Books AS B ON T.title_id = B.title_id
        JOIN Movies AS M ON T.title_id = M.title_id
        WHERE B.book_rating IS NOT NULL
            AND M.movie_rating IS NOT NULL
    """)
    rows = cur.fetchall()

    data = []
    for row in rows:
        data.append({
            "title": row[0],
            "book_rating": row[1],
            "book_count": row[2],
            "movie_rating": row[3],
            "movie_count": row[4],
        })

    return data
```

# Calculation

```
def compute_preference_counts(data):
    books_better = 0
    movies_better = 0
    ties = 0
    total = 0

    for row in data:
        book_rating = row.get("book_rating")
        movie_rating_10 = row.get("movie_rating")
        movie_rating_5 = convert_movie_rating(movie_rating_10)

        if book_rating is None or movie_rating_5 is None:
            continue

        total += 1
        if movie_rating_5 > book_rating:
            movies_better += 1
        elif book_rating > movie_rating_5:
            books_better += 1
        else:
            ties += 1

    return books_better, movies_better, ties, total
```

```
def preference_percentage(preference_counts):
    books_better, movies_better, ties, total = preference_counts

    if total == 0:
        return (0, 0, 0)

    books_better_pct = books_better / total
    movies_better_pct = movies_better / total
    ties_pct = ties / total
    return books_better_pct, movies_better_pct, ties_pct
```

## Question 1:

Percentage of Book Preferred and Movie Preferred

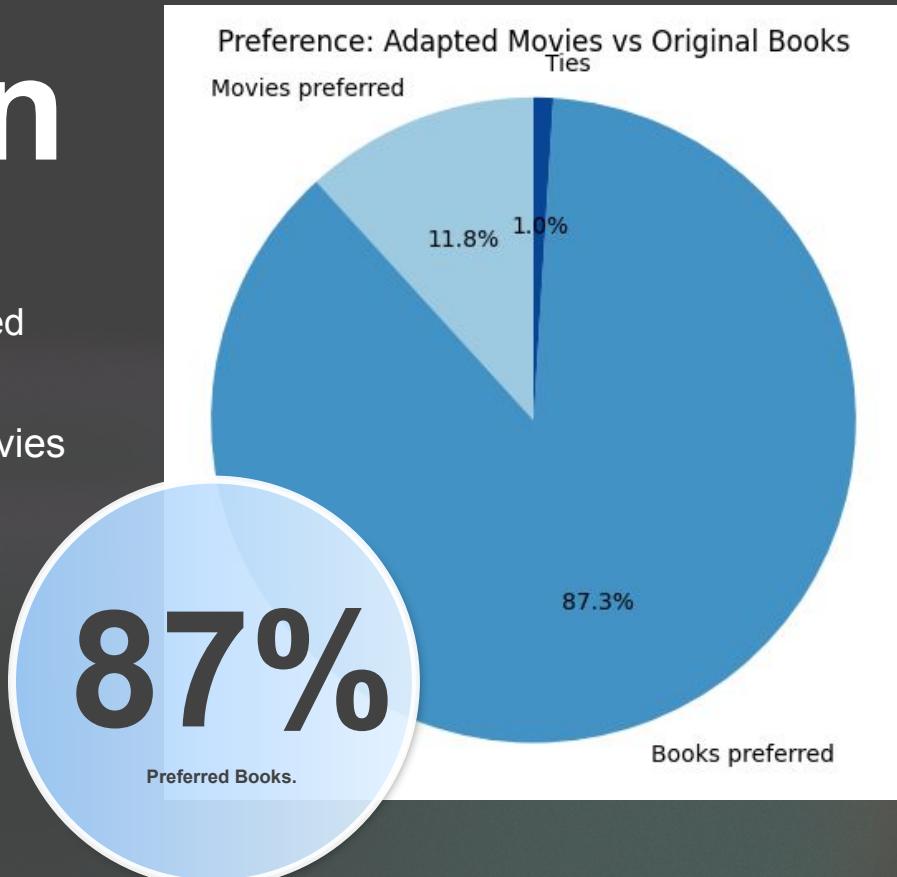
# Visualization

## Question 1:

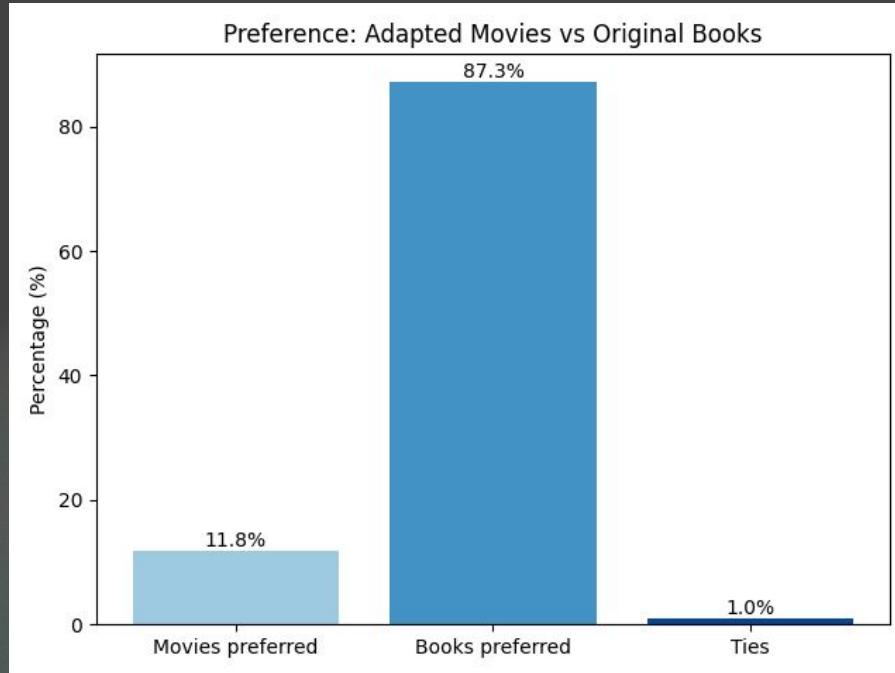
Percentage of Book Preferred and Movie Preferred

- 87.3% → preference for books over movies
- 11.8% → preferring movies
- 1% → tie.

This suggests that the majority of people prefer the book over the movie in terms of book-to-movie adaptations.



# Visualization



**Question 1:**  
Percentage of Book Preferred  
and Movie Preferred

# Calculation

```
def preference_percentage(preference_counts):
    books_better, movies_better, ties, total = preference_counts

    if total == 0:
        return (0, 0, 0)

    books_better_pct = books_better / total
    movies_better_pct = movies_better / total
    ties_pct = ties / total
    return books_better_pct, movies_better_pct, ties_pct
```

**Question 2:**  
Books and Movies Rating correlations

```
def pearson_correlation(x, y):
    if len(x) != len(y) or len(x) < 3:
        return None

    r, p = pearsonr(np.array(x), np.array(y))
    return r, p

def linear_regression(x, y):
    if len(x) != len(y) or len(x) < 3:
        return None

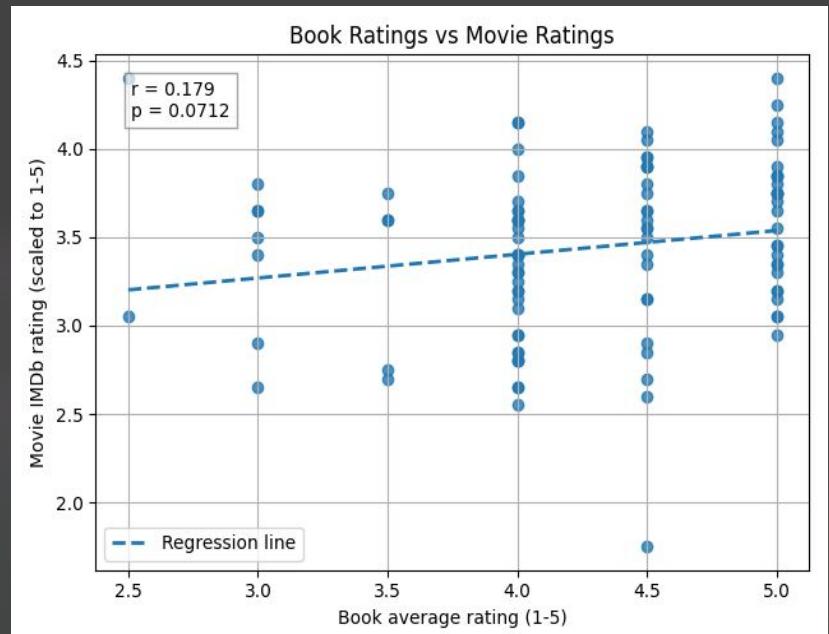
    result = linregress(np.array(x), np.array(y))
    return {
        "slope": result.slope,
        "intercept": result.intercept,
        "r_value": result.rvalue,
        "p_value": result.pvalue,
        "stderr": result.stderr
    }
```

# Visualization

## Question 2:

Books and Movies Rating correlations

- The scatter plot shows a slightly upward trend
- Correlation coefficient of 0.179 and a p-value of 0.07 → correlation between the two variables is weak and statistically non-significant.

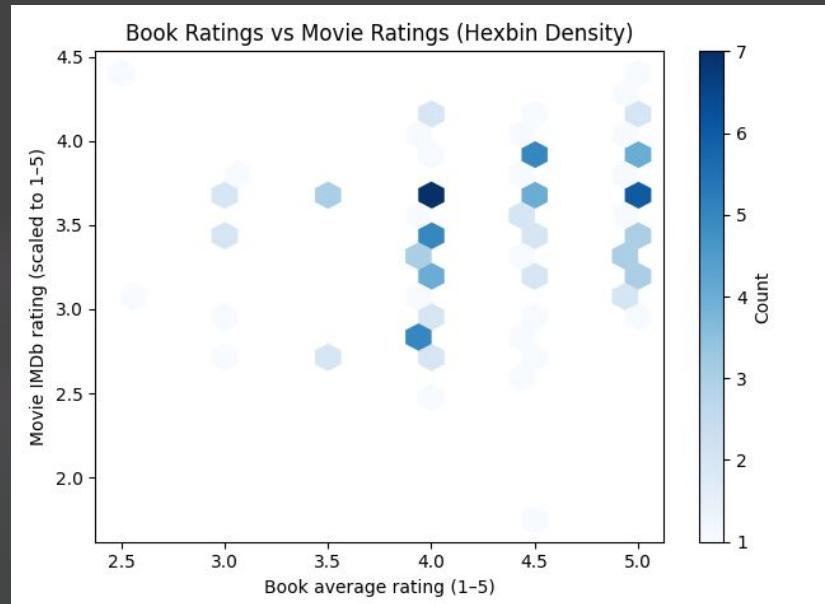


# Visualization

## Question 2:

Books and Movies Rating correlations

- Higher-rated books tend to have more movie adaptations.
- Higher book rating does not guarantee a higher movie adaptation rating.
- Adaptations rarely outperform low-rated books.
- Books typically receive slightly higher ratings than their movie adaptations.



# Challenges

---

**Chih-Hsiang:**

- New API
- Structures
- Approach

**Xiwen:**

- Relational Database
- Calculation Threshold

---

# Thank You

Presented by :  
**Chih-Hsiang Chang & Xiwen Mark**  
(and I actually want to go home)