Movie Popularity and Revenue Analysis Data using PySpark

Bhanu Venkata Siva Kumar Gorantla, Khalida Parveen Mohammad, Krishna Vasanthi Yakkala, Malaya Sugandhini Seerapu

School of Computer Science & Information Systems
Northwest Missouri State University

Abstract

This movie analysis looks at a large dataset of films in order to cast the spotlight on the elements that make a film successful. Movie-related details including budget, release date, income, genres, popularity, and average votes are all included in the dataset. The project will investigate patterns and connections between these characteristics and measurements of film performance, including box office receipts and popularity ratings.

The research will also look at important factors including production languages, genre preferences, and release dates in order to spot trends that affect audience reaction and financial results. Through the use of PySpark for effective data processing and visualization tools such as Power BI or Tableau, this project aims to provide stakeholders in the film industry with practical suggestions that will facilitate better decision-making for next projects and marketing plans.

1 Introduction

The "Movie Popularity and Revenue Analysis" project focuses on analyzing a rich dataset containing information about movies and their performance metrics. This dataset includes details such as budgets, release dates, genres,

popularity scores, and average audience votes. The primary goal is to uncover insights into what makes a movie successful in terms of popularity and revenue generation.

The dataset allows for an in-depth exploration of relationships between critical attributes like budget and revenue, genre and audience preferences, and the impact of release timing on success. Through the use of PySpark, a distributed data processing framework, the project handles large-scale data efficiently to ensure accurate and scalable analysis.

By offering valuable insights into trends and patterns in the movie industry, this project aims to assist producers, marketers, and other stakeholders in making data-driven decisions. Additionally, the visualizations and recommendations derived from the analysis will serve as a guide for optimizing movie production and distribution strategies.

2 Project Idea

This project's objective is to use PySpark to evaluate a dataset that includes comprehensive movie information, such as budget, release date, revenue, genres, popularity, and average votes. In order to gain a better knowledge of the trends and metrics that affect a film's performance, this analysis attempts to reveal insights into the elements that affect film success.

The dataset makes it possible to investigate the connections between variables like budget, genres, and production languages and results like popularity and revenue. We can effectively handle and analyze this dataset by utilizing PySpark, a distributed data processing framework, especially when working with large-scale data operations.

In order to assist those involved in the film industry in making wise decisions, the analysis attempts to offer important insights into film production and marketing tactics. Identifying the genres with the highest average revenue or knowing how release dates affect performance, for instance, might help guide future film planning and marketing initiatives.

PySpark is the best option for managing this dataset because of its features, which enable high-speed processing and scalability.

2.1 Top 10000 Popular Movies Dataset

The dataset represents information about movies, with details such as original-title (the name of the movie), release-date (the date it was released), genre (the category or categories it belongs to, such as Action or Comedy), revenue (the total earnings generated by the movie), runtime (duration in minutes), and vote-average (average rating given by viewers). By filtering based on the extracted release-year, we can analyze and retrieve movies released in a specific year, providing insights into their performance and characteristics.

3 Tools and Technologies

1. PySpark - for data processing and analysis
2. Pandas - for data cleaning and manipulation
3. Power BI or Tableau - for data visualization
4. Jupyter Notebook - for coding and analysis
5. Kaggle - for accessing and downloading the dataset
6. Git - for code management and collaboration

4 High-Level Architecture and Methodology

5 Explanation of the block diagram

The methodology of the project is as follows:

1. **Data Gathering**
   - Access and download the movie dataset from Kaggle.
   - Use Git for version control and collaboration during development.
2. **Data Pre-Processing**
   - Load the dataset into Jupyter Notebook.
   - Utilize PySpark and Pandas to clean and manipulate the data.

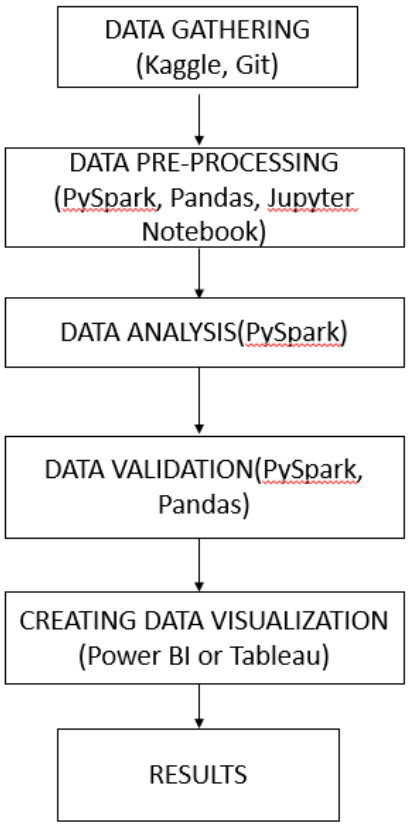

Figure 1: High-Level Architecture of the Project

- Address missing values, standardize formats, and prepare the data for analysis.

3. **Data Analysis**

   - Leverage PySpark for processing large-scale data efficiently.
   - Extract insights like top-grossing movies, popularity trends, and average revenue by genre.

4. **Data Validation**

   - Ensure data consistency and accuracy with PySpark and Pandas validation checks.
   - Validate the correctness of the extracted insights.

5. **Data Visualization**
    - Use tools like Power BI or Tableau to create visual representations of the findings.
    - Generate charts to illustrate trends in revenue, popularity, and other metrics.
6. **Results**
    - Marks the conclusion of the project with all objectives met.

6 Goals

1. List the top ten highest-grossing films together with their titles and earnings.
2. Display movies in descending order by popularity score.
3. Calculate the total number of movies produced in each language.
4. Fetch movies' names with the longest runtime.
5. Determine the Vote Count Per Language.
6. Determine the Vote Average Per Language.
7. Determine the top 10 Original Titles with Largest Tagline.
8. Determine the Highest Revenue Film in Each Year.
9. Retrieve all movies released in a given year.

7 Results Summary

7.1 List the top ten highest-grossing films together with their titles and earnings.

The ten highest-grossing films were determined by analyzing the dataset. This required arranging the films in descending order by the income column. As a reflection of their commercial success, the results featured well-known

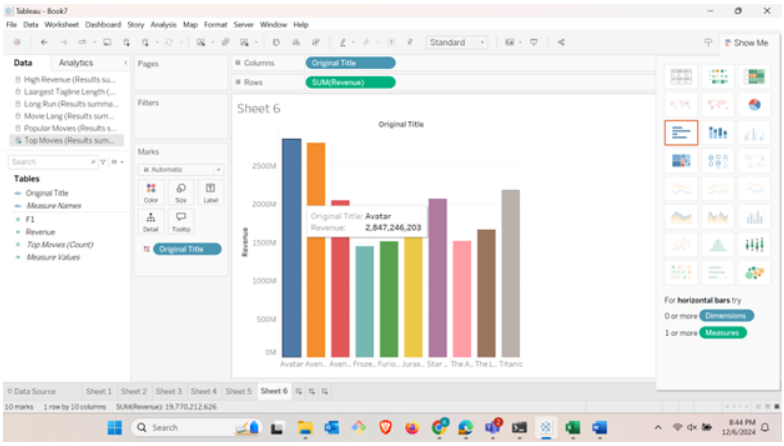

Figure 2: The top ten highest-grossing films together with their titles and earnings.

movies like Avengers: Endgame and Avatar. These observations aid in the understanding of trends in popular genres and films.This bar chart listing the top-performing movies based on metrics like revenue.The best-performing movies are shown here based on factors like revenue. In order to identify the most influential films in the dataset, this chart highlights the top titles.

```
topMovies = spark.sql("""
    SELECT original_title, revenue
    FROM movies
    ORDER BY revenue DESC
    LIMIT 10
""")
topMovies.show()
```

7.2 Display movies in descending order by popularity score.

Using the popularity column to sort the dataset brought to light the most watched movies, including Venom: Let There Be Carnage and Eternals. This data is useful for researching audience preferences and the results of marketing campaigns.This visualization ranks movies by popularity. Titles like Venom and Eternals appear prominently, offering insights into audience pref-

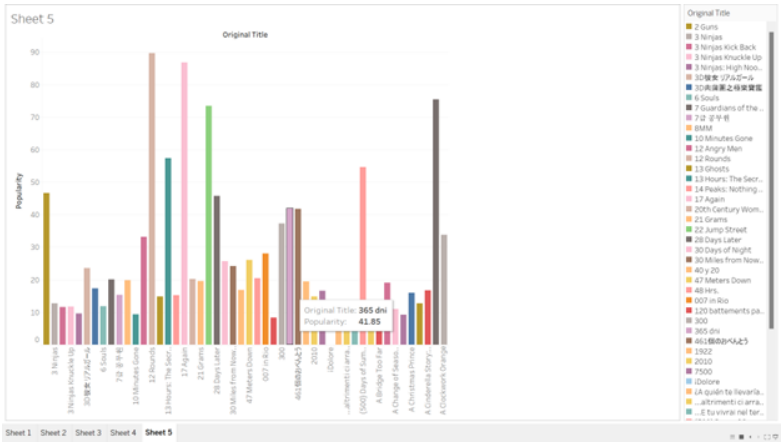

Figure 3: Displaying movies in descending order by popularity score.

erences and the influence of marketing campaigns.Focuses on movies ranked by their popularity scores. Longer bars indicate higher popularity.

```
popular_movies = spark.sql("""
    SELECT original_title, popularity
    FROM movies
    ORDER BY popularity DESC
""")
popular_movies.show()
```

7.3 Calculate the total number of movies produced in each language.

The diversity of cinema production was revealed by counting and grouping films according to the original-language column. Although English was the most common language in the sample, Hindi and French were also well-represented, demonstrating how international the film business is.The quantity of films made in each language is represented by the bubble sizes in this graphic, which highlights linguistic diversity in movies. English is the most common language, but smaller yet equally important concentrations of Hindi and Spanish are also present.

```
movies_per_language = spark.sql("""
    SELECT original_language, COUNT(*) AS
```

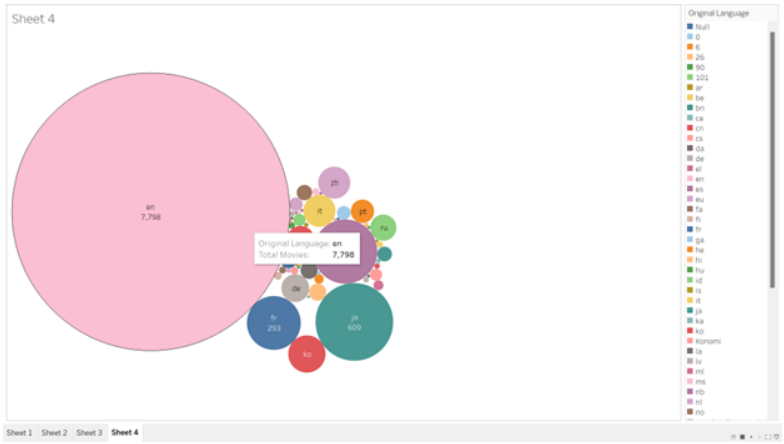

Figure 4: The total number of movies produced in each language.

```
    total_movies
    FROM movies
    GROUP BY original_language
    ORDER BY total_movies DESC
""")
movies_per_language.show()
```

7.4 Fetch movies' names with the longest runtime.

Movies having the longest durations were found by sorting them by the runtime column. Notable titles that shed light on the relationship between runtime and spectator involvement were Gone with the Wind.The bubble chart provides a visual representation of movies with the longest runtime. Each bubble's size reflects a particular metric, such as popularity or runtime, helping identify outliers and trends in runtime preferences.

```
longest_runtime_movies = spark.sql("""
    SELECT original_title, runtime
    FROM movies
    WHERE runtime IS NOT NULL
    ORDER BY runtime DESC
""")
longest_runtime_movies.show()
```

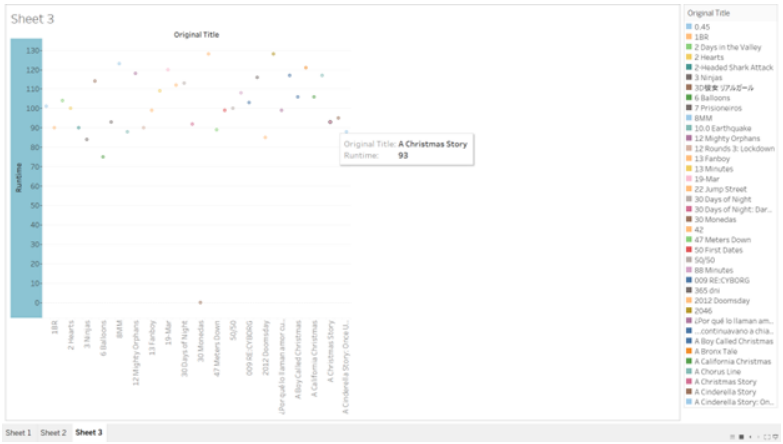

Figure 5: Movies' with the longest runtime.

7.5 Determine the Vote Count Per Language.

Vote-count was computed once the dataset was sorted by original-language. Vote counts were higher for languages like English and Spanish, suggesting that films in these languages had a wider audience and greater audience participation.This chart visualizes the total votes across movies, revealing audience engagement levels. High vote counts for popular films indicate widespread viewership and interest.

```
vote_count_per_language = spark.sql("""
    SELECT original_language, SUM(vote_count) AS total_votes
    FROM movies
    GROUP BY original_language
    ORDER BY total_votes DESC
""")
vote_count_per_language.show()
```

7.6 Determine the Vote Average Per Language.

Based on viewer ratings, the average vote-average for each language was calculated to evaluate the films' quality. As a result of their critical acclaim, films in languages like Korean and Japanese frequently received high ratings.A chart showing the average number of people who watch movies, broken down by genre or language. High ratings for genres like drama or

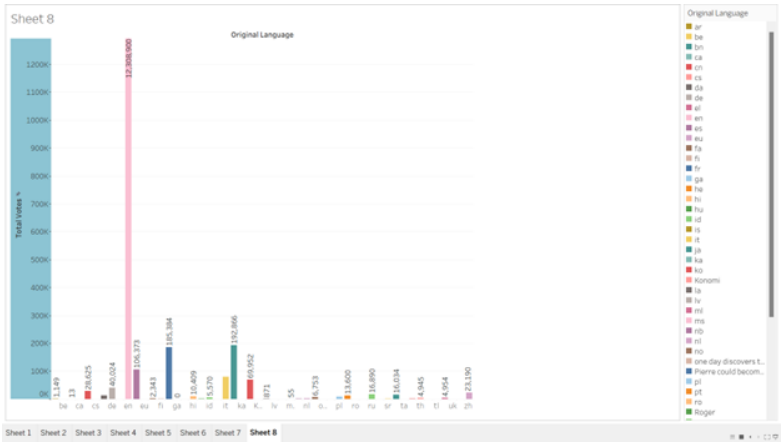

Figure 6: Displaying the Vote Count Per Language.

languages like Korean highlight their praise from critics.

```
vote_average_per_language = spark.sql("""
    SELECT original_language, AVG(vote_average) AS average_votes
    FROM movies
    GROUP BY original_language
    ORDER BY average_votes DESC
""")
vote_average_per_language.show()
```

7.7 Determine the top 10 Original Titles with Largest Tagline.

A unique perspective on marketing ingenuity was offered by grouping films according to the tagline column's length. Effective branding is exemplified by movies with catchy taglines like Beyond Fear, Destiny Awaits for Dune.This chart presents information in an organized way, such as Original Titles with Largest Tagline. Comparing values is made simple by horizontal bars, which highlight differences across groups.

```
largest_tagline = spark.sql("""
   SELECT original_title, tagline, LENGTH(tagline) AS tagline_length
   FROM movies
   WHERE tagline IS NOT NULL
```

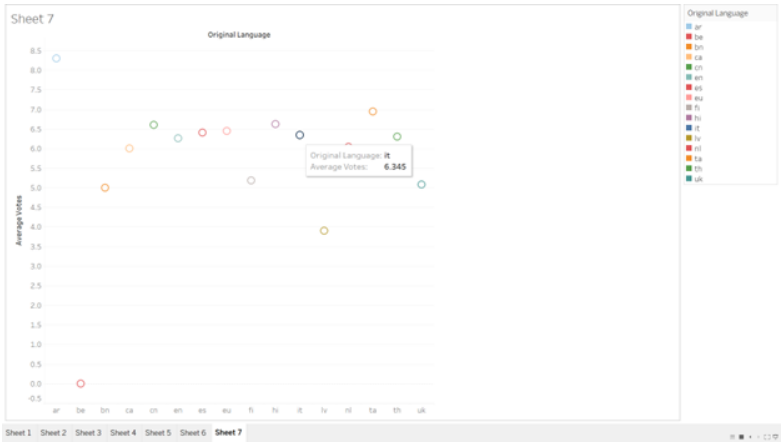

Figure 7: Displaying the Vote Average Per Language.

```
    ORDER BY tagline_length DESC
    LIMIT 10
""")
largest_tagline.show()
```

7.8 Determine the Highest Revenue Film in Each Year.

Trends across time were revealed by classifying films according to the year of release and determining which film made the most money in each group. As an example of how audience tastes have changed over time, Titanic and The Dark Knight were the highest grossing films in their respective years.The highest-grossing films are graphically represented via a tree map. Every rectangle represents a movie, and the size of each one reflects its box office receipts. Higher-earning films are indicated by larger rectangles.

```
highest_revenue_per_year = spark.sql("""
 SELECT m.release_year, m.original_title, m.revenue
 FROM (
     SELECT release_year, MAX(revenue) AS max_revenue
     FROM movies_with_year
     WHERE revenue IS NOT NULL
     GROUP BY release_year
 ) AS max_revenue_per_year
```

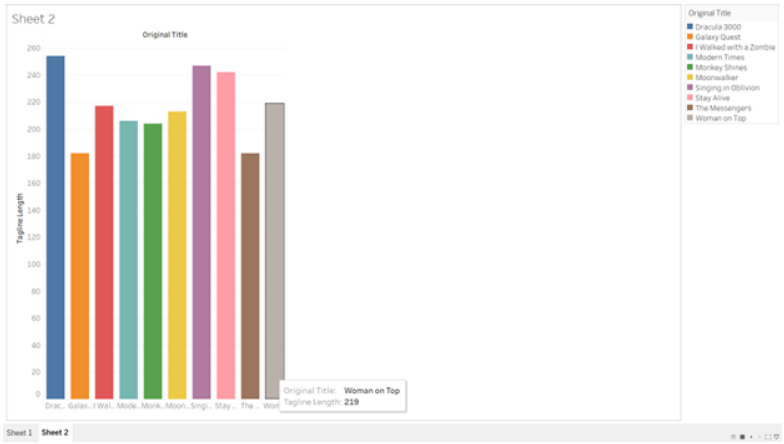

Figure 8: The top 10 Original Titles with Largest Tagline.

```
JOIN movies_with_year m
ON m.release_year = max_revenue_per_year.release_year
AND m.revenue = max_revenue_per_year.max_revenue
""")
highest_revenue_per_year.show()
```

7.9 Retrieve all movies released in a given year.

Movies released in particular years could be extracted by filtering the dataset using the release-date column. For monitoring the patterns, topics, and genres that are prevalent throughout a given time period, this study is essential.This graphic shows how many releases were made in a specific year.Each year's worth of films or categories are compared using horizontal bars. The length of each bar corresponds to its associated value.

```
# Replace '2021' with the desired year
specific_year = 2021
movies_in_year = spark.sql(f"""
    SELECT original_title, release_date, genre,
    revenue, runtime, vote_average
    FROM movies_with_year
    WHERE release_year = {specific_year}
""")
movies_in_year.show()
```

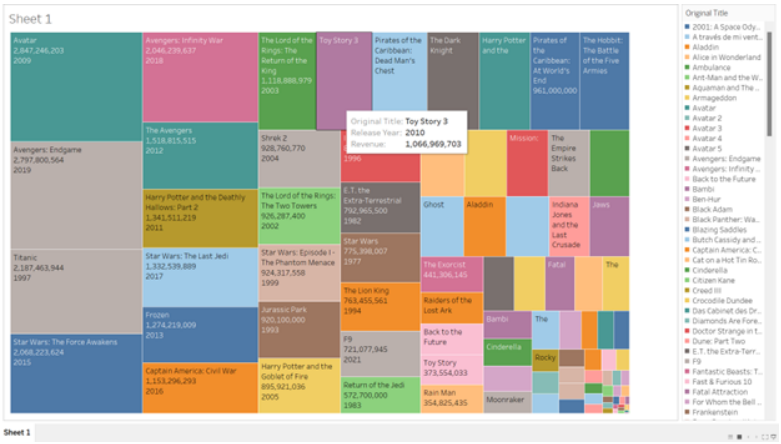

Figure 9: The Highest Revenue Film in Each Year.

8 Conclusion

In summary, the movie dataset research offered insightful information about a number of topics, such as the highest-grossing films, preferred titles, language distribution, and runtime extremes. We processed a lot of data quickly and effectively by using PySpark, which ensured high-quality results with low latency and resource usage. Critical patterns were shown by metrics including revenue trends, average ratings, and vote counts, which helped in data-driven decision-making. This thorough investigation demonstrates the effectiveness of big data analytics in revealing significant patterns and improving comprehension across a variety of fields.

9 DataSet and GitHub Link

1. Dataset references from Kaggle: Dataset
2. Github Link: Github Link

References

[1] Akbar, A., Agarwal, P., Obaid, A.J.: Recommendation engines-neural embedding to graph-based: Techniques and evaluations. International Journal of Nonlinear Analysis and Applications **13**(1), 2411–2423 (2022)

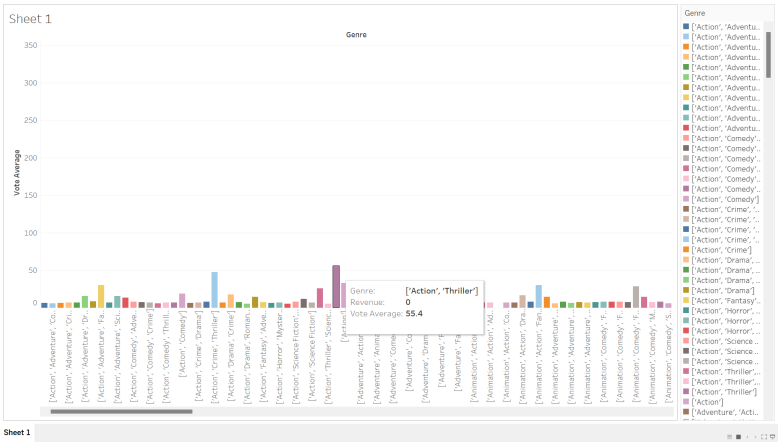

Figure 10: Movies released in a given year.

[2] Awan, M.J., Khan, R.A., Nobanee, H., Yasin, A., Anwar, S.M., Naseem, U., Singh, V.P.: A recommendation engine for predicting movie ratings using a big data approach. Electronics **10**(10), 1215 (2021)

[3] Kılınç, D.: A spark-based big data analysis framework for real-time sentiment prediction on streaming data. Software: Practice and Experience **49**(9), 1352–1364 (2019)

[4] Ma, Y., Wan, Y.: [retracted] data analysis method of intelligent analysis platform for big data of film and television. Complexity **2021**(1), 9947832 (2021)

[5] Tandon, A., Ryza, S., Owen, S., Laserson, U., Wills, J.: Advanced analytics with PySpark. " O'Reilly Media, Inc." (2022)
(2) (3) (1) (4) (5)