**IMDB 2024 Data Scraping and Visualizations**

**1. Introduction**

The IMDB 2024 Data Scraping and Visualizations project aims to extract and analyze movie data from IMDb for the year 2024. This project involves leveraging **Selenium** for web scraping, **Pandas** for data processing, **SQL** for structured storage, and **Streamlit** for interactive visualization. The primary goal is to identify trends in movie ratings, voting counts, and genre popularity, providing users with insights into the current movie landscape.

This project is particularly useful for entertainment analysts, movie enthusiasts, and data scientists who wish to explore IMDb movie statistics interactively.

**2. Problem Statement**

With a continuously expanding movie industry, there is a need to analyze and categorize new releases efficiently. IMDb provides a vast repository of data, but manually extracting insights is time-consuming. This project automates the data extraction process, stores the information in a structured format, and presents it via an interactive dashboard.

**Objectives:**

* **Scrape IMDb movie data for 2024** including movie names, genres, ratings, voting counts, and duration.
* **Store and structure the data** in a database for easy retrieval and analysis.
* **Visualize the data** dynamically using Streamlit to enable exploration and insight generation.
* **Implement interactive filters** to allow users to refine data by various criteria such as genre, ratings, and duration.

**3. Technology Stack**

**Languages and Tools Used:**

* **Programming Language:** Python
* **Web Scraping:** Selenium
* **Data Processing:** Pandas
* **Database:** MySQL
* **Visualization:** Streamlit, Matplotlib, Seaborn

**Why These Technologies?**

* **Selenium** allows interaction with dynamic web pages, handling JavaScript-heavy content on IMDb.
* **Panda** provide efficient data manipulation and cleaning.
* **SQL** offers structured storage with optimized querying capabilities.
* **Streamlit** creates a seamless user interface for data visualization and filtering.

**4. Approach**

**Step 1: Web Scraping with Selenium**

* Used **Selenium WebDriver** to navigate the IMDb 2024 movie list.
* Extracted key details such as **Movie Name, Genre, Ratings, Voting Counts, and Duration**.
* Implemented **explicit waits** to handle dynamic page loading.
* **Pagination handling** ensured that data across multiple pages was captured.
* Stored extracted data **temporarily in CSV files** before merging them into a structured dataset.

**Step 2: Data Storage in SQL**

* Created a **normalized database schema** for efficient querying.
* Used **SQLAlchemy ORM** to manage database operations.
* Imported cleaned CSV data into **MySQL**, ensuring indexing for faster access.
* Used **SQL queries to validate data integrity** and eliminate duplicates.

**Step 3: Data Analysis & Visualization with Streamlit**

* Designed a **Streamlit dashboard** for real-time data exploration.
* Created **multiple interactive visualizations**, including:
  + **Top 10 Movies by Ratings and Votes**
  + **Genre Distribution**
  + **Average Movie Duration by Genre**
  + **Rating Distribution**
  + **Voting Trends**
  + **Genre vs. Ratings Heatmap**
  + **Shortest and Longest Movies**
* Integrated **interactive filters** to allow users to refine data dynamically based on:
  + **Duration (Short, Medium, Long)**
  + **Rating Threshold (Above a certain IMDb rating)**
  + **Voting Count (Popularity based filtering)**
  + **Genre Selection**

**5. Implementation**

**Web Scraping (Selenium)**

* Implemented **automated browser navigation** using Selenium.
* Used **XPath** to locate movie details.
* Managed **dynamic content loading** using implicit and explicit waits.
* Handled **View more button option using** **ActionChains** **and scrolling** to extract a complete dataset.
* Saved the extracted data as **individual CSV files per genre**.

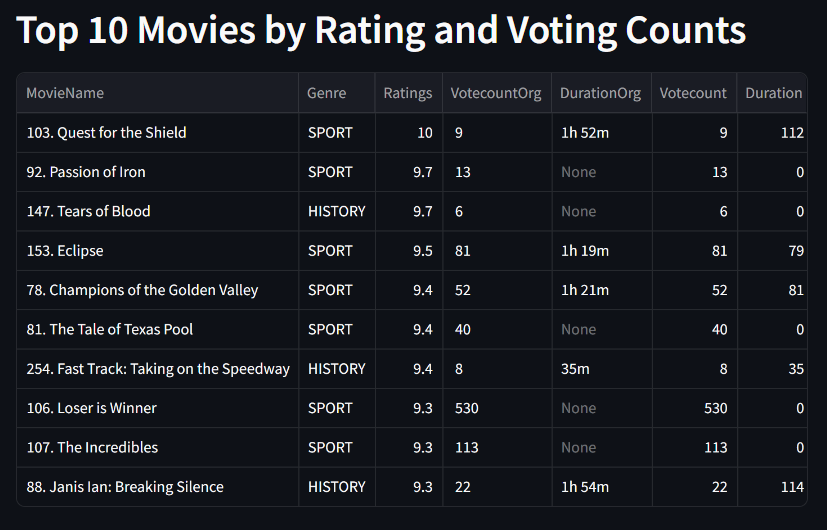
**Database Integration**

* Created **table for merged dataset** to optimize query performance.
* Used **dataframe to sql insertion using SQLAlchemy** for efficiency.
* Validated data using **SQL constraints** (e.g., primary keys).

**Visualization with Streamlit**

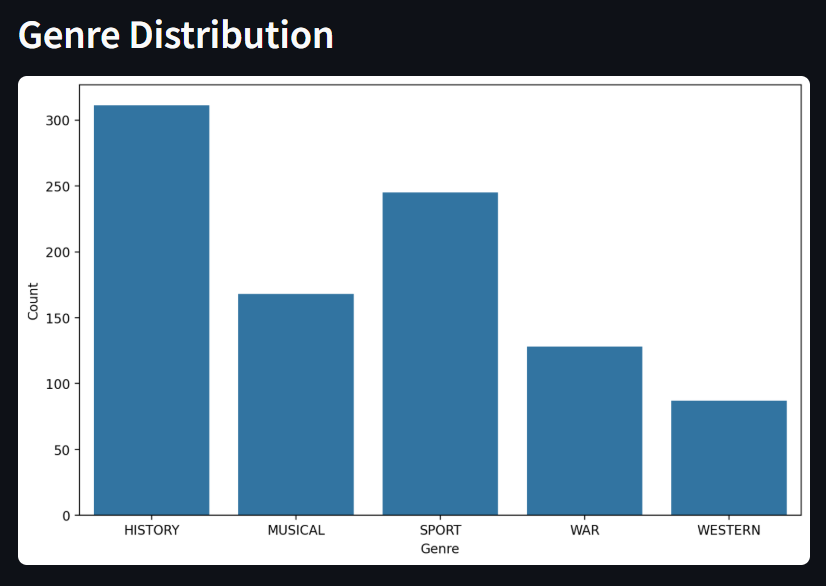
* Developed an **interactive UI** using Streamlit components.
* Designed a **user-friendly filtering system** for easy navigation.
* Optimized performance using **Streamlit caching** for large datasets.
* Integrated **downloadable reports** so users can export data in CSV format.

**6. Results & Insights**



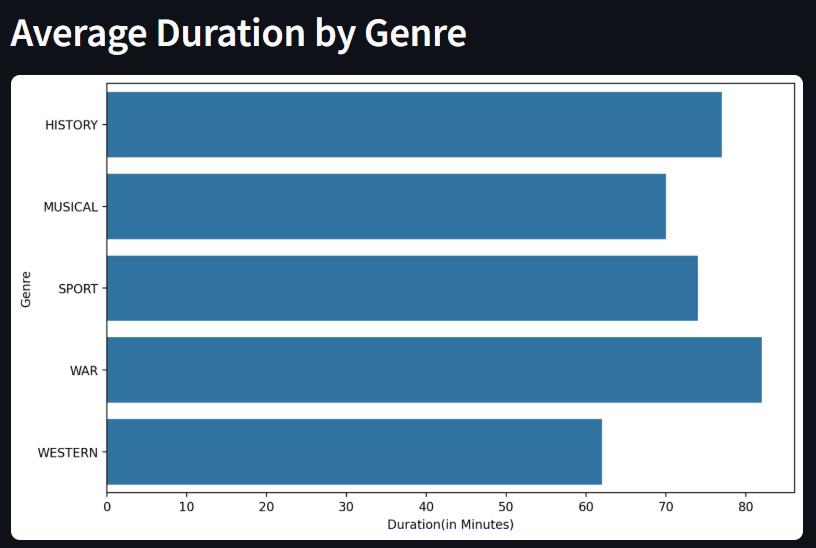
**Insights:**

Sport and History genre is most likely interested by audience based on Ratings given by them. Mostly highest ratings having less vote count and highest vote count has less rating.



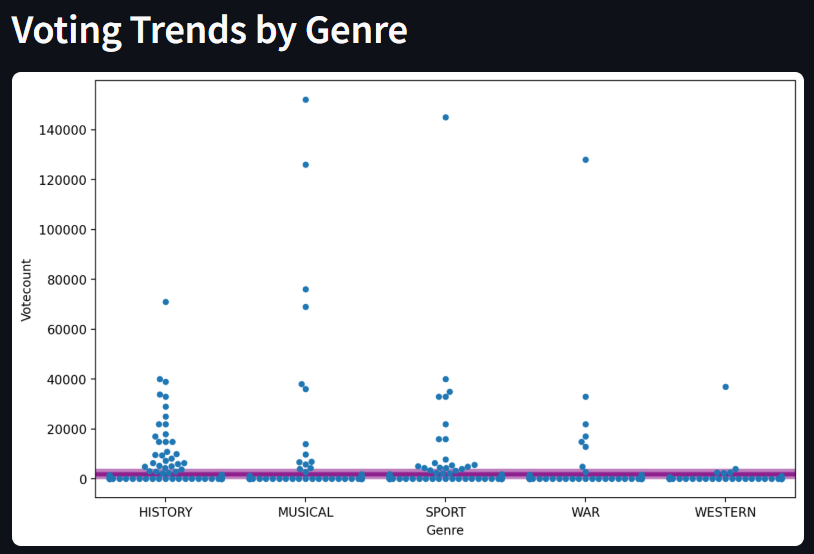
**Insights:**

In History genre more movies are released in the year of 2024 compared to other genre.



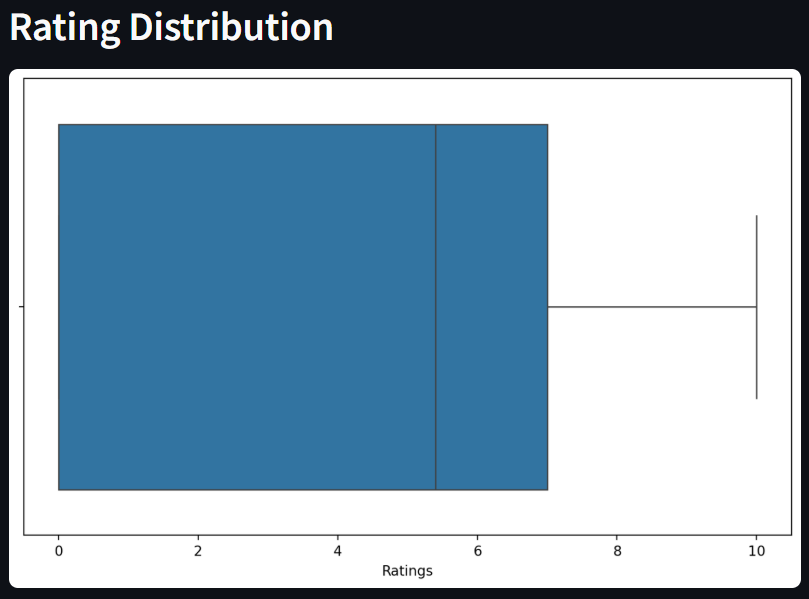
**Insights:**

In War genre and History genre movies are often have more duration to mention the fiction/real time in detail.



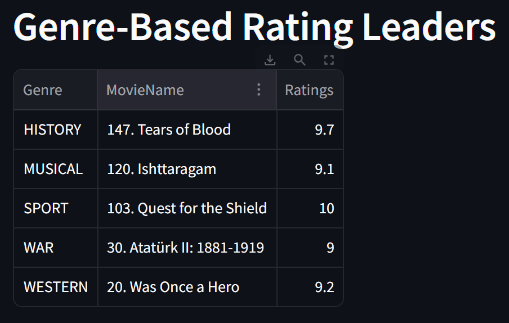
**Insights:**

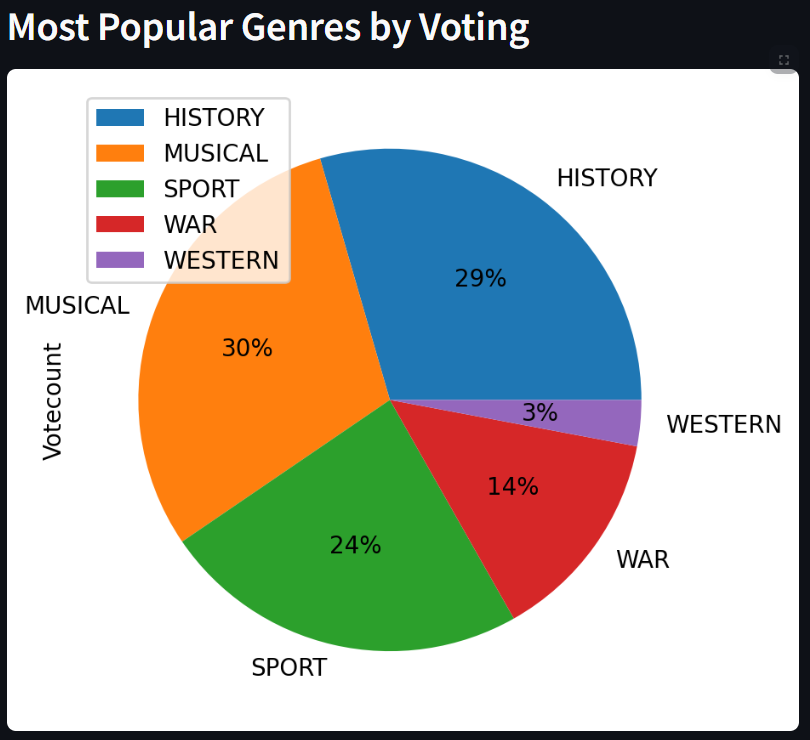
History movies consistently receive high vote counts, indicating a strong preference among audiences for voting. And the purple line denote the average voting count across all genres, which indicates most of movies under each genre are having very less vote count.



**Insights:**

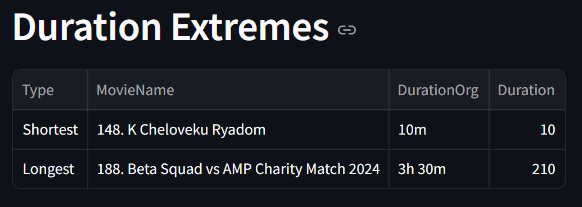
The overall distribution of ratings is relatively balanced, with a slight skew towards higher ratings. This suggests that audiences generally rate movies favorably but are critical of movies that don’t meet their expectations.

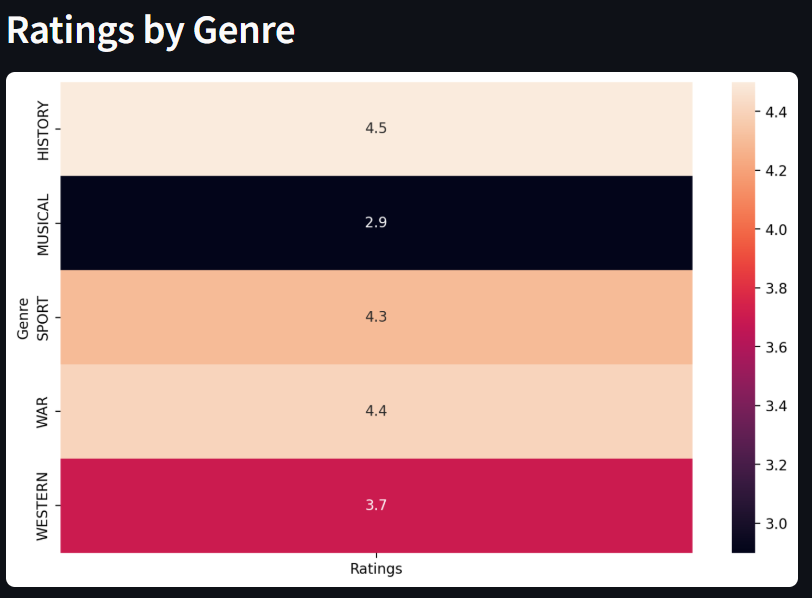




**Insights:**

**History, Musical** **and Sport** genres lead in popularity, receiving the highest votes among all genres.



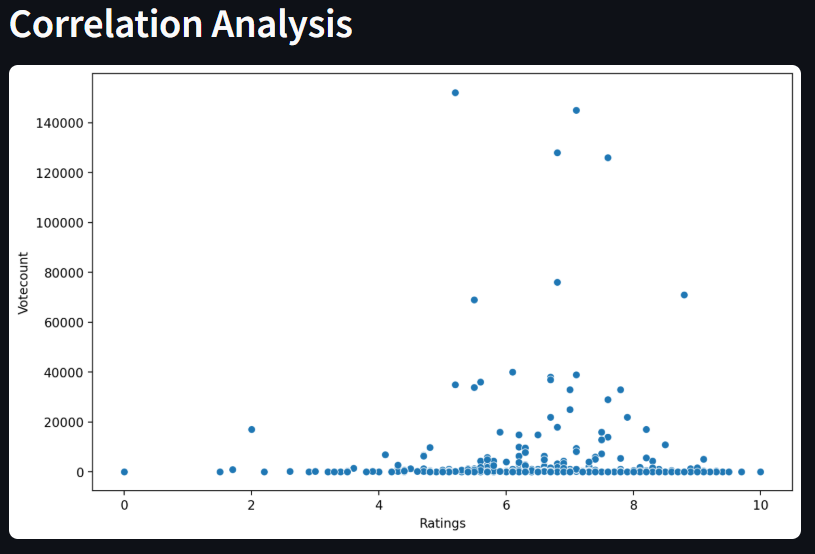


**Insights:**

**History, War** **and Sport** genres lead in average ratings, showcasing their popularity and strong audience preference.

**Western** genres have moderate ratings, indicating consistent and stable reception.

**Musical** genres have lower average ratings, suggesting that these genres might be more polarizing among audiences.



**Insights:**

There is a strong positive correlation between ratings and vote counts, indicating that Average-rated movies between 5 to 8 Ratings generally receive more votes.

**7. Challenges & Solutions**

|  |  |
| --- | --- |
| **Challenge** | **Solution** |
| Dynamic web content loading | Used explicit waits and dynamic element handling |
| Handling view more on IMDb | Automated scrolling and page transitions — |
| UI in Streamlit | Implemented UI by learning Streamlit in short period of time. |

**8. Conclusion**

This project successfully automates IMDb data extraction, organizes it in a structured format, and presents insights through an interactive Streamlit dashboard. The approach enables data analysts and movie enthusiasts to explore IMDb trends efficiently.

**9. References**

* IMDb official website: [https://www.imdb.com](https://www.imdb.com/)
* Streamlit Documentation: <https://docs.streamlit.io/>
* Selenium Guide: <https://www.selenium.dev/documentation/>