Analyzing Movie Trends: A Project on Genres and Ratings

1. **The data and its source :**

   The data used in this analysis comes from the MovieLens Dataset, a well-known and publicly available resource curated by the GroupLens research lab. It contains detailed information about movies, their genres, and user ratings. The dataset is widely used in research and educational projects for exploring data analysis, recommendation systems, and trends in viewer behavior.

   Link for the website : Datasets

   The two datasets used for this project are :

   movies.csv
   ratings.csv

2. **A description of data exploration and data cleaning steps :**

1. Data Inspection:

- The movies dataset has 27,278 entries with three columns: movieId, title, and genres. All columns have non-null values, indicating no missing data.
- The ratings dataset has 20,000,263 entries with four columns: userId, movieId, rating, and timestamp. Similarly, no missing data is present in this dataset.

Memory Usage:

- movies: 639.5 KB
- ratings: 610.4 MB

Action: No immediate action was required to handle missing values as all columns are complete.

2. Initial Dataset Exploration:

- Using head():
- The movies dataset was previewed to understand the structure of movie titles and their associated genres.
- The ratings dataset was previewed to examine user ratings and timestamps.

Observation:

- The genres column in the movies dataset contains pipe-separated (|) genres.
- The timestamp column in the ratings dataset stores time in Unix format, which might need transformation if used for time-based analysis.

3. Merging Datasets:

- The movies and ratings datasets were merged on the movieId column using an inner join.
- The resulting dataset contains a combination of movie information and corresponding user ratings.
- Aggregated metrics were computed:
- avg_rating: The mean rating for each movie.
- total_ratings: The total number of ratings received by each movie.

Action: This merge ensures all further analysis focuses on movies that have both details and ratings.

4. Cleaning Steps:

- Removing Invalid Genres:

- Rows where the genres column contains the value (no genres listed) were dropped. These rows are not relevant for genre-based analysis.
- After this step, the dataset was reduced to 26,502 rows.

- Removing Duplicates:

- Duplicate rows were removed to ensure each movie is represented uniquely in the dataset.

5. Null Value Check:

- After cleaning, a null value check confirmed that the dataset is complete with no missing values.

3. **Three comparison questions with the unit of analysis, the comparison values and how they are computed** :

A.) What is the distribution of movies across genres?

- Unit of Analysis: Genres

  The analysis focuses on individual genres like "Comedy," "Drama," "Horror," etc., to determine their representation in the dataset.

- Comparison Values: Number of Movies per Genre
  For each genre, the count of movies is computed as the primary value for comparison.

- How it is Computed:

Step 1: Split the genres column into individual genres using the split('|') method. This allows a single movie associated with multiple genres (e.g., "Comedy|Drama") to be included in counts for both "Comedy" and "Drama."

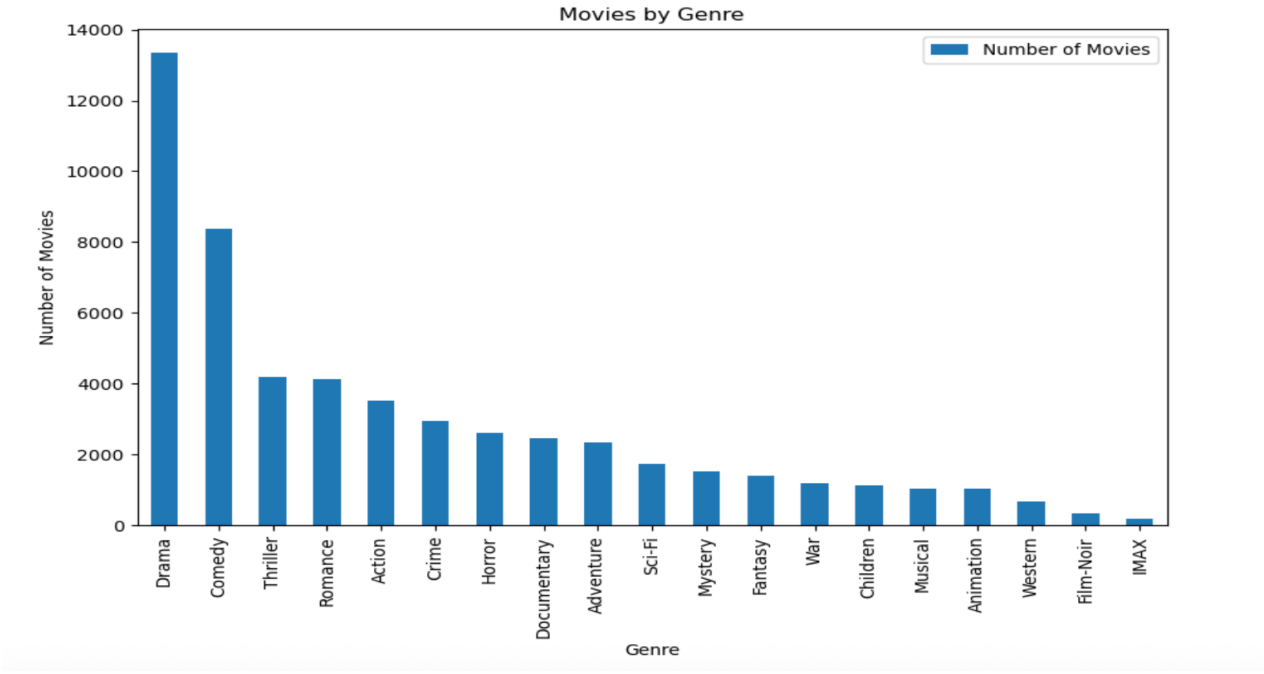
Step 2: Explode the dataset so that each genre occupies its own row. For instance, if a movie has three genres, it will appear in three rows—one for each genre.

Step 3: Perform a value_counts() operation on the genres column to calculate the frequency of each genre.

Step 4: Exclude genres labeled as (no genres listed) since they don't provide meaningful information.

- Why It's Useful:

This comparison helps understand the relative popularity or representation of various genres in the dataset. For instance, genres like "Drama" might dominate, whereas niche genres like "Film-Noir" could have fewer movies.



B.) How do single-genre and multi-genre movies compare by decade?

- Unit of Analysis: Decades
  Movies are grouped by their release decade (e.g., 1990s, 2000s, etc.) to analyze trends over time.

- Comparison Values:

Number of Single-Genre Movies: The count of movies associated with only one genre within each decade.

Number of Multi-Genre Movies: The count of movies associated with more than one genre within each decade.

- How it is Computed:

Step 1: Extract the release year from the title column using a regular expression. The year is then converted into a decade column by rounding down to the nearest multiple of 10 (e.g., 1995 becomes 1990).

Step 2: Use the genre_count column created earlier to classify

movies: Single-Genre Movies: Filter movies where

genre_count == 1.

Multi-Genre Movies: Filter movies where genre_count > 1.

Step 3: Group the dataset by decade and count the number of single-genre and multi-genre movies for each decade.

- Why It's Useful:

This comparison provides insights into how movie complexity (in terms of genres) has evolved over time. For instance, recent decades might have more multi-genre movies due to changing audience preferences or industry trends.

C.) How do total movies and ratings compare across decades?

- Unit of Analysis: Decades.
- Comparison Values:

Total Movies (Total Movies): The count of movies released in each decade.

Total Ratings (total_ratings): The count of all ratings for movies in each decade.

- How it is Computed:

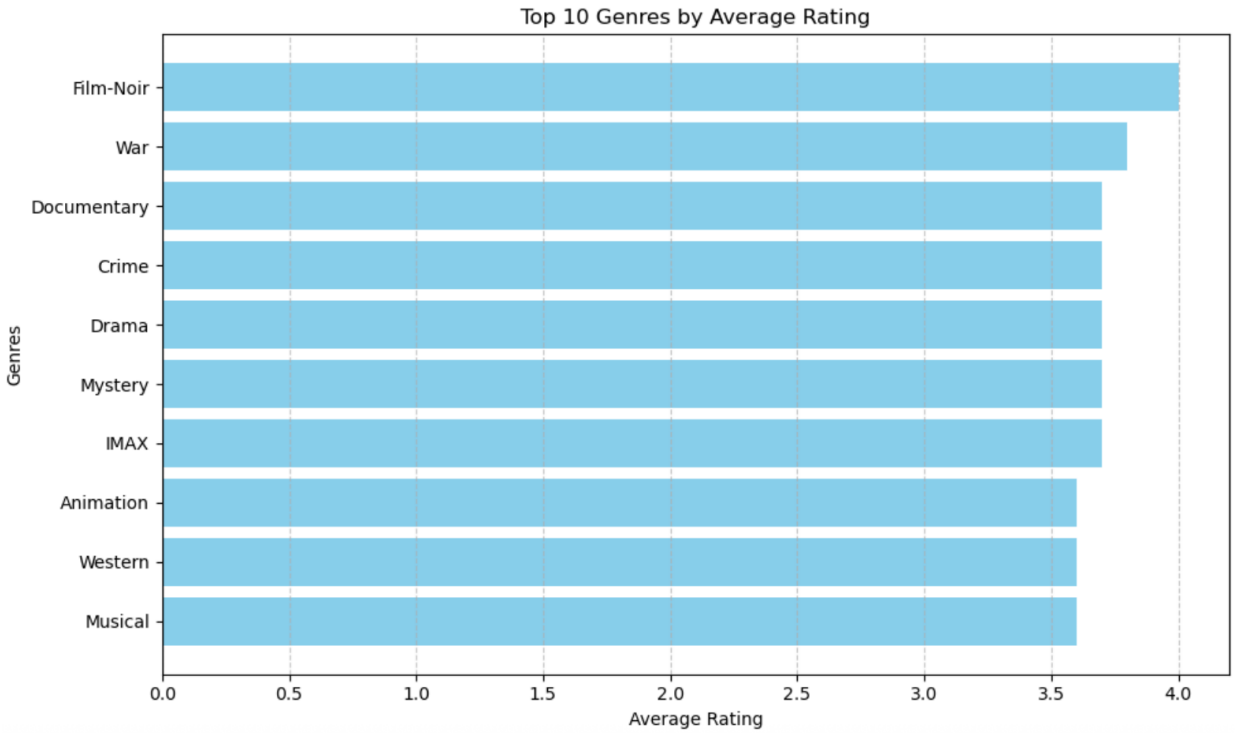
Extracted the decade from the year column in the movies dataset.

Merged the ratings dataset with movies to associate ratings with decades.

Grouped by decade and computed:

Total Movies as the count of movies.

total_ratings as the count of the rating column.



4. **A description of the program :**

The program analyzes two datasets, movies and ratings, to generate insights about movie genres, decades, and user engagement. It begins by cleaning and formatting the data, including splitting genres into individual categories, extracting release years, and categorizing movies by decade. The program then performs three levels of analysis: genre-level (e.g., total movies and ratings by genre), decade-level (e.g., total movies and ratings per decade), and user-level (e.g., average ratings and total ratings per user). It integrates the two datasets using movieId as the common key for combined insights. The results are visualized using bar charts and saved as .csv files for further use. This approach ensures meaningful comparisons and clear summaries for analysis

5. Description of the Output Files

Output Files:

1. genre_summary.csv:
- Contains the total number of movies for each genre.
- Columns: Genre, Number of Movies.

2. multiple_genres.csv:
- Documents the count of movies associated with multiple genres.
- Columns: Description, Value.

3. movies_with_max_genres.csv:
- Lists movies with the maximum number of genres.
- Columns: movieId, title, genres, genre_count.

4. movies_by_decade.csv:
- Contains the number of movies released in each decade.
- Columns: Decade, Number of Movies.

5. genre_type_ratio.csv:
- Compares single-genre and multi-genre movies by decade.
- Columns: decade, Single-Genre Movies, Multi-Genre Movies.

6. genre_comparison.csv:

- Compares total movies and user ratings by genre.
- Columns: Genre, Total Movies, Avg Rating, Total Ratings.

7. decade_comparison.csv:
- Compares total movies and user ratings by decade.
- Columns: Decade, Total Movies, Avg Rating, Total Ratings.