## Report on Data Preparation and Merging for Movie Recommendation Dataset

### Overview

The objective of this data preparation process was to create a comprehensive and enriched dataset for movie recommendations. This dataset combines information from multiple sources (movie.csv, rating.csv, and link.csv) to include details on movie titles, genres, user ratings, release years, and unique identifiers for external data enrichment. We employed several steps to clean, merge, and fill in missing values, resulting in a complete dataset with minimal gaps and standardized movie information.

### Step 1: Initial Data Loading and Title Cleanup

1. **Loading Movies Dataset:** The primary dataset, movie.csv, was loaded to extract information on each movie's title, genre, and release year. However, in many cases, the release year was embedded within the movie title (e.g., "Toy Story (1995)"), which required additional cleaning.
2. **Title and Year Extraction:**
   * **Year Extraction:** The release year was extracted from each movie title and placed in a new column, year.
   * **Title Cleanup:** The original title was modified by removing the year portion, leaving only the movie name (e.g., "Toy Story" instead of "Toy Story (1995)").
   * **Reformatting Titles with Misplaced Articles:** Titles were corrected for misplaced articles such as "The," "A," and "An" that were found at the end (e.g., "Lion King, The" was corrected to "The Lion King").

This initial cleaning ensured that titles were consistent, facilitating easier matching and merging with other data sources.

### Step 2: Merging Ratings Data

1. **Loading Ratings Dataset:** We loaded the rating.csv file, which contains user ratings for each movie. This dataset includes three main columns: movieId, userId, and rating.
2. **Merging Movies with Ratings:** An outer join was performed between data\_movie and data\_rating on the movieId column.
   * **Outer Join:** This type of join was chosen to retain all movies in the dataset, even if they lacked ratings. This ensured that movies without user ratings were still included for potential recommendations based on other attributes like genre.
3. **Outcome:** After merging, we had a dataset containing:
   * Each movie's title, genre, and release year.
   * User ratings for movies where available.
   * Retained all movies, even those without ratings, to ensure dataset completeness.

### Step 3: Merging Links Data for External Identifiers

1. **Loading Links Dataset:** The link.csv file was loaded to include unique external identifiers (TMDB IDs) for each movie, essential for data enrichment from external sources like The Movie Database (TMDB).
2. **Adding tmdbId to the Main Dataset:** We merged the link.csv data with our main dataset on the movieId column to add the tmdbId column.
   * **Left Join:** A left join was used here to ensure all movies in our existing dataset were retained, even if some did not have corresponding TMDB IDs in the links dataset.
3. **Outcome:** After this merge, each movie entry in the dataset had a tmdbId if available. This enabled us to retrieve additional details, such as missing release years, from external sources like TMDB.

### Step 4: Enriching Data with TMDB API for Missing Release Years

1. **Identifying Missing Release Years:** After the initial merges, some movies still had missing year values. To address this, we used the tmdbId to fetch release years from TMDB for movies with missing year values.
2. **Fetching Release Years Using TMDB API:**
   * **API Call:** A function was created to query the TMDB API using the tmdbId and retrieve the release\_date. Only the year portion was extracted to populate the year column.
   * **Updating the Dataset:** For each movie with a missing year, the fetched year from TMDB was inserted in a temporary tmdb\_year column.
3. **Updating Year Column:** Finally, the main year column was updated with values from tmdb\_year where it was previously missing. This ensured that as many movies as possible had complete release year data.

### Step 5: Manual Filling of Remaining Missing Years

1. **Manual Year Mapping:** After using TMDB to fill in missing years, a few movies still lacked release years. For these cases, we manually populated the year column using a predefined mapping of movieId to year.
2. **Applying Manual Mapping:** The manual mapping was applied to fill missing values directly in the year column. This served as a final fallback step, ensuring no movie entry was left with a missing release year.

### Step 6: Final Checks and Export

1. **Summary of Missing Values:** After all merges and data enrichment steps, a final check was performed to confirm that missing values were minimized, particularly in key columns such as year, title, and tmdbId.
2. **Saving the Final Dataset:** The fully prepared dataset, now containing:
   * Standardized titles
   * Genres
   * User ratings
   * Release years
   * TMDB identifiers

### was saved as new\_dataset2.csv. This dataset serves as a robust foundation for building a recommendation model, providing comprehensive information for each movie while ensuring data consistency and completeness.

Conclusion

This process combined data from multiple sources, systematically cleaned and enriched the information, and handled missing values to create a comprehensive dataset ready for movie recommendation modeling. By merging the data from movie.csv, rating.csv, and link.csv and using external data from TMDB where needed, we achieved a final dataset that balances detail with completeness, making it suitable for further analysis and recommendation system development.