Movie Data Analysis

Project: Milestone 1

Samuel Aboye

College of Science and Technology

Bellevue University

Masters in data science

Course Number: DSC:540

Project Subject Area:

The project aims to analyze and uncover factors contributing to movies' commercial success and audience reception by integrating and examining data from The Movie Database (TMDB), The Open Movie Database (OMDb) API, and IMDb. This comprehensive approach will leverage detailed movie credits, financial metrics, audience ratings, and other relevant data to provide insights into industry trends, patterns, and predictors of movie performance.

Data Sources:

Flat File:

- **Description of data source:** The provided flat files, **tmdb_5000_credits.csv** and **tmdb_5000_movies.csv**, contain detailed information on movie credits and various movie attributes from The Movie Database (TMDB). These files include data on cast, crew, budgets, revenues, genres, and more for thousands of movies.
- **Link or Flat File uploaded:**

  https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata

```
[9]: df = pd.read_csv('tmdb_5000_movies.csv')

[10]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   budget                4803 non-null   int64
 1   genres                4803 non-null   object
 2   homepage              1712 non-null   object
 3   id                    4803 non-null   int64
 4   keywords              4803 non-null   object
 5   original_language     4803 non-null   object
 6   original_title        4803 non-null   object
 7   overview              4800 non-null   object
 8   popularity            4803 non-null   float64
 9   production_companies  4803 non-null   object
 10  production_countries  4803 non-null   object
 11  release_date          4802 non-null   object
 12  revenue               4803 non-null   int64
 13  runtime               4801 non-null   float64
 14  spoken_languages      4803 non-null   object
 15  status                4803 non-null   object
 16  tagline               3959 non-null   object
 17  title                 4803 non-null   object
 18  vote_average          4803 non-null   float64
 19  vote_count            4803 non-null   int64
dtypes: float64(3), int64(4), object(13)
memory usage: 750.6+ KB
```

```
[11]: df.head(10)
```

API:

- **Description of data source:** The OMDb API (The Open Movie Database) is a **RESTful** web service for obtaining movie information. It offers detailed data, including titles, year, ratings, plot descriptions, and poster images for movies and TV series. In order to expand the dataset, I plan to utilize the movie titles extracted from the 'tmdb_5000_credits.csv' file as search queries for the OMDb API. Subsequently, the movie data obtained from the API will be saved in a new CSV file, systematically amalgamating an extensive compilation of movie metadata.**Link:** OMDb API

```
import pandas as pd
import requests

# Load the dataset of movie titles
df_titles = pd.read_csv('tmdb_5000_movies.csv')

#OMDb API key
api_key = 'f70fda88&t'

# Define the function to get movie details from OMDb API
def get_movie_details(api_key, title):
    """
    Fetches movie data from OMDb API based on the title.
    """
    url = f"https://www.omdbapi.com/?t={title}&apikey={api_key}"
    response = requests.get(url)
    if response.ok:
        return response.json()
    else:
        return None

# Iterate over movie titles in the DataFrame and call the OMDb API
# We'll store each movie's data in this list
movies_data = []

for title in df_titles['title']:
    movie_details = get_movie_details(api_key, title)
    if movie_details:
        movies_data.append(movie_details)
    else:
        print(f"Data for {title} could not be fetched.")

# Create a DataFrame from the list of movie data
df_movies = pd.DataFrame(movies_data)

# Save the movies data DataFrame to a new CSV file
df_movies.to_csv('open_movie_data.csv', index=False)

print(f"Data for {len(df_movies)} movies fetched and saved to open_movie_data.csv.")
```

```
Data for Nanny McPhee and the Big Bang could not be fetched.
Data for Harold & Kumar Escape from Guantanamo Bay could not be fetched.
Data for Of Gods and Men could not be fetched.
Data for What the #$*! Do We (K)now!? could not be fetched.
Data for The Witch could not be fetched.
Data for #Horror could not be fetched.
Data for 4797 movies fetched and saved to open_movie_data.csv.
```

```
df_omdb = pd.read_csv('open_movie_data.csv')
```

```
df_omdb.shape
```

```
(4797, 27)
```

Website:

- **Description of data source:** IMDb (Internet Movie Database) offers a comprehensive database and ratings for movies, TV shows, and celebrities. It's a crucial source for movie ratings, reviews, and detailed cast and crew information.
- **Link:** IMDb

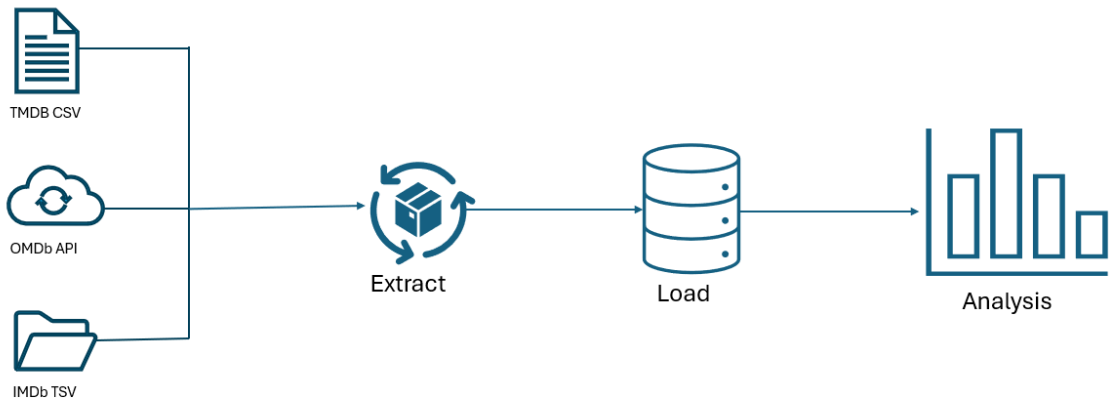

Figure 1. The project processes.

Relationships:

The data from each source are interconnected through movie titles and unique identifiers such as IMDb IDs. For instance, the tmdb_5000_movies.csv file includes an id field that corresponds to TMDB's unique identifier for each movie, which can potentially be matched with the IMDb ID available through the OMDb API and IMDb's own dataset. Furthermore, movie titles and release years are common keys that can link data across these sources, allowing for an enriched dataset combining detailed credits, financial data, ratings, and descriptive metadata from multiple authoritative sources.

Project Approach/Plan:

The project aims to merge these diverse datasets to comprehensively analyze movies, focusing on factors affecting their success, audience reception, and financial performance. The approach will involve:

1. **Data Cleaning and Preprocessing:** Standardizing movie titles, handling missing values, and resolving discrepancies in movie identifiers across datasets.
2. **Data Integration:** Merging datasets on common keys (e.g., movie titles, IMDb IDs) to create a unified dataset.
3. **Analysis:** Conducting exploratory data analysis (EDA) to identify trends, patterns, and outliers in the movie industry.
4. **Modeling (if applicable):** Applying statistical or machine learning models to predict movie success metrics based on various factors.

Concerns/Challenges:

- **Data Quality and Consistency:** Ensuring accuracy in matching movies across different sources despite potential discrepancies in titles, release dates, or missing data.
- **API Rate Limits and Accessibility:** Navigating API usage limits and ensuring sustainable access to up-to-date data.
- **Handling Large Datasets:** Efficiently process and merge large datasets without compromising performance.

Ethical Implications:

- **Privacy Concerns:** Ensuring that any data related to individuals (e.g., cast, crew) is handled responsibly, respecting privacy and avoiding misuse.
- **Bias and Fair Representation:** Acknowledging that the databases may inherently reflect industry biases, such as underrepresentation of certain groups in leading roles or within certain genres.

- **Impact on Perception:** The way data is presented and analyzed could influence public perception of movies, actors, or the industry, necessitating a balanced and fair approach to analysis and interpretation.

This project aims to provide a comprehensive overview of the movie industry by combining data from different reliable sources. The main objective is to discover insights into the factors that contribute to the success or failure of films. However, the project might face challenges, including issues related to data quality, ethical considerations, and dealing with a large amount of data. Therefore, I will approach the data with respect and sensitivity toward its origins and implications, aiming for a profound and respectful analysis.

References:

- TMDB Movie Metadata. (n.d.). Retrieved from https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata

- OMDb. (n.d.). OMDb API - The Open Movie Database. Retrieved from https://www.omdbapi.com/

- IMDb Developer. (n.d.). Introducing the New IMDb API. Retrieved from https://developer.imdb.com/