CHAPTER I - INTRODUCTION

1.1 Scope of analysis:

The IMDB Top 250 movies dataset provides valuable insights into the highest-rated films based on audience reviews and ratings. This dataset helps analyze the key attributes that contribute to a movie's high ranking, including genre, director influence, cast impact, and audience reception. Understanding these factors enables filmmakers, critics, and researchers to identify trends and patterns that define cinematic success.

This study aims to explore how various aspects such as movie ratings, release years, genres, and box office performance affect a film's standing on IMDB. By analyzing data collected through web scraping, a structured database is built to efficiently store and manage movie-related information. This database enables seamless data retrieval and comprehensive analysis of trends across different decades and genres.

By applying exploratory data analysis (EDA) techniques, we can visualize and interpret patterns within the Top 250 movies dataset. This includes identifying the most dominant genres, the distribution of ratings over the years, and how certain directors and actors consistently produce highly-rated films. Additionally, by structuring the dataset in a relational database, we can enhance data accessibility and scalability for future studies.

This research bridges the gap between data analytics and film studies by leveraging data science methodologies. The insights derived can be beneficial for film production houses, streaming platforms, and movie enthusiasts seeking to understand what makes a movie critically and commercially successful. The findings contribute to data-driven decision-making in the entertainment industry, offering guidance for future productions and audience engagement strategies.

1.2 Objective

The primary objective of this study is to analyze the IMDB Top 250 movies dataset to uncover key factors that contribute to a movie's high rating and success. This includes understanding genre distribution, director and actor influence, and rating trends over time.

Specifically, the project aims to:

- Collect and preprocess IMDB Top 250 movie data through web scraping.
- Design and implement a structured database for efficient data storage and retrieval.
- Perform exploratory data analysis (EDA) to identify patterns and trends in movie ratings, genres, and release years.
- Provide insights that can guide future film production, marketing strategies, and audience engagement.

By achieving these objectives, this research enhances the understanding of what makes a movie critically acclaimed and popular among audiences, offering valuable insights for filmmakers, industry analysts, and movie enthusiasts.

1.3 Data Collection

The data for this project is collected using Selenium and Scrapy's Selector to scrape IMDB's official website. Web scraping enables the automated extraction of relevant movie details, including:

- Movie title
- IMDB rating
- Genre
- Release year
- Director and main cast
- Box office revenue (if available)

To achieve this, a headless Chrome browser is launched using Selenium, which loads the IMDB Top 250 movies page and ensures that all content, including lazy-loaded

elements, is fully rendered. The script then extracts relevant movie details using CSS selectors provided by Scrapy's Selector module.

To ensure data accuracy and reliability:

- The scraper waits for the webpage to fully load before extracting content.
- It dynamically scrolls the page to capture all movies.
- Extracted data is validated for missing values and cleaned before storage.

Once collected, the raw data undergoes preprocessing, including handling missing values, standardizing formats, and removing duplicates. This cleaned dataset is then stored in a relational database for further analysis and visualization. By structuring the data collection process effectively, we ensure a high-quality dataset that supports meaningful insights into the trends and patterns of the top 250 movies.

CHAPTER II - DATA UNDERSTANDING

2.1 Data understanding

The dataset consists of structured information on the Top 250 highest-rated movies on IMDB, stored in an SQLite database. It contains essential details about each movie, including its title, director, IMDB rating, release year, runtime, budget, box office revenue, genre, and award history. By analyzing these attributes, we can explore the factors contributing to a movie's critical and commercial success. The data allows us to identify trends in genre popularity, financial performance, and audience reception over different decades. Understanding these elements provides insights into how the film industry has evolved and what factors consistently influence a movie's ranking on IMDB. The structured format of the dataset facilitates comprehensive exploratory analysis, enabling comparisons across genres, directors, and production companies. Additionally, by incorporating financial metrics such as budgets and gross earnings. This dataset serves as a valuable resource for predicting movie performance, guiding filmmakers in making data-driven decisions, and offering researchers a deeper understanding of audience preferences in the film industry. The dataset consists of structured information on the Top 250 highest-rated movies on IMDB, stored in an SQLite database. This dataset contains various attributes that describe movie details, financial performance, and critical reception. These attributes help analyze the factors contributing to a movie's success and popularity over time.

Imported libraries:

1. Pandas – for reading the dataset files

2. NumPy – for numerical calculations

3. Matplotlib – for graphic visualization

4. Seaborn – for graphical visualization of the data

5.OS - for interacting with the operating system

6.Sqlite3 – used to integrate the SQLite database with Python

7.Re – provides regular expression support

8.Plotly – designed for creating interactive visualizations

9. Requests – a library for making HTTP requests

10. Scrapy – a free and open-source web-crawling framework written in Python.

11.Selenium - used to carry out automated test cases for browsers or web applications

Scraped Dataset:

Scrape dataset using Scrapy and selenium

```
df=pd.DataFrame({"title":titles,"directors":directors,"Writers":writers,"Imdb_Ratings":imdp_rating,"Release_year":release_years,"Runtime":runtimes,"Motion_picture_Rating":motion_pic_rating,"budgets":budge
df
```

Python

	title	directors	Writers	Imdb_Ratings	Release_year	Runtime	Motion_picture_Rating	budgets	gross_us_and_canada	gross_worldwide	opening_weekend_gross_in_uscanada	Languages	Origin	
0	The Shawshank Redemption	Frank Darabont	[Stephen King, Frank Darabont]	9.3	1994	2h 22m	A	$25,000,000 (estimated)	$28,767,189	$29,332,133	$727,327	[English]	[United States]	[Epic,
1	The Godfather	Francis Ford Coppola	[Mario Puzo, Francis Ford Coppola]	9.2	1972	2h 55m	A	$6,000,000 (estimated)	$136,381,073	$250,342,198	$302,393	[English, Italian, Latin]	[United States]	Ga T
2	The Dark Knight	Christopher Nolan	[Jonathan Nolan, Christopher Nolan, David S. G...	9.0	2008	2h 32m	UA	$185,000,000 (estimated)	$534,987,076	$1,009,057,329	$158,411,483	[English, Mandarin]	[United States, United Kingdom]	[Actic Sup C

Figure 2.1.1

Shape of the dataset

Check the shape of the dataset **data. Shape ()** command

```
movies_df.shape
```

```
(27003, 19)
```

Figure 2.1.2

This dataset contains 27003 records and 19 columns

2.2 Data Description

A variable consists of two parts – the label and the data type. Data types can be numeric (integers, real numbers) or strings. The data type can sometimes be tricky; for example, US postal codes are numeric but need to be treated as strings. Once the labels and data types are known, you can group attributes into two kinds for modeling: 5

Continuous Variables: These are numbers which can range from negative infinity to positive infinity. You would associate with the labels a sense of magnitude, maximum and minimum. You can sort on such variables and filter by ranges.

Categorical Variables: These variables can have a limited set of values, each of which indicate a sub-type. For example, Direction is a categorical variable because it can be either North, South, East, or West. You can filter on or group by a specific value or values of a categorical variable

Now, let's look into the variables of our dataset. Once you have identified the variables of interest, summary statistics help you understand the nature of each variable. Each attribute's summary statistics such as **count, standard deviation, mean, minimum, maximum and IQR values** are calculated using the describe() function. To dive into the dataset, Python Programming is used for further process.

Here **describe()** function is used in python to derive the overall summary of the dataset. But in the dataset most of the values are categories. Therefore all datatypes are included in the function as describe() function only takes numeric datatype as default.

```
movies_df.describe()
```

Python

	movie_id	Imdb_Ratings	release_year	budgets(in millions)	gross_us_and_canada(in millions)	gross_worldwide(in millions)	opening_weekend_gross_in_uscanada(in millions)	Oscar	wins	nominations
count	27003.000000	27003.000000	27003.000000	27003.000000	27003.000000	27003.000000	27003.000000	27003.000000	27003.000000	27003.000000
mean	110.210754	8.346443	2000.673407	71.324297	170.462430	452.581306	40.153835	2.781284	67.134615	89.549235
std	71.014113	0.260433	18.306547	79.127105	191.420754	574.450270	70.593678	2.392742	76.100442	89.272559
min	0.000000	8.000000	1921.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	53.000000	8.100000	1993.000000	18.000000	32.000000	47.000000	0.000000	1.000000	15.000000	23.000000
50%	110.000000	8.300000	2003.000000	40.000000	85.000000	230.000000	8.000000	2.000000	43.000000	61.000000
75%	173.000000	8.500000	2015.000000	95.000000	282.000000	714.000000	50.000000	4.000000	85.000000	124.000000
max	245.000000	9.300000	2024.000000	356.000000	858.000000	2799.000000	357.000000	11.000000	355.000000	369.000000

Figure 2.2.1

Summarizing each attribute

Let's get to know about each and every variable below using describe () function individually

Movie_id:

```
movies_df['movie_id'].describe()
```

```
count    27003.000000
mean       110.210754
std         71.014113
min          0.000000
25%         53.000000
50%        110.000000
75%        173.000000
max        245.000000
Name: movie_id, dtype: float64
```

figure 2.2.2

Title:

```
movies_df['title'].describe()
```

```
count            27003
unique             246
top       Blade Runner
freq               1008
Name: title, dtype: object
```

Figure 2.2.3

Directors:

```
movies_df['directors'].describe()
```

```
count                27003
unique                 152
top       Christopher Nolan
freq                  2178
Name: directors, dtype: object
```

Figure 2.2.4

Imdb_Ratings:

```
movies_df['Imdb_Ratings'].describe()
```

```
count    27003.000000
mean         8.346443
std          0.260433
min          8.000000
25%          8.100000
50%          8.300000
75%          8.500000
max          9.300000
Name: Imdb_Ratings, dtype: float64
```

Figure 2.2.5

Release_year:

```
movies_df['release_year'].describe()
```

```
count    27003.000000
mean      2000.673407
std         18.306547
min       1921.000000
25%       1993.000000
50%       2003.000000
75%       2015.000000
max       2024.000000
Name: release_year, dtype: float64
```

Figure 2.2.6

Runtime:

```
movies_df['Runtime'].describe()
```

```
count     27003
unique      100
top        2h 9m
freq        1503
Name: Runtime, dtype: object
```

Figure 2.2.7

Motion_picture_Rating:

```
movies_df['Motion_picture_Rating'].describe()
```

```
count     27003
unique       14
top          UA
freq       8851
Name: Motion_picture_Rating, dtype: object
```

Figure 2.2.8

Budgets(in millions):

```
movies_df['budgets(in millions)'].describe()
```

```
count    27003.000000
mean        71.324297
std         79.127105
min          0.000000
25%         18.000000
50%         40.000000
75%         95.000000
max        356.000000
Name: budgets(in millions), dtype: float64
```

Figure 2.2.9

Gross_us_and_canada(in millions):

```
movies_df['gross_us_and_canada(in millions)'].describe()
```

```
count    27003.000000
mean       170.462430
std        191.420754
min          0.000000
25%         32.000000
50%         85.000000
75%        282.000000
max        858.000000
Name: gross_us_and_canada(in millions), dtype: float64
```

Figure 2.2.10

Gross_worldwide(in millions):

```
movies_df['gross_worldwide(in millions)'].describe()
```

```
count    27003.000000
mean       452.581306
std        574.450270
min          0.000000
25%         47.000000
50%        230.000000
75%        714.000000
max       2799.000000
Name: gross_worldwide(in millions), dtype: float64
```

Figure 2.2.11

Opening_weekend_gross_in_uscanada(in millions):

```
movies_df['opening_weekend_gross_in_uscanada(in millions)'].describe()
```

```
count    27003.000000
mean        40.153835
std         70.593678
min          0.000000
25%          0.000000
50%          8.000000
75%         50.000000
max        357.000000
Name: opening_weekend_gross_in_uscanada(in millions), dtype: float64
```

Figure 2.2.12

Oscar:

```
movies_df['Oscar'].describe()
```

```
count    27003.000000
mean         2.781284
std          2.392742
min          0.000000
25%          1.000000
50%          2.000000
75%          4.000000
max         11.000000
Name: Oscar, dtype: float64
```

Figure 2.2.13

Wins:

```
movies_df['wins'].describe()
```

```
count    27003.000000
mean        67.134615
std         76.100442
min          1.000000
25%         15.000000
50%         43.000000
75%         85.000000
max        355.000000
Name: wins, dtype: float64
```

Figure 2.2.14

Nominations:

```
movies_df['nominations'].describe()
```

```
count    27003.000000
mean        89.549235
std         89.272559
min          0.000000
25%         23.000000
50%         61.000000
75%        124.000000
max        369.000000
Name: nominations, dtype: float64
```

Figure 2.2.15

Genres:

```
movies_df['genres'].describe()
```

```
count     27003
unique      138
top       Drama
freq       3152
Name: genres, dtype: object
```

Figure 2.2.16

Writers:

```
movies_df['Writers'].describe()
```

```
count                 26999
unique                  430
top       Christopher Nolan
freq                    869
Name: Writers, dtype: object
```

Figure 2.2.17

Languages:

```
movies_df['languages'].describe()
```

```
count       25608
unique         49
top       English
freq         11195
Name: languages, dtype: object
```

Figure 2.2.18

Origin_country:

```
movies_df['origin_country'].describe()
```

```
count             26832
unique               58
top       United States
freq             12366
Name: origin_country, dtype: object
```

Figure 2.2.19

Production_companys:

```
movies_df['production_companys'].describe()
```

```
count            24420
unique             256
top       Warner Bros.
freq              2090
Name: production_companys, dtype: object
```

Figure 2.2.20

CHAPTER III- DATA CLEANING AND NORMALIZATION

3.1 Introduction

Data cleaning is the process of detecting and correcting errors, inconsistencies, and inaccuracies in the dataset to enhance its quality. It involves identifying and handling missing values, removing duplicates, correcting data formats, and ensuring data validity. Cleaning the IMDb dataset was necessary to make it more reliable for analysis and database storage. By systematically addressing data issues, we ensured that inconsistencies were eliminated, redundant entries were removed, and data was structured for efficient querying and processing.

Normalization, on the other hand, is the process of organizing data within a database to reduce redundancy and improve efficiency. This step ensures that data is structured in a way that prevents anomalies and enhances database performance. The normalization process included breaking down multi-value fields into separate tables, ensuring referential integrity, and storing data in a relational format for easy access and retrieval.

- Handling Missing Values: Filling missing numerical data and cleaning currency fields.
- Removing Duplicates: Identifying and eliminating redundant records.
- Standardizing Data Formats: Fixing motion picture ratings and converting data types.
- Normalizing Data: Splitting multi-value columns and storing data in separate SQL tables.
- Validating Data: Cross-checking with IMDb API and performing manual data integrity checks.

3.2 Handling Missing Values

Missing values were handled by applying appropriate transformations:

- Filling Missing Numerical Data:

```
df['budgets'] = df['budgets'].fillna(0)
df['gross_us_and_canada']=df['gross_us_and_canada'].fillna(0)
df['gross_worldwide']=df['gross_worldwide'].fillna(0)
df['opening_weekend_gross_in_uscanada']=df['opening_weekend_gross_in_uscanada'].fillna(0)
```

Figure 3.2.1

- Cleaning Money Fields:

```
def clean_money(budget):
    if not budget:
        return 0.0

    if isinstance(budget, (int, float)):
        return float(budget)

    if "$" in budget and "(estimated)" in budget:
        budget = budget.replace("$", "").replace("(estimated)", "").strip()

    budget = budget.replace(",", "")

    try:
        return float(budget)
    except ValueError:
        return 0.0

df['budgets']=df['budgets'].apply(lambda budget: clean_money(budget))
```

```
def clean_money(value):
    if pd.isna(value):
        return 0.0

    if isinstance(value, str):
        value = value.replace("$", "").replace(",", "").strip()

    try:
        return float(value)
    except ValueError:
        return 0.0
columns_to_clean = ['gross_us_and_canada', 'gross_worldwide', 'opening_weekend_gross_in_uscanada']

df[columns_to_clean] = df[columns_to_clean].apply(lambda col: col.map(clean_money))

pd.options.display.float_format = '{:.2f}'.format
```

Figure 3.2.2

3.3 Handling Duplicate Records

Duplicate entries were checked and removed to ensure data integrity:

- Checked for duplicates:

```
df.duplicated().sum()
```

```
np.int64(0)
```

Figure 3.3.1

- Removed duplicate rows while keeping the most complete entry:

```
df.drop_duplicates(inplace=True)
```

Figure 3.3.2

3.4 Standardizing Data Formats

Standardization ensures consistency across different data fields:

- Fixing Motion Picture Ratings:

```
replace_rating=['1h 27m','2h 21m','2h 27m','2h 29m']
df['Motion_picture_Rating'] = df['Motion_picture_Rating'].replace(replace_rating, "Not Rated")
```

Figure 3.4.1

- Converting Data Types:

```
movies_df['movie_id']=movies_df['movie_id'].astype('int32')
movies_df['Motion_picture_Rating']=movies_df['Motion_picture_Rating'].astype('category')
```

Figure 3.4.2

3.5 Data Normalization

To avoid redundancy, the dataset was structured into multiple tables:

- Splitting Multi-Value Columns:

```
def explode_column(df, column_name, new_column_name):
    df_expanded = df[['movie_id', column_name]].copy()  # Use movie_id directly
    df_expanded[column_name] = df_expanded[column_name].apply(lambda x: eval(x) if isinstance(x, str) and x.startswith('[') else [])
    df_expanded = df_expanded.explode(column_name).dropna().reset_index(drop=True)
    df_expanded.rename(columns={column_name: new_column_name}, inplace=True)
    return df_expanded

# Create separate DataFrames
writers_df = explode_column(df, 'Writers', 'writer_name')
genres_df = explode_column(df, 'Genre', 'genre_name')
production_companies_df = explode_column(df, 'Production_company', 'company_name')
languages_df = explode_column(df, 'Languages', 'language_name')
origin_df = explode_column(df, 'Origin', 'origin_country')

df = df.drop(columns=['Writers', 'Genre', 'Production_company', 'Languages', 'Origin'])
```

Figure 3.5.1

- Storing Data in SQL Tables:

Create Database and Store the Normalized DataFrame as Tables

```
[17] conn=sqlite3.connect(os.path.join("../data/clean/","imdb_movie_details.db"))

[18] df.to_sql("movies",conn, if_exists='replace', index=False)
...  246

[19] writers_df.to_sql("Writers",conn, if_exists='replace', index=False)
...  548

[20] genres_df.to_sql("Geners",conn, if_exists='replace', index=False)
...  1477

[21] production_companies_df.to_sql("Production_companys",conn, if_exists='replace', index=False)
...  404

[22] languages_df.to_sql("Languages",conn, if_exists='replace', index=False)
...  395

[23] origin_df.to_sql("Origins",conn, if_exists='replace', index=False)
...  432
```

Figure 3.5.2

CHAPTER IV DATABASE DESIGN

4.1 Introduction

The **IMDb Movie Database** is designed to efficiently store and manage detailed information about movies, including their metadata, financial performance, and industry recognition. This structured relational database captures key aspects of films, such as titles, directors, IMDb ratings, release years, and financial earnings. Additionally, it maintains relationships between movies and their associated writers, genres, production companies, languages, and countries of origin. By utilizing a **normalized schema**, the database ensures **data integrity, minimizes redundancy, and optimizes query performance**.

At the core of the database is the **Movies table**, which acts as the primary entity, linking to several supporting tables that store additional attributes such as writers, genres, and production details. Each movie can belong to multiple genres, be produced by multiple companies, and have multiple associated languages, making it a **many-to-many relational model** in some aspects. The design follows **Third Normal Form (3NF)**, ensuring efficient data storage and retrieval while maintaining consistency.

This structured database is beneficial for **film analysts, data scientists, and entertainment industry professionals** looking to perform **in-depth data analysis, trend forecasting, and reporting**. By organizing movie data in a relational format, it allows users to efficiently query and extract meaningful insights about the film industry, making it a **powerful tool for movie analytics and business intelligence**.

4.2 Database Schema Overview

The IMDb movie database schema consists of **six primary tables**, each serving a specific purpose:

- **Movies**: Stores fundamental movie details such as title, director, IMDb ratings, release year, financial statistics, and award recognitions.
- **Writers**: Associates movies with their respective screenwriters.
- **Genres**: Classifies movies into different genres.
- **Production Companies**: Identifies the production companies involved in the making of each movie.

- **Languages**: Specifies the languages in which a movie is available.
- **Origins**: Indicates the country of origin for each movie

4.3 Detailed Table Structures

4.3.1 Movies Table

Column	Data Type	Description
movie_id	INTEGER	Unique identifier for each movie (Primary Key)
title	TEXT	Movie title
directors	TEXT	Directors of the movie
imdb_ratings	REAL	IMDb rating of the movie
release_year	INTEGER	Year of release
runtime	TEXT	Duration of the movie
motion_picture_rating	TEXT	Age rating of the movie (e.g., PG, R)
budget	REAL	Production budget of the movie
gross_us_canada	REAL	Earnings in the US and Canada
gross_worldwide	REAL	Total worldwide earnings
opening_weekend_gross	REAL	Revenue generated in the opening weekend in US/Canada
Oscar	INTEGER	Number of Oscars won
wins	INTEGER	Total number of awards won
nominations	INTEGER	Total number of award nominations

4.3.2 Writers Table

Column	Data Type	Description
movie_id	INTEGER	Foreign Key referencing Movies (movie_id)
writer_name	TEXT	Name of the screenwriter

4.3.3 Genres Table

Column	Data Type	Description
movie_id	INTEGER	Foreign Key referencing Movies (movie_id)
genre_name	TEXT	Genre classification (e.g., Action, Drama)

4.3.4 Production Companies Table

Column	Data Type	Description
movie_id	INTEGER	Foreign Key referencing Movies (movie_id)
company_name	TEXT	Name of the production company

4.3.5 Languages Table

Column	Data Type	Description
movie_id	INTEGER	Foreign Key referencing Movies (movie_id)
language_name	TEXT	Language of the movie

4.3.6 Origins Table

Column	Data Type	Description
movie_id	INTEGER	Foreign Key referencing Movies (movie_id)
origin_country	TEXT	Country of origin

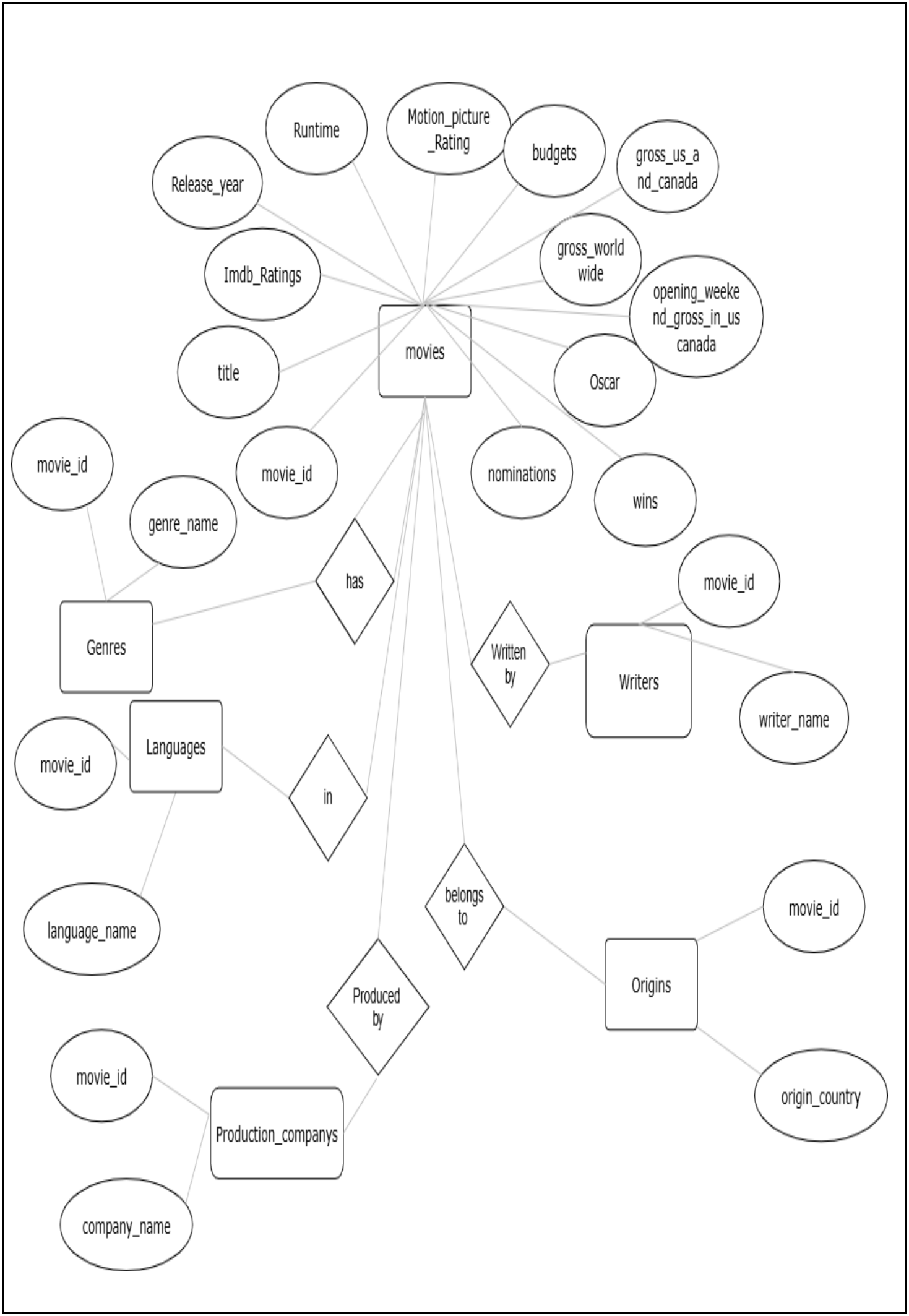
4.4 Entity Relationship (ER) Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.ER diagram used for

- ER diagrams represent the E-R model in a database, making them easy to convert into relations (tables).
- ER diagrams provide the purpose of real-world modeling of objects which makes them intently useful.
- ER diagrams require no technical knowledge of the underlying DBMS used.
- It gives a standard solution for visualizing the data logically.

ER Model is used to model the logical view of the system from a data perspective which consists of these symbols:

- **Rectangles:** Rectangles represent Entities in the ER Model.
- **Ellipses:** Ellipses represent Attributes in the ER Model.
- **Diamond:** Diamonds represent Relationships among Entities.
- **Lines:** Lines represent attributes to entities and entity sets with other relationship types.
- **Double Ellipse:** Double Ellipses represent Multi-Valued Attributes.
- **Double Rectangle:** Double Rectangle represents a Weak Entity.


Runtime
Motion_picture_Rating
budgets
gross_us_and_canada
Release_year
gross_worldwide
Imdb_Ratings
opening_weekend_gross_in_us canada
movies
title
Oscar
movie_id
movie_id
nominations
wins
genre_name
has
movie_id
Genres
Written by
Writers
writer_name
movie_id
Languages
in
language_name
belongs to
movie_id
Origins
Produced by
movie_id
origin_country
Production_companys
company_name

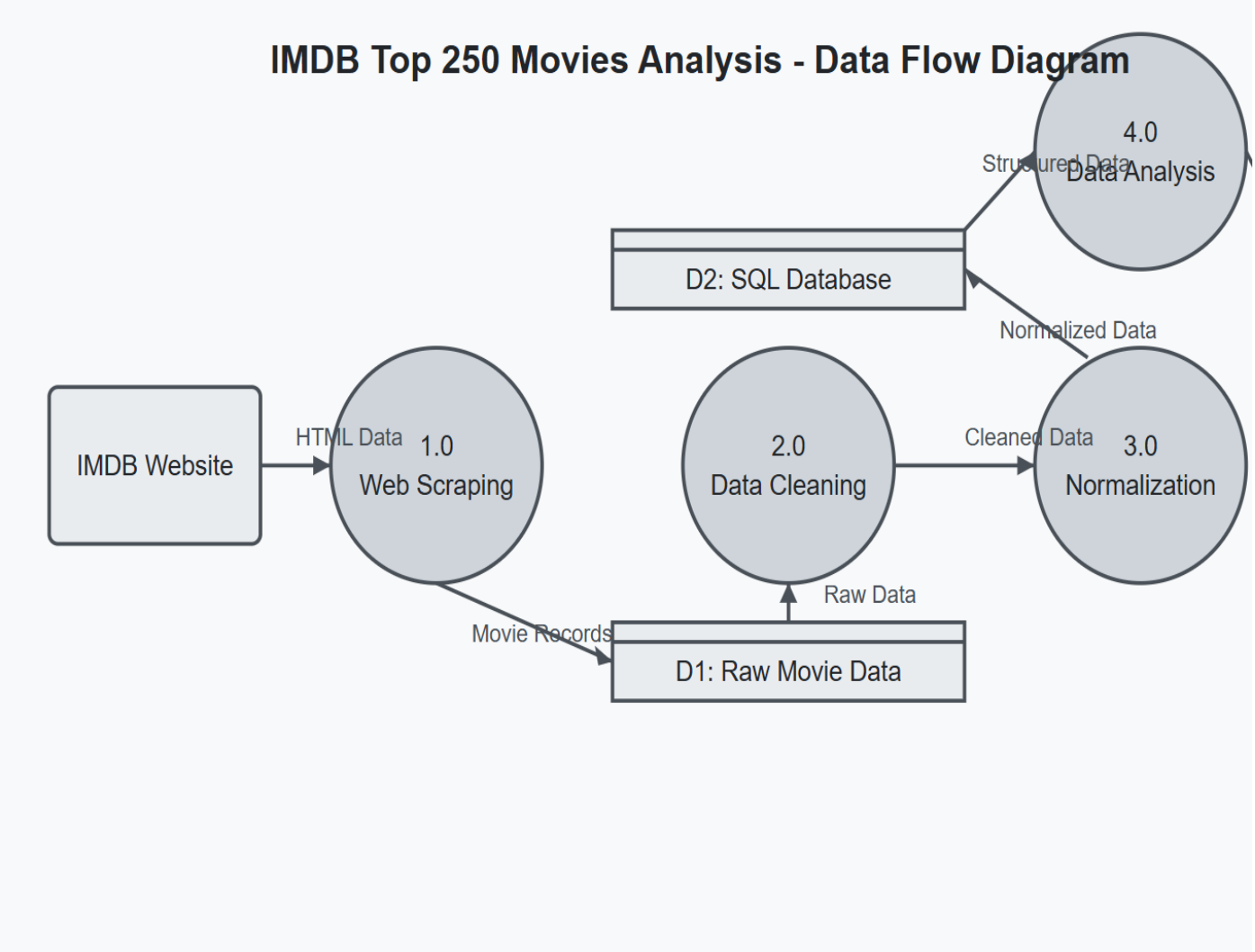
4.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation that depicts the flow of data through the IMDB Top 250 Movies Analysis system. This diagram illustrates how data moves from external sources through various processing steps to its final destination. For this project, the DFD helps visualize the entire data pipeline from data collection to analysis.

Purpose

The purpose of this DFD is to provide a clear understanding of:

- How movie data is collected from the IMDB website
- The transformation processes applied to the raw data
- The storage mechanisms used for the processed data
- The flow of information to the analysis phase



CHAPTER V – EXPLORATORY DATA ANALYSIS

5.1 Introduction

Exploratory Data Analysis (EDA) is an important first step in data science projects. It involves looking at and visualizing data to understand its main features, find patterns, and discover how different parts of the data are connected.

EDA helps to spot any unusual data or outliers and is usually done before starting more detailed statistical analysis or building models.

Exploratory Data Analysis (EDA) is important for several reasons, especially in the context of data science and statistical modeling. Here are some of the key reasons why EDA is a critical step in the data analysis process:

- Helps to understand the dataset, showing how many features there are, the type of data in each feature, and how the data is spread out, which helps in choosing the right methods for analysis.
- EDA helps to identify hidden patterns and relationships between different data points, which help us in and model building.
- Allows to spot errors or unusual data points (outliers) that could affect your results.
- Insights that you obtain from EDA help you decide which features are most important for building models and how to prepare them to improve performance.
- By understanding the data, EDA helps us in choosing the best modeling techniques and adjusting them for better results.

5.2 Data visualization

Data visualization is the graphical representation of information. In this guide we will study what is Data visualization and its importance with use cases.

Data visualization translates complex data sets into visual formats that are easier for the human brain to understand. This can include a variety of visual tools such as:

- **Charts**: Bar charts, line charts, pie charts, etc.

- **Graphs**: Scatter plots, histograms, etc.
- **Maps**: Geographic maps, heat maps, etc.
- **Dashboards**: Interactive platforms that combine multiple visualizations.

The primary goal of data visualization is to make data more accessible and easier to interpret allow users to identify patterns, trends, and outliers quickly. This is particularly important in big data where the large volume of information can be confusing without effective visualization techniques.

Why it is important ,Accessibility, Identifying trends and patterns, Marketing, Better business decisions, Risk assessment, Supply chain efficiency

```
movies_df.columns
```

```
Index(['movie_id', 'title', 'directors', 'Imdb_Ratings', 'release_year',
       'Runtime', 'Motion_picture_Rating', 'budgets(in millions)',
       'gross_us_and_canada(in millions)', 'gross_worldwide(in millions)',
       'opening_weekend_gross_in_uscanada(in millions)', 'Oscar', 'wins',
       'nominations', 'genres', 'Writers', 'languages', 'origin_country',
       'production_companys'],
      dtype='object')
```

Figure 5.2.1

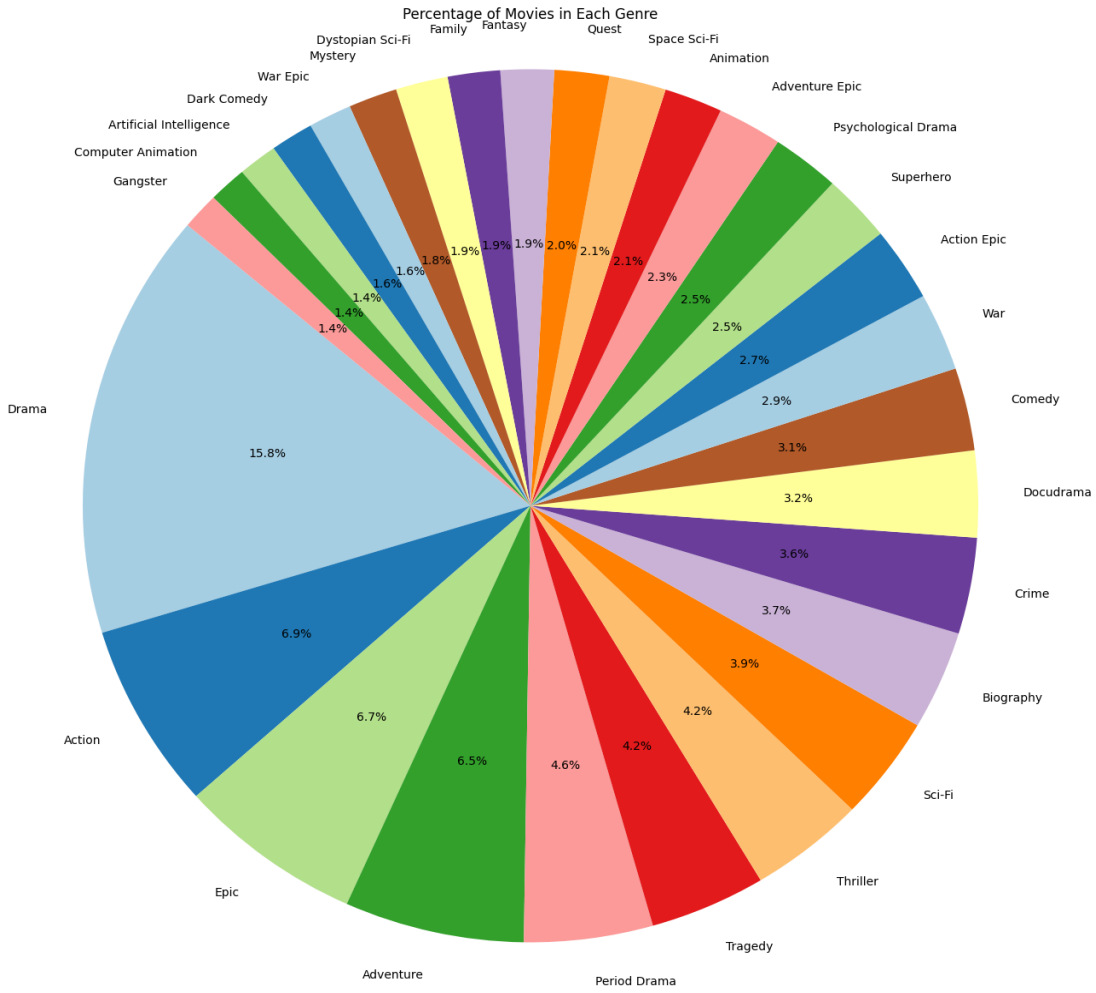
```
movies_df.dtypes
```

✓ 0.0s

```
movie_id                                        int32
title                                          object
directors                                      object
Imdb_Ratings                                  float64
release_year                                    int64
Runtime                                        object
Motion_picture_Rating                        category
budgets(in millions)                            int64
gross_us_and_canada(in millions)                int64
gross_worldwide(in millions)                    int64
opening_weekend_gross_in_uscanada(in millions)  int64
Oscar                                           int64
wins                                            int64
nominations                                     int64
genres                                         object
Writers                                        object
languages                                      object
origin_country                                 object
production_companys                            object
dtype: object
```

Figure 5.2.2

Movie Genre Distribution Analysis

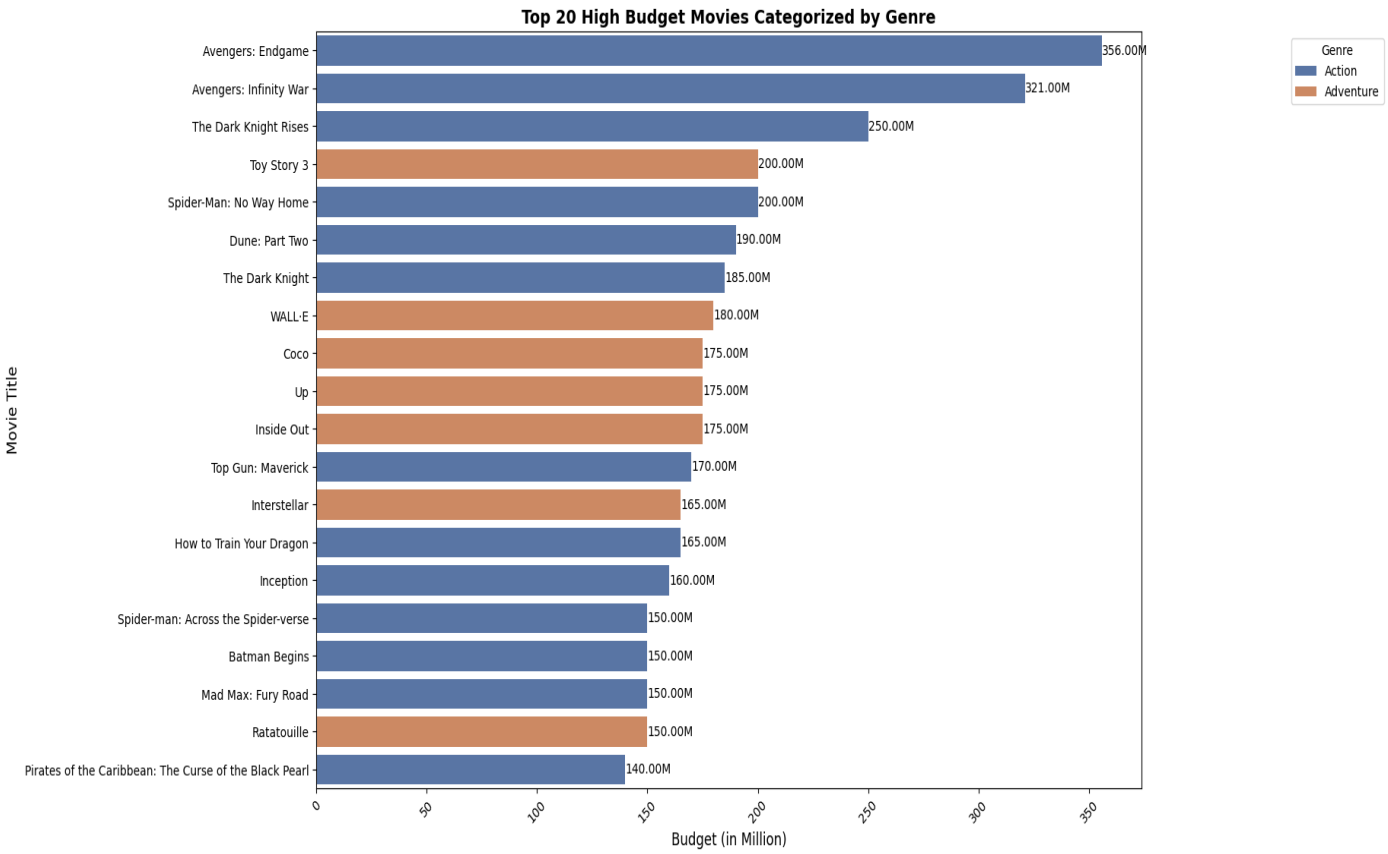


The pie chart illustrates the percentage distribution of movies across various genres. Drama emerges as the most dominant genre, making up **15.8%** of all movies, reflecting its widespread appeal and storytelling depth. Other prominent genres include **Action (6.9%)**, **Epic (6.7%)**, and **Adventure (6.5%)**, which indicates the popularity of high-energy and large-scale storytelling in films.

Additionally, genres like **Thriller (4.2%)**, **Sci-Fi (3.9%)**, and **Crime (3.7%)** contribute significantly to the overall distribution, showcasing the audience's interest in suspenseful and futuristic narratives. Smaller genres such as **Mystery, Dystopian Sci-Fi, and War Epic** each account for around **1-2%**, highlighting niche preferences.

This visualization provides insights into the diversity of movie genres, helping filmmakers, analysts, and industry professionals understand trends in film production and audience demand. The balanced representation of different genres ensures a broad range of storytelling options in the entertainment industry.

Top 20 High Budget Movies Categorized by Genre



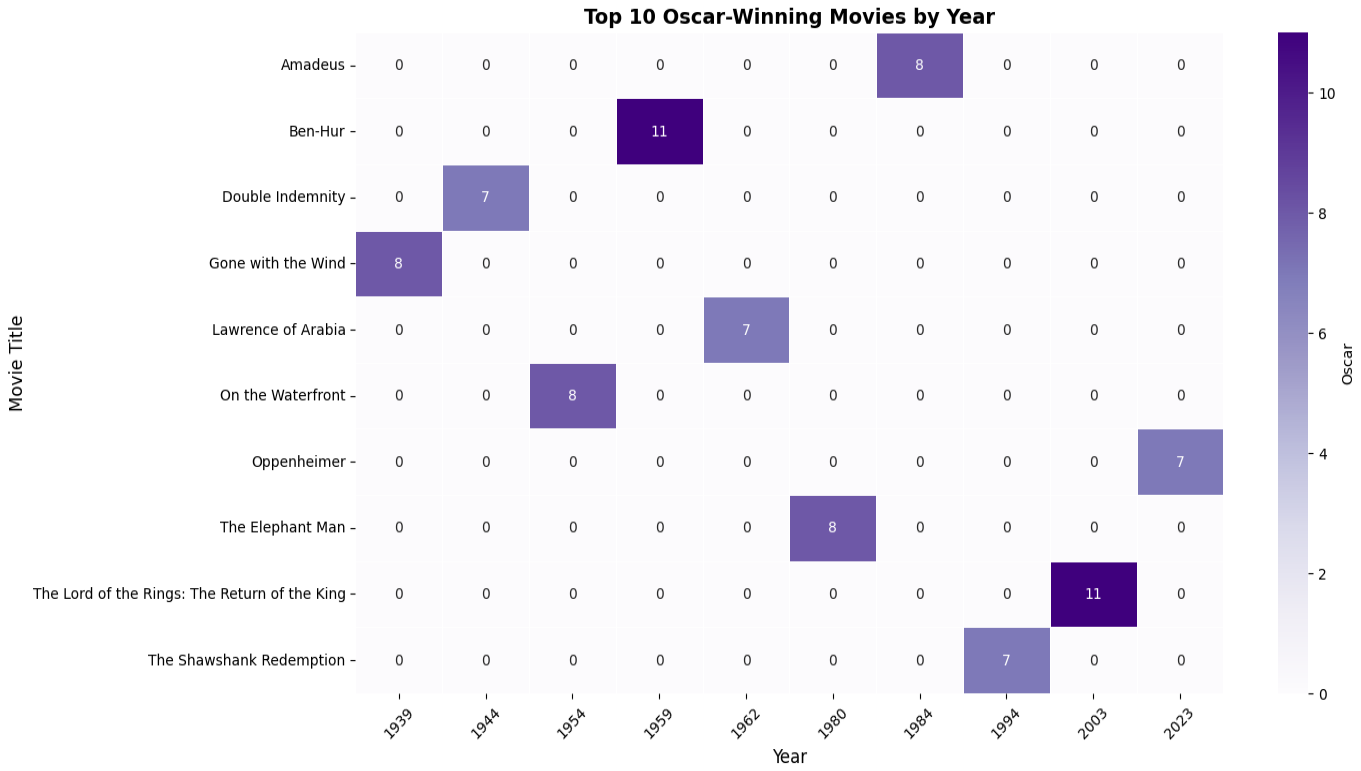
The bar chart presents the **top 20 highest-budget movies**, categorized into **Action** and **Adventure** genres. The budget values are represented in **millions of dollars (M)**.

Among the highest-budget movies, **"Avengers: Endgame"** leads with **$356M**, followed by **"Avengers: Infinity War" ($321M)** and **"The Dark Knight Rises" ($250M)**, all belonging to the **Action** genre. This suggests that action movies, particularly superhero franchises, tend to have the highest production costs due to their extensive use of CGI, stunts, and large-scale production efforts.

In the **Adventure** category, animated movies such as **"Toy Story 3" ($200M)**, **"WALL-E" ($180M)**, and **"Coco" ($175M)** also feature among the highest-budget films. This highlights that animated adventure films require substantial investments in animation technology, voice acting, and post-production work.

The chart effectively showcases the dominance of **Action** movies in terms of budget, with Adventure movies also making a significant mark. The distinction between genres helps in understanding the financial trends in the film industry and how different movie categories allocate their budgets.

Top 10 Oscar-Winning Movies by Year



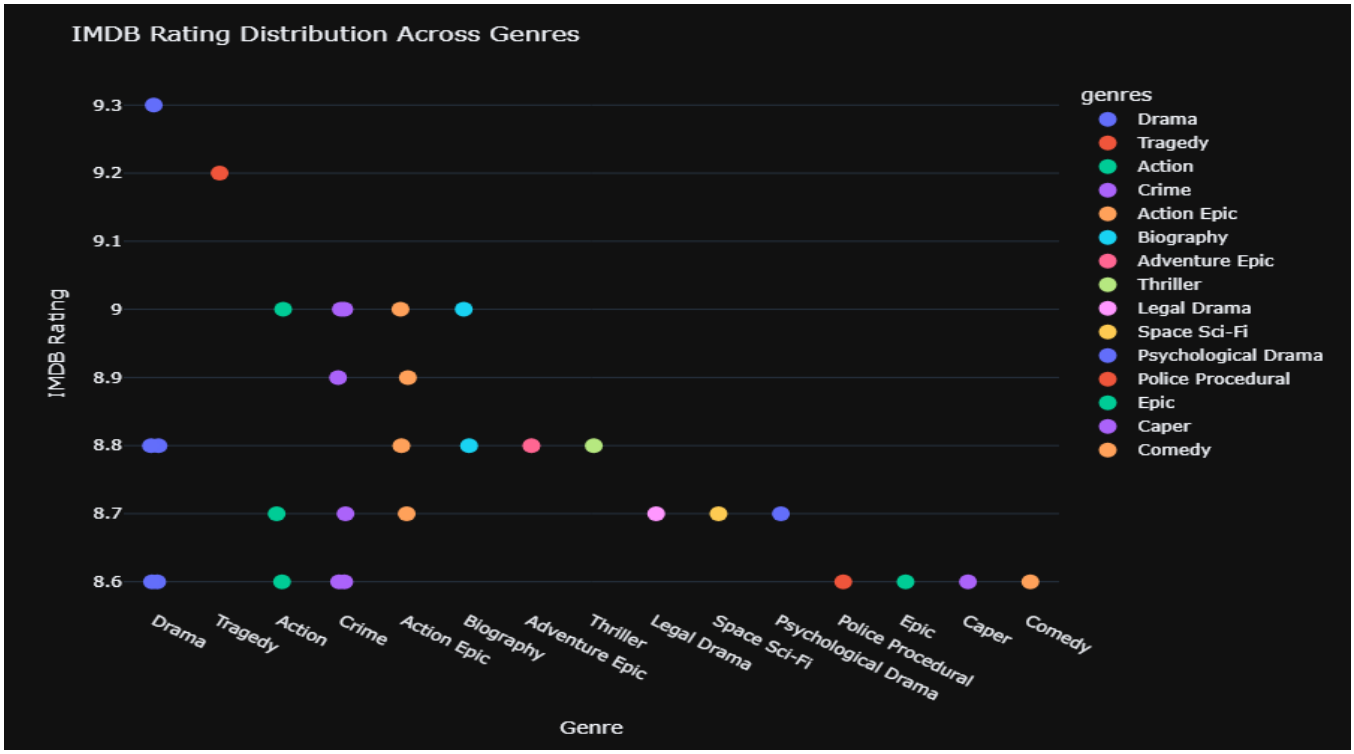
The heatmap visualizes the **Oscar wins** for the **top 10 movies**, distributed across different years. The intensity of the color represents the number of Oscars won, with darker shades indicating a higher number of awards.

Key observations from the chart:

- **"Ben-Hur" (1959) and "The Lord of the Rings: The Return of the King" (2003)** stand out as the **most awarded films**, each winning **11 Oscars**.
- Other highly awarded movies include **"Gone with the Wind" (1939) and "On the Waterfront" (1954), each securing 8 Oscars.**
- The recent film **"Oppenheimer" (2023)** also made it into the top list, winning **7 Oscars**, showing its significant success at the Academy Awards.
- The distribution of Oscar-winning movies spans from **1939 to 2023**, highlighting how film excellence has evolved over the decades.

This visualization effectively showcases the historic dominance of certain films in the Academy Awards and provides insight into the trends of highly acclaimed movies across different eras.

IMDB Rating Distribution Across Genres



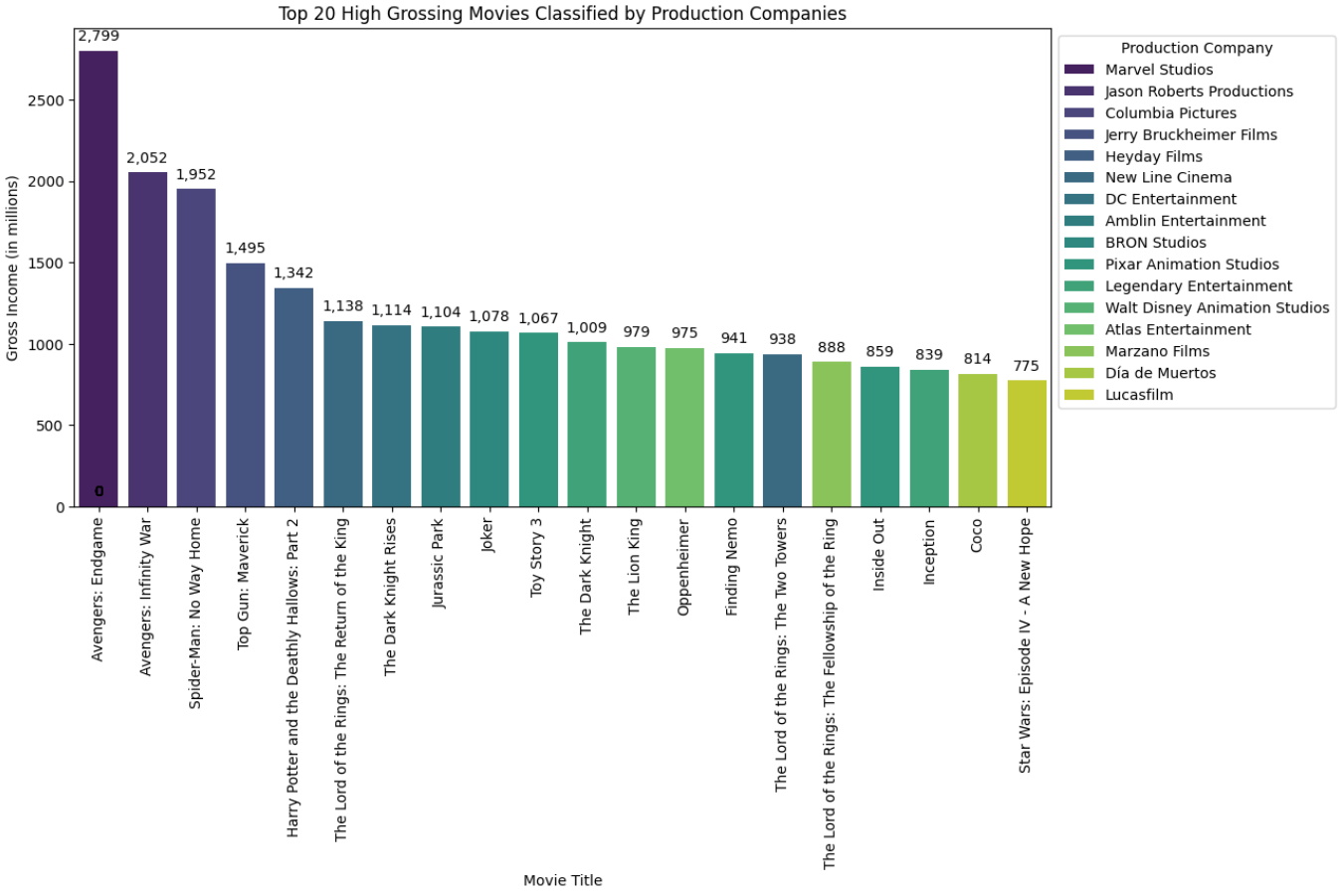
The scatter plot presents the **IMDB ratings** of movies across various genres, showcasing how different film categories are rated by audiences. Each dot represents a specific movie, and its position along the y-axis indicates its IMDB rating. The colors correspond to different genres as shown in the legend.

Key insights from the visualization:

- **Drama and Tragedy genres** feature some of the **highest-rated movies**, with ratings exceeding **9.2**.
- **Action, Crime, and Thriller movies** generally have ratings ranging from **8.6 to 9.0**, indicating a strong but slightly lower audience reception compared to top-rated dramas.
- **Genres such as Comedy and Space Sci-Fi** have fewer high-rated movies, with most ratings clustering around **8.6 to 8.8**.
- The spread of ratings suggests that **certain genres tend to receive higher critical acclaim**, particularly **Drama, Tragedy, and Action Epics**.

This visualization provides an insightful comparison of how well different genres are received by audiences based on IMDB ratings. It highlights patterns in film ratings and helps in understanding genre preferences among viewers.

Top 20 High-Grossing Movies Classified by Production Companies



This bar chart presents the **top 20 highest-grossing movies**, categorized by their **production companies**. The y-axis represents the **gross income (in millions of dollars)**, while the x-axis lists the movie titles. Different colors represent different production studios.

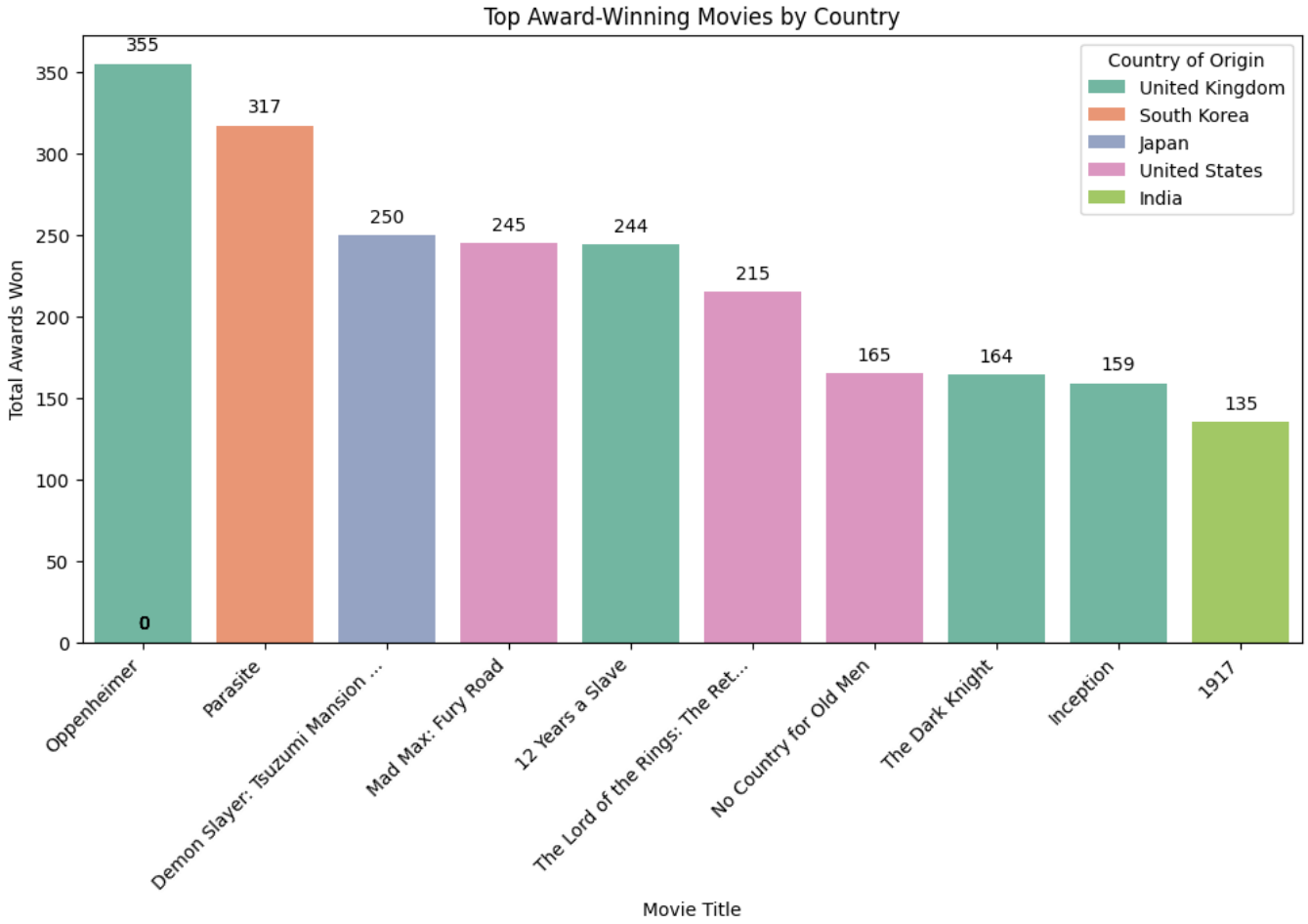
Key Insights:

- **Marvel Studios dominates** the list, with *Avengers: Endgame* leading at **$2.799 billion**, followed by *Infinity War* (**$2.052 billion**) and *Spider-Man: No Way Home* (**$1.952 billion**).
- **Top Gun: Maverick**, produced by **Jerry Bruckheimer Films**, stands out as one of the highest-grossing non-superhero films with **$1.495 billion**.

- **Fantasy and Sci-Fi franchises** like *Harry Potter*, *The Lord of the Rings*, *Jurassic Park*, and *Star Wars* consistently appear, reinforcing their long-standing box-office success.
- **Animated films** such as *Finding Nemo*, *Inside Out*, and *Coco* by **Pixar and Disney Animation Studios** show strong audience appeal, grossing between **$814 million and $975 million**.
- **Oppenheimer**, a historical drama by **Legendary Entertainment**, has also crossed **$975 million**, proving that well-crafted, non-franchise films can also achieve massive success.

This visualization provides a **comprehensive overview of box-office trends**, showcasing how **franchises, animation studios, and major production companies contribute to global movie revenues**.

Top Award-Winning Movies by Country



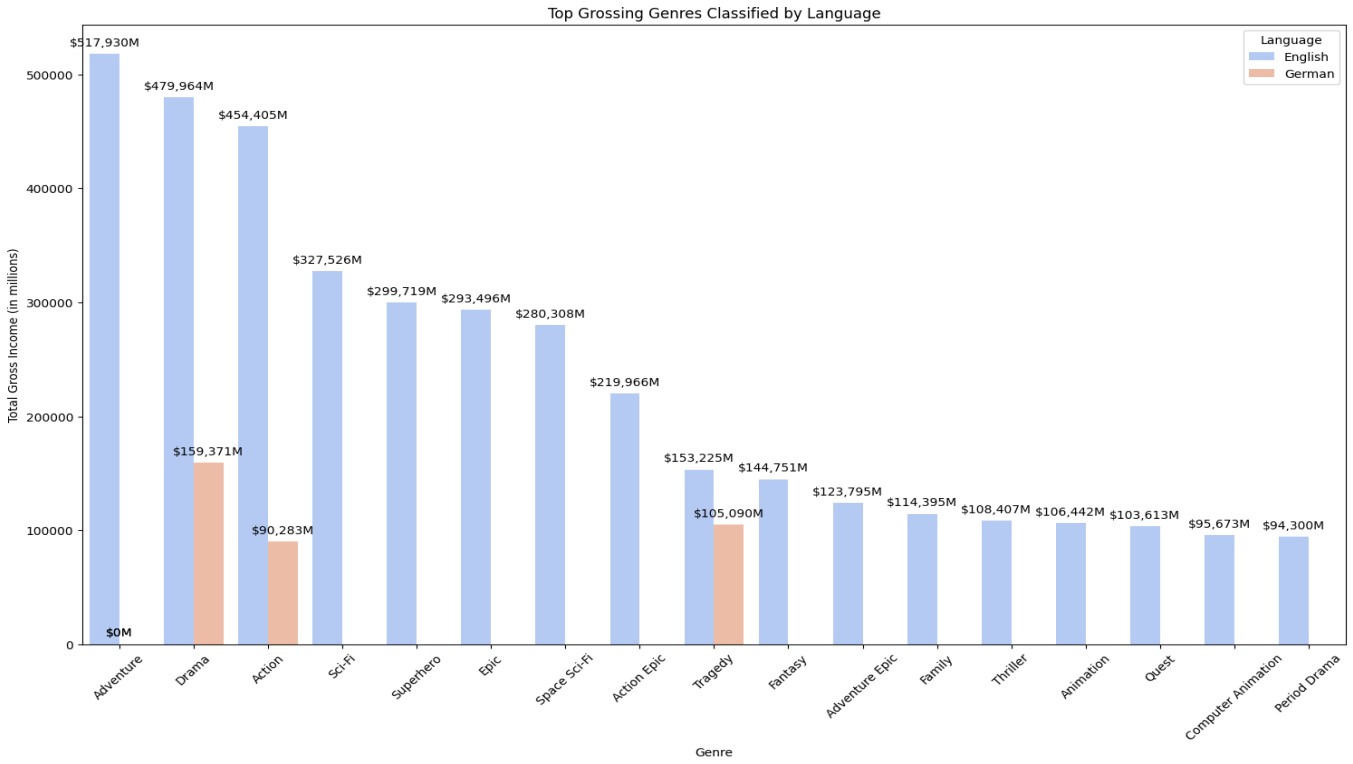
This bar chart represents the **top award-winning movies**, categorized by their **country of origin**. The y-axis shows the **total number of awards won**, while the x-axis lists the movie titles. Different colors represent different countries.

Key Insights:

- **Oppenheimer (United Kingdom) leads with 355 awards**, making it the most awarded film in this dataset.
- **Parasite (South Korea) follows closely with 317 awards**, showcasing its global recognition as a non-English film.
- **Demon Slayer: Tsuzumi Mansion Arc (Japan) earned 250 awards**, proving the popularity of anime on an international level.
- **Mad Max: Fury Road (United States) and 12 Years a Slave (United States) both won around 245 awards**, indicating strong performances in the action and drama genres.
- **The Lord of the Rings: The Return of the King (United Kingdom) secured 244 awards**, continuing its legendary status in cinema.
- **1917 (India) received 135 awards**, highlighting India's growing global influence in filmmaking.

This visualization showcases **how different countries contribute to award-winning cinema**, with a mix of **Hollywood blockbusters, Asian cinema, and European masterpieces**.

Top Grossing Genres Classified by Language



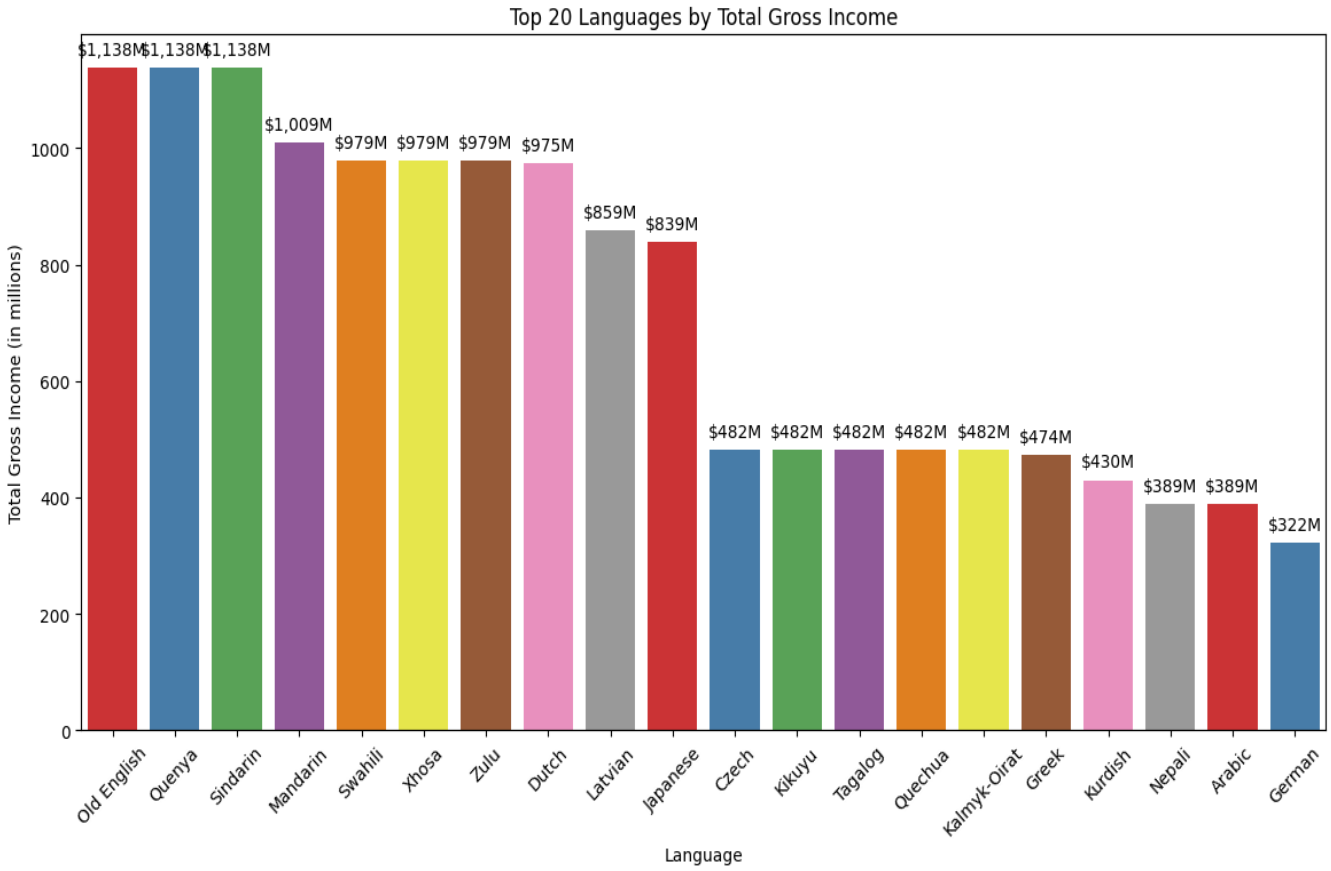
This bar chart represents the **highest-grossing movie genres**, categorized by **language**. The y-axis represents **total gross income (in millions)**, while the x-axis lists the **genres**.

Key Insights:

- **Adventure is the highest-grossing genre**, with over **$517,930M** in earnings (English).
- **Drama (German) is the most successful non-English genre**, generating **$159,371M** in revenue.
- **Action, Sci-Fi, and Superhero genres** continue to dominate global box offices, each surpassing **$290,000M**.
- **The Tragedy genre (German) also performed well**, grossing **$105,090M**.
- **Genres like Animation, Thriller, and Family movies** still hold significant revenue, proving their universal appeal.

This visualization highlights the **strong influence of English-language films** in high-grossing genres while also showcasing **notable contributions from German-language films**.

Top 20 Languages by Total Gross Income

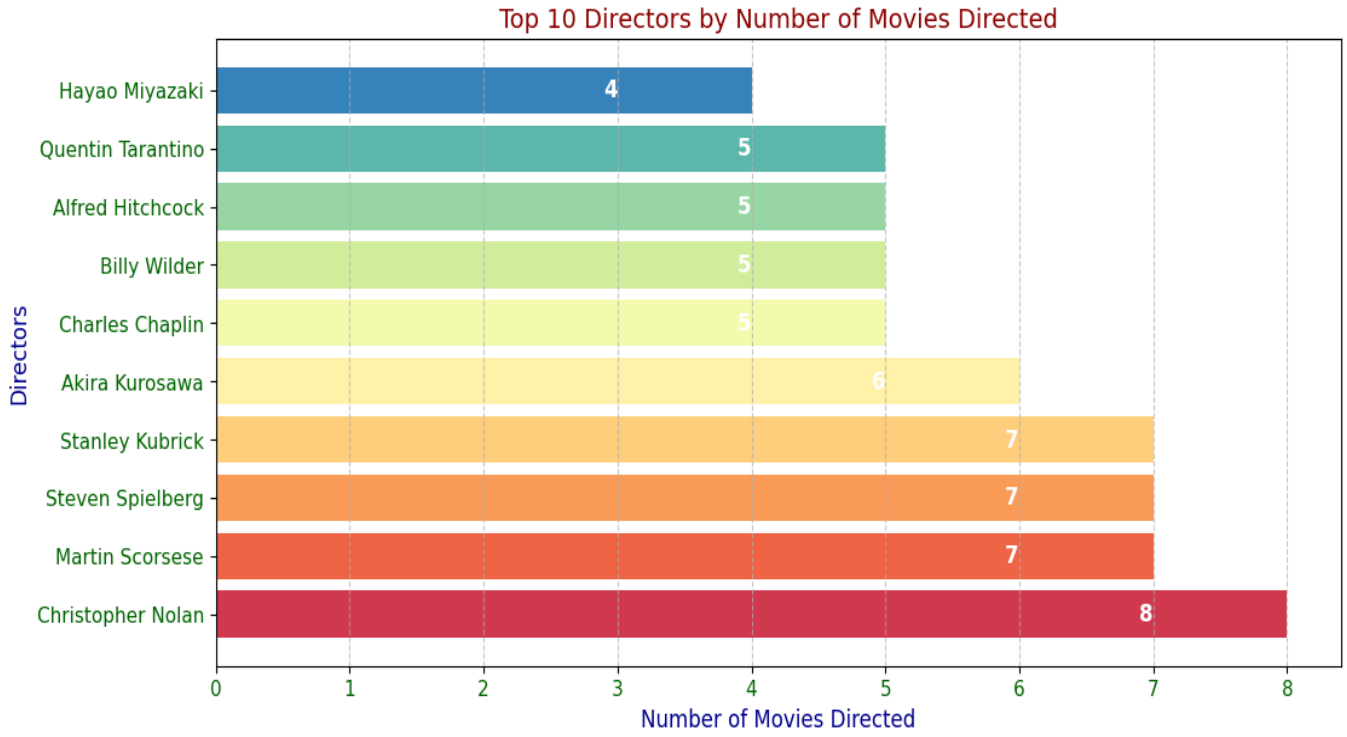


This visualization presents the **top 20 languages** ranked by their **total gross income (in millions)**.

Key Insights:

- **Old English, Quenya, and Sindarin** lead with the highest earnings (**$1,138M** each). These languages are associated with fantasy films like *The Lord of the Rings*.
- **Mandarin ranks high with $1,009M**, showcasing China's growing influence in the global film industry.
- **Swahili, Xhosa, and Zulu** have significant earnings (~$979M), reflecting the success of African culture in cinema.
- **Japanese and Dutch also make a strong showing**, earning **$839M** and **$859M**, respectively.
- **German ranks the lowest on the list with $322M**, indicating a smaller yet still notable market share.

Top 10 Directors by Number of Movies Directed

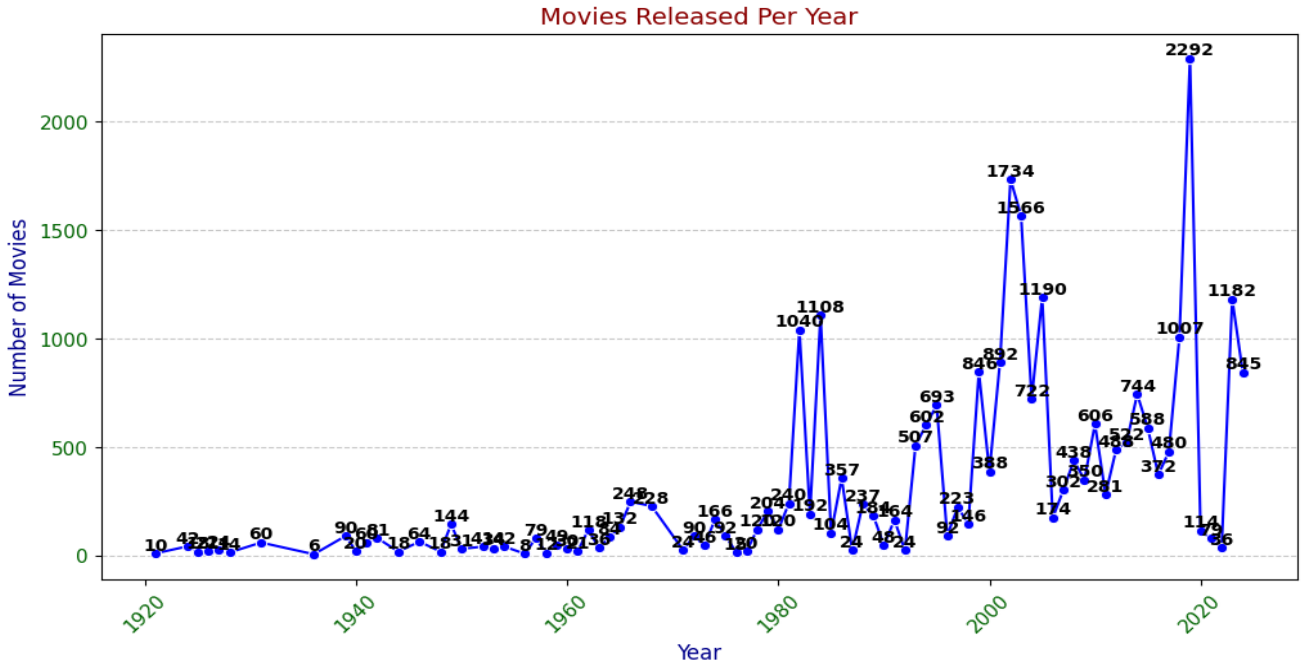


This bar chart shows the **top 10 directors** ranked by the **number of movies they have directed**.

Key Insights:

- **Christopher Nolan** leads with **8 movies**, closely followed by **Scorsese, Spielberg, and Kubrick with 7 each**.
- **Akira Kurosawa (6) and Charles Chaplin (5) are among legendary directors with fewer films**.
- **Hayao Miyazaki has directed the fewest movies (4) among the top 10**, but his impact is undeniable.

Movies Released Per Year

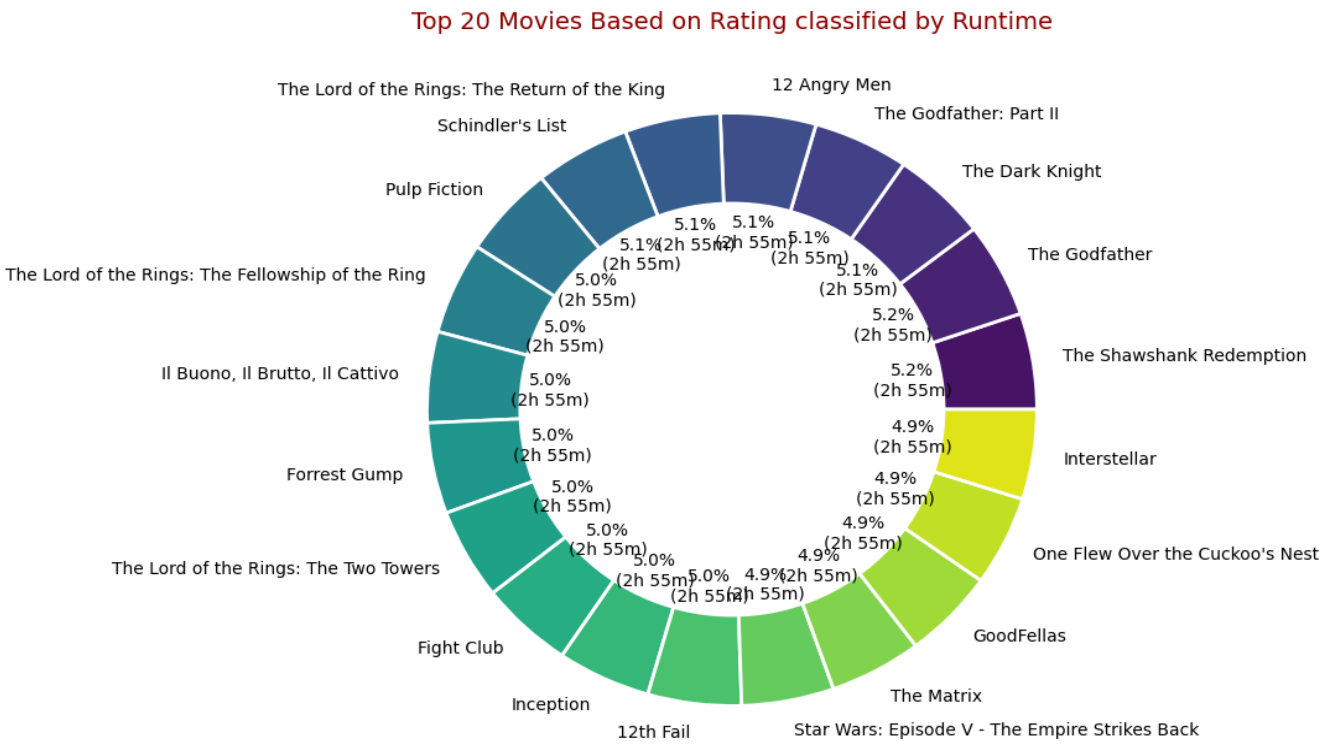


This line chart illustrates the **number of movies released per year** after removing duplicates.

Key Insights:

- **Early 1900s:** Very few movies were released annually.
- **1980s-1990s:** The film industry saw **consistent growth**, crossing **1000 movies in some years**.
- **2000s Boom:** A **sharp increase** in movie releases, peaking in 2010 with **1734 films**.
- **Biggest Spike (2022):** The highest number of movies released in a year was **2292**.
- **Fluctuations in 2020s:** Likely impacted by **COVID-19**, causing a dip and a rebound.

Top 20 Movies Based on Rating classified by Runtime



The donut chart represents the **top 20 highest-rated movies**, displaying their **rating distribution** and **individual runtimes**. Each segment corresponds to a movie, with percentage values indicating its rating share.

Key Insights:

- The chart includes highly acclaimed films like *The Shawshank Redemption*, *The Godfather*, *12 Angry Men*, and *The Dark Knight*.
- Ratings range from **4.9% to 5.2%**, showing that these movies are closely ranked in terms of audience and critic appreciation.
- Each movie is labeled with its runtime, but all appear to be **2h 55m**, which may be a generalization or a data issue.
- The **color-coded sections** help differentiate movies, making it easier to compare their ratings visually.

This chart provides an overview of the **top-rated movies**, illustrating how they compare in terms of **ratings and runtime trends**. A more detailed breakdown of actual runtimes could further enhance its accuracy.

In conclusion, this data visualization project provides critical insights into the global film industry's patterns and trends. The analysis reveals how Drama dominates content production (15.8% of all films), while Action and Adventure genres command the highest budgets and box office returns, exemplified by Marvel Studios' "Avengers: Endgame" ($356M budget, $2.799B gross). We've identified clear correlations between genre, budget allocation, and financial performance, while tracking the industry's exponential growth from minimal early 1900s production to 2,292 releases in 2022.

The visualizations further highlight the international evolution of cinema, with significant contributions from non-English productions like South Korea's "Parasite" (317 awards) and substantial revenue from Mandarin and other language films. This comprehensive analysis serves stakeholders across the industry value chain - from production companies allocating resources based on genre performance to investors identifying emerging market trends and filmmakers recognizing underserved audience segments with growth potential.

CHAPTER – VI CONCLUSION

This comprehensive analysis of the IMDB Top 250 movies dataset has provided valuable insights into the patterns and trends that define cinematic excellence and commercial success in the film industry. Through systematic data collection, cleaning, normalization, and exploratory data analysis, we have uncovered several key findings that contribute to our understanding of what makes a movie critically acclaimed and commercially successful.

The research revealed that Drama dominates the genre distribution, accounting for 15.8% of all top-rated films, followed by Action (6.9%) and Epic (6.7%), highlighting audience preferences for emotionally engaging and high-impact storytelling. High-budget productions, particularly in the Action and Adventure genres, demonstrated significant commercial success, with Marvel Studios' blockbusters like "Avengers: Endgame" leading both in terms of budget allocation ($356M) and worldwide gross revenue ($2.799B).

Our analysis of Oscar-winning films showed that classics like "Ben-Hur" (1959) and "The Lord of the Rings: The Return of the King" (2003) remain the most decorated with 11 Academy Awards each, while recent productions such as "Oppenheimer" (2023) have also achieved significant recognition with 7 Oscars, demonstrating the enduring standards of cinematic excellence across decades.

The data also highlighted the global nature of film production and consumption, with non-English productions gaining substantial recognition. South Korea's "Parasite" earned 317 awards, ranking second only to the UK's "Oppenheimer" (355 awards), while films featuring fantasy languages (Old English, Quenya, Sindarin) and Mandarin showed impressive gross earnings, indicating the expanding international market for diverse cinematic content.

Director analysis revealed that filmmakers like Christopher Nolan (8 films), Martin Scorsese (7 films), and Steven Spielberg (7 films) have consistently produced highly-rated movies, demonstrating the impact of directorial vision on film quality and audience reception. The temporal analysis showed exponential growth in film production, from minimal releases in the early 1900s to a peak of 2,292 movies in 2022, with notable fluctuations during the COVID-19 pandemic period.

This project successfully implemented a normalized database design that efficiently organizes movie data across six primary tables, facilitating complex queries and relationships between

different aspects of film production. The schema design supports future research and allows for scalable data management as the dataset grows.

The insights derived from this study are valuable for various stakeholders in the film industry. Production companies can make informed decisions about genre selection and budget allocation based on historical performance data. Investors can identify emerging trends and potential growth areas in the market. Filmmakers and content creators can recognize underserved audience segments and align their creative vision with market preferences. Additionally, streaming platforms and distributors can optimize their content acquisition strategies based on genre performance and audience reception patterns.

Future research could expand on this foundation by incorporating sentiment analysis of audience reviews, exploring the impact of streaming platforms on traditional movie metrics, and developing predictive models for box office performance based on pre-release indicators. The methodology and database structure established in this project provide a solid framework for such extended analyses.

In conclusion, this data-driven approach to understanding the IMDB Top 250 movies has bridged the gap between film studies and data analytics, offering evidence-based insights that can guide strategic decision-making in the entertainment industry while enhancing our appreciation of the evolving art of cinema.

CHAPTER - VII BIBLIOGRAPHY

https://www.imdb.com/chart/top/

https://chatgpt.com/c/67911ae5-d974-800e-8632-f9e569bcf484

https://www.geeksforgeeks.org/introduction-of-er-model/

https://pypi.org/project/selenium/

https://www.geeksforgeeks.org/relationships-in-sql-one-to-one-one-to-many-many-to-many/