**Project Documentation: IMDb Movie Data Extraction and Analysis**

**By: Faisal Rafiq**

**1. Project Overview**

**Objective**

The goal of this project is to:

* Extract movie-related data from the IMDb website.
* Organize the extracted data into relational SQL tables.
* Analyze the data using Power BI to generate visualizations and insights.
* Make a dashboard

**Target Website**

* **Website:** IMDb (Internet Movie Database)
* **Data to Extract:**
  + Movie Titles
  + Directors
  + Cast & Crew
  + Release Dates
  + Genres
  + IMDb Ratings
  + User Reviews
  + Box Office/Budget
  + Technical Details (runtime, languages, filming locations)
  + Trailer & Pictures

**Tools and Technologies**

* Python (for web scraping using BeautifulSoup)
* SQL Database (MS-SQL)
* Power BI (for visualizing the data)
* Pandas (for data manipulation in Python)
* BeautifulSoup (for web scraping)

**2. Planning and Website Analysis**

**2.1 Data Exploration on IMDb**

Before starting the extraction, we’ll visit IMDb and identify the pages containing the desired data:

* **Main Page:** IMDb Home
* Navigate to movie titles, individual movie pages, and sections like cast, genres, box office, reviews, etc.

**Structure of IMDb Menu**

The menu is broken down into the following main sections:

**Movies**

* + Release Calendar
  + Top 250 Movies
  + Most Popular Movies
  + Browse Movies by Genre
  + Top Box Office
  + Showtimes & Tickets

**Movie News**

* + India Movie Spotlight
  + TV Shows
  + What’s on TV & Streaming
  + Top 250 TV Shows
  + Most Popular TV Shows
  + Browse TV Shows by Genre
  + TV News

**Watch**

* + What to Watch
  + Latest Trailers
  + IMDb Originals
  + IMDb Picks
  + IMDb Spotlight
  + IMDb Podcasts

**Awards & Events**

* + Oscars
  + Emmys
  + TIFF
  + Festival Season
  + Hispanic Heritage Month
  + STARmeter Awards
  + Awards Central
  + All Events

**Celebs**

* + Born Today
  + Most Popular Celebs
  + Celebrity News

**Community**

* + Help Center
  + Contributor Zone
  + Polls

**Explanation:**

**1. Movies**

* **Release Calendar:** Extract upcoming movie release dates.
* **Top 250 Movies:** Extract the IMDb Top 250 Movies list, including rank, title, rating, and year of release.
* **Most Popular Movies:** Extract the most popular movies based on IMDb user votes.
* **Browse Movies by Genre:** List all movie genres and scrape data related to movies under each genre.
* **Top Box Office:** Scrape the current box office numbers for top-performing movies.
* **Showtimes & Tickets:** Provide information on where and how to buy tickets for upcoming movies.
* **Movie News:** Scrape the latest movie news and updates.
* **India Movie Spotlight:** Extract data specifically highlighting movies from India.

**2. TV Shows**

* **What’s on TV & Streaming:** Scrape information about TV shows currently airing or available on streaming platforms.
* **Top 250 TV Shows**: Extract the IMDb Top 250 TV Shows, including rank, title, rating, and year.
* **Most Popular TV Shows:** Scrape the most popular TV shows as per IMDb ratings.
* **Browse TV Shows by Genre:** Extract data related to TV shows categorized by genres.
* **TV News:** Extract the latest TV news.

**3. Watch**

* **What to Watch:** Suggest movies or TV shows based on trending topics or user preferences.
* **Latest Trailers:** Scrape the latest movie and TV show trailers.
* **IMDb Originals:** List and extract IMDb's original series or shows.
* **IMDb Picks:** Scrape recommendations made by IMDb editors.
* **IMDb Spotlight:** Highlight featured movies or shows.
* **IMDb Podcasts:** Provide data on the latest IMDb podcasts.

**4. Awards & Events**

* **Oscars:** Scrape data related to Oscars nominations and winners.
* **Emmys:** Extract data related to Emmy nominations and winners.
* **TIFF:** Provide data from the Toronto International Film Festival (TIFF).
* **Festival Season:** Scrape data from various movie festivals.
* **Hispanic Heritage Month:** Highlight movies and TV shows related to Hispanic heritage.
* **STARmeter Awards:** Scrape data from the IMDb STARmeter awards.
* **Awards Central:** List various other awards and winners.
* **All Events:** Extract data from all events IMDb covers.

**5. Celebs**

* **Born Today:** Extract a list of celebrities born on the current date.
* **Most Popular Celebs:** Scrape IMDb’s most popular celebrity ranking.
* **Celebrity News:** Provide the latest news on celebrities.

**6. Community**

* **Help Center:** General help information from IMDb.
* **Contributor Zone:** Information about contributing to IMDb.
* **Polls:** Scrape data related to IMDb user polls and results.

**Movie Details to Scrape:**

* **Movie Title:** 12 Angry Men
* **Year of Release:** 1957
* **Duration:** 1h 36min
* **Rating:** 9.0/10
* **Genres:** Crime, Drama
* **Plot/Description:** "The jury in a New York City murder trial is frustrated by a single member whose skeptical caution forces them to more carefully consider the evidence before jumping to a hasty verdict."
* **Director:** Sidney Lumet
* **Stars:** Henry Fonda, Lee J. Cobb, Martin Balsam
* **Trailer Link:** (Optional)

**2.2 IMDb Structure**

IMDb’s page structure follows a fairly consistent pattern:

* **Movie Page URL:** https://www.imdb.com/title/{movie\_id}/
* **Example:** https://www.imdb.com/title/tt4154796/ for **Avengers: Endgame.**
* **Key elements we will need to scrape:**
  + **Movie Title:** Available in <h1> or a specific class.
  + **Cast:** Can be found in the "Cast" section, usually in a table.
  + **Director, Writers, and Producers**: Displayed in the "Details" section.
  + **Genres:** Listed under the movie information.
  + **Rating:** Displayed in a prominent location on the movie page.
  + **Release Date:** In the "Details" section.
  + **Plot Summary:** Available under the "Storyline" section.
  + **Box Office Data:** Found under the "Box Office" section.
  + **And more.**

**2.3 IMDb Data Access**

IMDb offers two primary ways to access data:

* **Web Scraping:** Using Python libraries like BeautifulSoup.
* **IMDbPY:** A Python package designed to extract IMDb data programmatically.

**3. Web Scraping/Extracting Data**

**3.1 Choosing a Method**

**Option 1: Use BeautifulSoup for manual scraping.**

* Pros: Full control over data extraction.
* Cons: Requires handling page structure changes.

**Option 2: Use IMDbPY, a pre-built library for accessing IMDb data.**

* Pros: Less manual work, structured output.
* Cons: Limited customization for scraping specific data fields.

**3.2 Python Web Scraping Script (BeautifulSoup Example)**

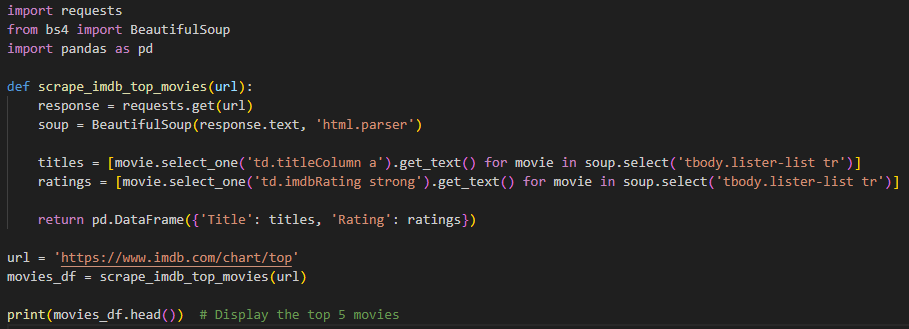
**Step 1: Setup Python Environment**

**Install necessary libraries:**

pip install requests beautifulsoup pandas

**Step 2: Basic IMDb Scraper**

A sample Python script for scraping movie titles and ratings:

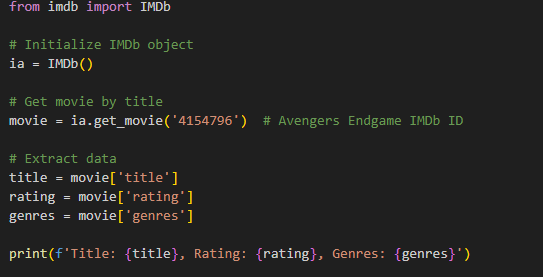


**3.3 IMDbPY Example**

Alternatively, if using IMDbPY:

pip install IMDbPY

**Sample code:**

****

**4. Data Modeling: SQL Database**

**4.1 Database Schema Design**

Here, we’ll create a relational schema that mirrors IMDb’s structure. Below is a possible design:

**Movies Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| movie\_id | **INT** | Unique ID |
| title | **VARCHAR** | Movie Title |
| release\_year | **INT** | Release Year |
| rating | **FLOAT** | IMDb Rating |
| box\_office | **BIGINT** | Box Office Revenue |
|  |  |  |

**Cast Table**

| **Column Name** |  | **Data Type** | **Description** |
| --- | --- | --- | --- |
| movie\_id |  | **INT** | Linked to Movies Table |
| actor\_id |  | **INT** | Linked to Actors Table |
| role |  | **VARCHAR** | Role Played |

**Genres Table**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| genre\_id | INT | Unique Genre ID |
| genre\_name | VARCHAR | Genre Name |
| movie\_id | INT | Linked to Movies Table |
|  |  |  |

**4.2 Creating SQL Tables**

Once you have your schema defined, you can use SQL to create these tables.

**SQL Script Example (MySQL):**

**CREATE TABLE Movies** (

movie\_id INT PRIMARY KEY,

title VARCHAR(255),

release\_year INT,

rating FLOAT,

box\_office BIGINT

);

**CREATE TABLE Cast** (

cast\_id INT PRIMARY KEY,

movie\_id INT,

actor\_id INT,

role VARCHAR(255),

FOREIGN KEY (movie\_id) REFERENCES Movies(movie\_id)

);

**CREATE TABLE Genres** (

genre\_id INT PRIMARY KEY,

movie\_id INT,

genre\_name VARCHAR(50),

FOREIGN KEY (movie\_id) REFERENCES Movies(movie\_id)

);

**5. Data Analysis with Power BI**

**5.1 Connecting SQL to Power BI**

* Launch Power BI.
* Click on "Get Data" and select SQL Server or MySQL (depending on your DBMS).
* Provide your database credentials and import your tables.

**5.2 Creating Visualizations**

* **Charts:** Bar charts showing movie ratings, genres, box office collections.
* **Time Series:** Visualize the trend of movie ratings over time.
* **Heat Maps:** Show relationships between genres and ratings.

**Example charts to build:**

* Top 10 Rated Movies by Genre
* Box Office Revenue over Time
* Average Ratings for Each Genre

**6. Final Steps & Deployment**

**6.1 Data Cleaning and Optimization**

* Ensure all data is accurate, clean, and free of duplicates.
* Normalize values (e.g., ratings, genres) for consistency.

**6.2 Power BI Dashboard Publishing**

After creating your visualizations, publish your Power BI report to the Power BI service for sharing and collaboration.

**7. Conclusion**

By following the outlined steps, we will be able to extract movie data from IMDb, store it in a structured format, and visualize it using Power BI to derive valuable insights from the data.