Hollywood Actors and Actresses

Bararu Robert Daniel

26 November 2024

Contents

1 What is this project about? **3**

2 Tools **4**

3 Algorithm **5**

4 Modules **6**

5 Data Structure **7**

1 What is this project about?

This project involves the development of a sophisticated software application designed to extract, analyze, and process data related to the top 50 popular Hollywood actors and actresses. The software will integrate with **IMDb (Internet Movie Database)**, one of the most comprehensive online databases providing information about films, television series, podcasts, home videos, video games, and online streaming content. The core objective of this software is to deliver a user-friendly and highly functional platform that provides detailed insights into the careers and achievements of these actors and actresses. The software will feature the following key functionalities:

1. **Comprehensive List:** Display a complete list of the 50 actors and actresses included in the project scope.
2. **Profiles:** Provide detailed information about each actor/actress, including their biography and career highlights.
3. **Filmography:** Present an organized list of all their movies, along with release years.
4. **Awards:** Showcase the awards won by each actor/actress, categorized by year and type.
5. **Genres:** Analyze and display the movie genres they have worked in throughout their careers.
6. **Movie Ratings:** Calculate and present the average ratings of their movies, both overall and on a year-by-year basis.
7. **Top 5 Movies:** Highlight their top 5 highest-rated movies, including release years and associated genres.

This software will be developed using **Python**, leveraging its rich ecosystem of libraries and tools tailored for data science, web scraping, and data visualization. The application will emphasize clarity, intuitive navigation, and visually appealing presentation of data, making it accessible and engaging for users.

2 Tools

This software will be designed using **PyCharm**, a robust integrated development environment (IDE) developed by the Czech company **JetBrains**. PyCharm is widely used for Python programming due to its comprehensive features, including intelligent code assistance, debugging tools, and seamless integration with libraries and frameworks.

To establish a connection with IMDb and retrieve the required data, two approaches are being considered:

1. **IMDb Official API:** The official IMDb API, accessible via **Amazon Web Services (AWS)** under the names IMDb Data or IMDb Content for Amazon Customers, is a reliable source designed primarily for professional and commercial use. This API provides structured and regularly updated data, including information about movies, TV shows, actors, awards, and more. However, the IMDb API is not free, and its cost varies depending on the services chosen. Advanced features and scalability increase the associated expenses, which might make it less suitable for academic or personal projects.

2. **IMDbPy:** An alternative to the official API is **IMDbPy**, a Python library designed to interact with IMDb data. This tool enables data extraction through web scraping or local database integration, making it a powerful tool for personal and academic projects. IMDbPy is:

   - **Free and Open-Source:** Distributed under the GPL (General Public License), allowing unrestricted use and modification.
   - **Versatile:** Capable of accessing details about movies, TV shows, cast, crew, and ratings.

   Despite its advantages, IMDbPy comes with certain limitations:

   - **Unofficial Status:** IMDbPy is not an official IMDb product. Its use might violate IMDb's terms of service, particularly for large-scale scraping or commercial applications.
   - **No Guaranteed Access:** Since it does not use an official API, there is no agreement with IMDb ensuring data availability or consistency.
   - **Performance:** Direct scraping is slower than using an API, especially when dealing with large datasets. Additionally, IMDb could implement anti-scraping measures that might disrupt functionality.

To efficiently handle the retrieved information, the software will likely incorporate a **database** to store and manage data about actors, movies, awards, genres, and ratings. Databases such as **SQLite, PostgreSQL, MySQL** will be considered based on the project's scalability and performance requirements. This will allow for fast and organized retrieval of data during software operation.

3 Algorithm

The algorithms developed for this project will primarily focus on the extraction, processing, and presentation of data related to the top 50 Hollywood actors and actresses. A key component of these algorithms will be the use of the IMDbPy library, which will facilitate the scraping of IMDb's website. The following is an outline of the main algorithms to be implemented:

1. **Data Extraction:** The **IMDbPy** library will be employed to scrape the IMDb website. The data collected will encompass basic information about the top 50 actors and actresses, including their names, biographies, filmographies, awards.

2. **Data Storage:** Once the data has been collected, it will be saved in a database. This guarantees structured and persistent storage, making the data easily accessible for subsequent analysis. For specific analyses, data will be extracted from the database and loaded into Pandas DataFrame. This approach combines database persistence and organization with the flexibility and efficiency of Pandas for in-memory data processing.

3. **Data Analysis:** After the data is collected, it will be analyzed to extract meaningful insights, including:

    - **Avarage Ratings:** Calculating the average ratings of the actors' and actresses' films.
    - **Top 5 Movies:** dentifying the top 5 highest-rated movies for each actor/actress, along with their release years and genres.
    - **Awards:** Analyzing the number and type of awards each actor/actress has received over time.

4. **Data Visualization:** After analyzing the data, it is essential to present it in a format that is easily interpretable for the user. For data visualization we will opt to use, for example, Matplotlib to generate graphs and print the Dataframes (Pandas) in the terminal to be able to view lists, data etc...

4 Modules

For the development of this project, several Python modules will be utilized to ensure efficient data retrieval, analysis, and visualization. The key modules include:

1. **NumPy:** This module will be used for performing **numerical calculations** and **high-performance operations** on arrays and matrices. It is particularly suited for handling large datasets and performing complex mathematical operations such as matrix multiplication, linear algebra, and statistical analysis.

2. **Pandas:** Pandas is a powerful library for **data analysis** and **manipulation.** It will be used to manage and structure the data efficiently. The **DataFrame** and **Series** structures in Pandas allow for easy manipulation of structured data, making it an ideal choice for working with large volumes of information, such as actor profiles, movies, and ratings.

3. **Matplotlib:** To visualize the data, **Matplotlib** will be used for creating **graphs and charts**. This module enables the creation of various plots, such as bar charts, line graphs, and histograms, which will help present key insights from the data, such as average movie ratings or distribution of genres.

4. **Requests:** This module is essential for **making HTTP requests**. It will allow the software to interact with external APIs or retrieve web data. Through HTTP requests, the software can access and retrieve information from the IMDb API or other web sources, enabling dynamic data collection and updates.

5. **IMDbPy: IMDbPy** is a Python library designed for **interacting with the IMDb database**. It will be used to retrieve detailed information about movies, actors, actresses, and other related data from IMDb. Although it is not an official IMDb API, IMDbPy allows easy access to a wide range of IMDb data and will be particularly useful for pulling information about filmographies, awards, and ratings.

By integrating these modules, the project will have a robust set of tools for efficient data processing, analysis, and visualization, ensuring that the software is both powerful and user-friendly.

5 Data Structure

The data structures used in this project are crucial,as they directly influence the **management, organization, processing and analysis** of data. The following data structures will be utilized in this project:

1. **Dictionaries:** These are powerful and highly efficient data structures when it comes to storing key-value pairs. They enable fast access to data through a unique key, making it easy to retrieve specific information quickly.For example, it is possible to map each actor to a list of their movies/awards.

2. **DataFrame (Pandas):** A DataFrame is an ideal structure for managing large volumes of structured data. DataFrames allow for the organization of data into rows and columns, where each column holds a specific type of data. Additionally, DataFrames support the manipulation and modification of data with ease and can handle mixed data types. For example, we could analyze the actors' and movies' data in a tabular format.

3. **Lists:** Lists are perfect for storing ordered sequences of elements, such as actors or films. Python lists are mutable, meaning they allow for easy addition, modification, and removal of items. For example, they can be used to store the list of films in which an actor has appeared.

4. **Set:** An unordered collection of elements that does not allow duplicates, meaning that each element must be unique. For example, we can store the list of movie genres or awards, as these are unique and we don't need to worry about duplicates."

5. **Arrays (NumPy):** Numpy arrays are powerful structures designed for efficiently storing and manipulating numerical data. They offer better performance for calculations due to their **implementation in C**. In this project, they can be particulary useful for handling and analyzing film ratings. For example, they can be used to calculate the average rating of an actor's movies.