Project 2: Hollywood Actors and Actresses

Applied Data Science with Python - Mr. Lengyel - IN,IT - WiSe 25/26

Pedro Negri Leão Lambert

November 19, 2025

Project Overview

The primary objective of this project is to apply key data science methodologies—specifically data mining and information visualization—to create a user-friendly software application. The project involves extracting information about the top 50 popular Hollywood actors and actresses from IMDb, processing this data, and presenting it in a new, interactive format. Rather than dealing with purely unstructured data, the challenge lies in mining semi-structured HTML content from the web and transforming it into a clean, actionable dataset that can be easily explored by the web app users. This project then serves as a practical application of the full data pipeline, from extraction and cleaning to aggregation and final presentation in a modern web interface.

To ensure the solution is robust and professionally constructed, I have selected a specific set of industry-standard tools. The backend logic will be implemented using Python chosen for its extensive ecosystem of data manipulation libraries and ease of use, specially for data mining. I will use the FastAPI framework to serve this data because of the relative simplicity of the project and the built-in capability to generate Swagger documentation, which streamlines development with documentation. For the data extraction layer, the BeautifulSoup4 and Requests libraries will be utilized to handle HTTP communication and parse the HTML DOM tree. On the client side, React will be used to build a Single Page Application. All development will be made with Visual Studio Code, with Git and GitHub used for version control.

The core logic of the application relies on two main algorithms: extraction and aggregation. The data extraction process uses a DOM traversal algorithm. It begins by sending a request to the IMDb list URL, parsing the response, and iterating through the main container's child elements to locate the top 50 actors. It then follows the hyperlinks to each actor's profile page to mine specific details like filmography and genres. Following extraction, an aggregation algorithm is applied to calculate statistical metrics, specifically the average movie ratings.

The software architecture follows the Separation of Concerns principle, divided into distinct modules to maintain code quality. The backend consists of a Scraper Module, which isolates the logic for fetching and parsing HTML, and an API Controller, which acts as the orchestrator serving data to the frontend. The frontend is modularized into a Carousel Component for browsing actors and an Actor Card Component for displaying detailed views. This modularity ensures that changes to the extraction logic (e.g., if IMDb changes its layout) do not break the user interface.

To manage the flow of information efficiently, the project utilizes specific data structures. Python lists of dictionaries will serve as the primary internal container for actor data, as they allow for flexible storage of attributes like names, biographies, and nested lists of movies. For the final data transmission between the server and the client, these structures are serialized into JSON objects. Additionally, hash maps (dictionaries) may be employed to look up specific actor details by ID, ensuring O(1) access time when a user selects an actor from the interface.