# Applied Data Science with Python

## Milestone 1

Choice: Project 2, IMDb Software of Hollywood Actors and Actresses

“IMDb is an online database of information related to films, television programs, home videos, video games, and streaming content online – including cast, production crew and personal biographies, plot summaries, trivia, ratings, and fan and critical reviews” (Wikipedia).

This project consists of building a smaller, IMDb-type database, containing the names of the top 50 actors/actresses and information about them. The starting point of the project consists of the web scraping of the IMDb website. This part of code is enabled and powered thanks to the requests and BeautifulSoup libraries.

It all starts by sending a request to the URL containing the list, <https://www.imdb.com/list/ls053501318/> , and “elaborating” such request, creating a BeautifulSoup object from it. At this point, from this page, thanks especially to the findAll() method we are able to retrieve from the list the name and surname of each of the top 50 actors, their Bios, the title of their all-time movie and their specific link: in other words, the actor/actress’ specific link is a link that can be combined with the IMDb main URL (= https://www.imdb.com) to create a URL that will point to the IMDb webpage of the considered actor (e.g.: Christian Bale’s ID = /name/nm0000288/?ref\_=nmls\_hd, combined with the main URL creates the following URL, <https://www.imdb.com/name/nm0000288/?ref_=nmls_hd>, that leads directly to Christian Bale’s IMDb page). However, it is still possible to retrieve the name of the actor, his bio and his all-time movie’s title just from the list, without accessing any other web page.

Once we have each actor/actress’ specific link, we are able to access the IMDb page of every actor, and it is now possible to retrieve the missing pieces of information requested by the assignment about each of the actors/actresses and append them to lists, each of which contains only one piece of information about all actors(for example: information about Awards).

In regard to the awards section, it is possible to access, from the page of every actor, a URL to a table that contains the actor’s main accomplishments. The table can be scraped like we have seen in class, in order to get the information about Academy nominations and victories and their respective year and append them to a list.

Regarding the top 5 movies and their rating instead, it gets more complicated. It is necessary to use the URL to the actor’s highest rated films: <https://www.imdb.com/filmosearch/?explore=title_type&role=nm0000288&ref_=filmo_ref_typ&mode=detail&page=1&title_type=movie&sort=user_rating,desc> , where the highlighted part is the actor/actress’ IMDb ID, a code or number that univocally identifies the actor in the database(in this case it is still about Christian Bale). It is possible to switch the IMDb ID with another actor/actress’ IMDb ID to get to his/her highest rated films page. In this way, we can scrape this page and get the information needed about movie titles, ratings and genres, and append them to lists.

In case of need, information can also be gathered thanks to the IMDbPY library. IMDbPY is a Python, platform independent package, for retrieving and managing the data of the IMDb movie database about movies and people

Once all the information needed is gathered through single lists, we can merge them together using the pandas library and its methods into one big data frame. This also enables us to create a .csv file that contains all the information. To keep the program user friendly and well formatted it is possible to use the Flask library, in order to create an app and to keep it in order and easily approachable by users.