Task 2: Documentation Top 50 Actresses And Actors

Table Of Content

Overview

The software “Hollywood Actors and Actresses” should read the IMDb Top 50 actors list and save it. After that, it should request the movies by the actor's name from the TMDB database. The result should be saved and displayed in a user interface in a table format. The data frame should also be saved as a .csv file.

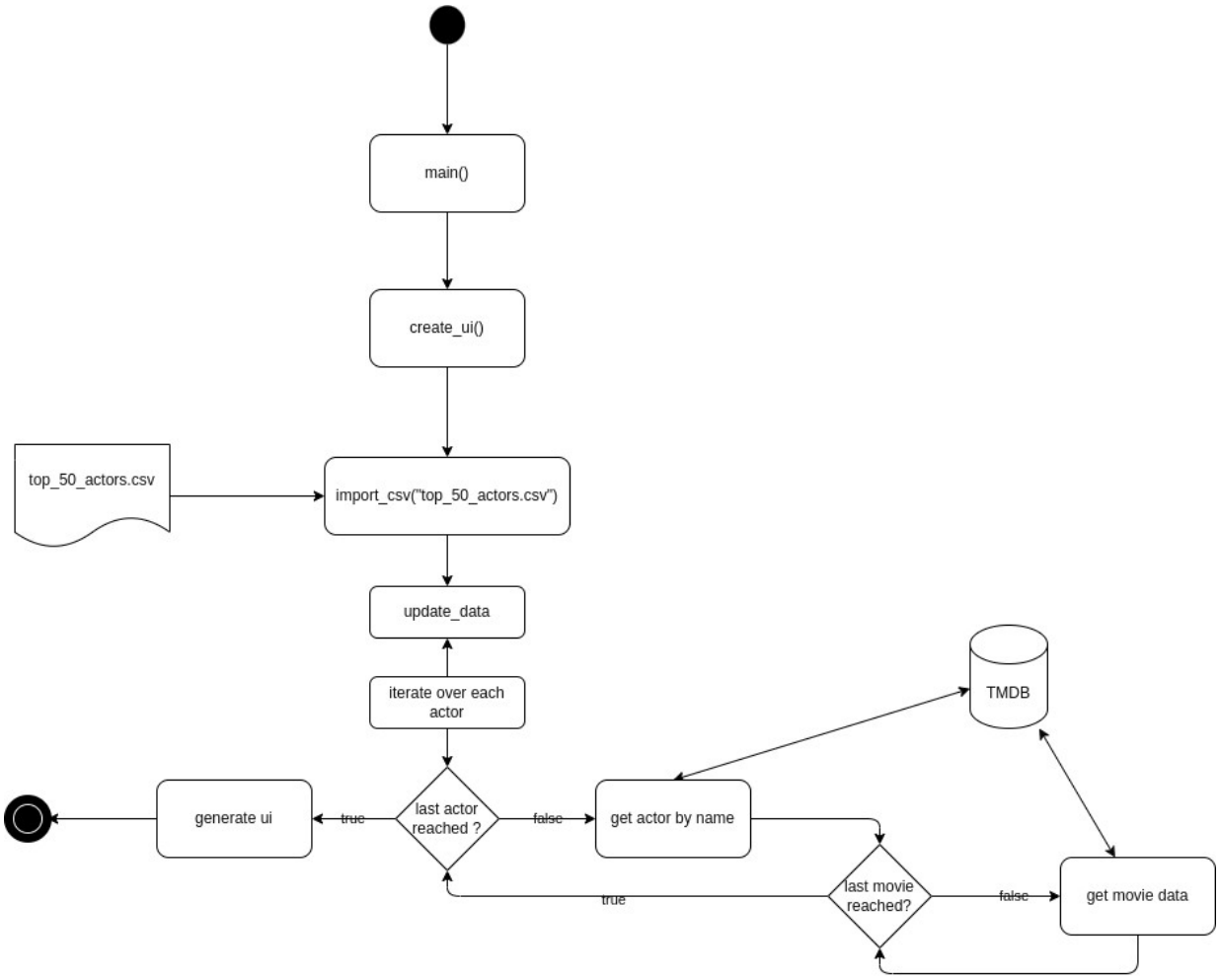
Dataflow

This is the flowchart for the application:

The application starts by invoking the create_ui function, which serves as the main entry point. Inside **create_ui,** a dataframe is created by importing the top 50 actresses and actors. The **create_ui** function then calls the update function to fetch any missing data from the TMDB database. The **update** function iterates over each actress or actor already present in the dataframe. For each individual:

- It checks every movie they were cast in.
- Retrieves any missing information about the actors or their movies.

The retrieved and updated information is displayed in a user-friendly interface in the form of a table.


main()
create_ui()
top_50_actors.csv
import_csv("top_50_actors.csv")
update_data
iterate over each actor
TMDB
generate ui
true
last actor reached ?
false
get actor by name
last movie reached?
true
false
get movie data

Modules

Because there are only two different classes in that project, I did not draw a class diagram. The project is organized in Input (csv and TMDB) , Data (Dataframe) and Output (UI).

ui.py

This module is responsible for creating the user interface of the application using the **Tkinter** library. The top actresses and actors are displayed in a table, which is implemented using the **Treeview** component. Additionally, this module serves as the main file of the project, where the program is initiated.

```python
import tkinter as tk
from tkinter import ttk
import actor_data as ad
from actor_data import *

class ui:

    def __init__(self):
        self.root = None

        self.root = tk.Tk()

        self.root.title("Top 50 Actors")

        # Create a Treeview widget to display the table
        treeview = ttk.Treeview(self.root)

        # Define the column headings
        data = ad.Data()
        data.import_list("top_50_actors.csv")
        data.update_data()
        self.df_actor= data.df
        treeview["columns"] = list(self.df_actor.columns)
        for col in self.df_actor.columns:
            treeview.heading(col, text = col)

        # Insert rows into the Treeview widget
        for index, row in self.df_actor.iterrows():
            treeview.insert('', 'end', values=list(row))

        # Place the Treeview widget in the window
        treeview.pack(expand=True, fill="both")
        self.root.mainloop()

if __name__ == "__main__":
    ui = ui()
```

actor_data.py

The **actor_data** module contains a class designed to manage the dataframe that stores all actor data.

- The `import_list` function initializes the dataframe by reading data from the file **top_50_actors.csv.**
- The **update_data** function retrieves missing information from the TMDB database. It accomplishes this by leveraging functions provided in the **tmdb_connect.py** module.
- **list_to_string** converts a list to a string
- **movie_list_to_string** converts the movie list to a string with the movie names
- **get_avg_rating** calculates the average rating of all the films, the actress or actor was in cast

```python
import pandas as pd
import tmdb_connect as tmdb

class Data:

    def __init__(self):
        self.df = None

    def import_list(self, str_path_list):

        # Reading a CSV file into a DataFrame
        self.df = pd.read_csv('top_50_actors.csv')



    def update_data(self):
        self.df = self.df.drop(["Position","Const","Created","Modified","Description"],axis=1)
        self.df["Movies"] = ""
        self.df["Genres"] = ""
        self.df["Avg Rating"] = ""
        for index, row in self.df.iterrows():
            print(row["Name"])

            actor_details = tmdb.get_actor_movies_with_details(tmdb.get_actor(row["Name"])['id'])

            self.df.at[index, "Genres"] = self.__list_to_string(actor_details[2])
            self.df.at[index, "Movies"] = self.__movie_list_to_string(actor_details[0][:5])
            self.df.at[index, "Rating"] = self.__get_avg_rating(movies= actor_details[0])



    def __list_to_string(self,list_input):…

    def __movie_list_to_string(self, movies_in):…

    def __get_avg_rating(self,movies):…
```

tmdb_connection.py

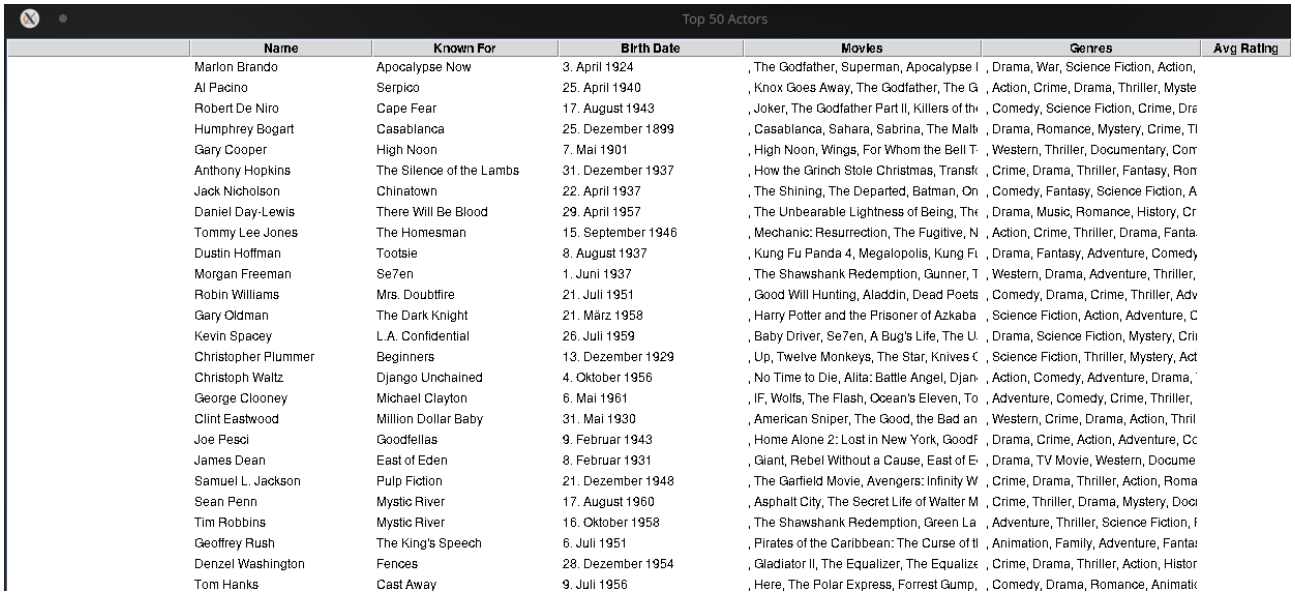
This module implements functions to retrieve data from the TMDB database using the **`simple_tmdb`** library. An API key, generated from TMDB's developer portal, is used for authentication.

- The **search_actor** function retrieves the TMDB ID of a specified actor or actress.
- The **get_actor_movies_with_details** function fetches detailed movie data for a given actor or actress, filling in any missing information.
- The **get_actor** function allows users to enter the name of an actor or actress and returns their details from the TMDB database.

```python
tmdb.API_KEY = 'x'  # Replace with your actual TMDb API key

def search_actor(actor_name):
    """
    Search for an actor by name and return their details.
    """
    search = tmdb.Search()
    response = search.person(query=actor_name)
    if not search.results:
        return None
    # Return the first result's details
    return search.results[0]

def get_actor_movies_with_details(actor_id):
    """
    Get all movies for a given actor with release year, popularity, and genres.
    """
    person = tmdb.People(actor_id)
    response = person.movie_credits()
    cast_movies = response.get('cast', [])
    list_genres = []

    # Add popularity, release year, and genres for each movie
    for movie in cast_movies:
        movie_details = tmdb.Movies(movie['id']).info()
        movie['popularity'] = movie_details.get('popularity', 0)  # Default to 0 if missing
        movie['rating'] = movie_details.get('vote_average', 0)  # Default to 0 for missing ratings
        movie['release_year'] = (
            movie.get('release_date', '').split('-')[0] if movie.get('release_date') else 'Unknown'
        )
        for genre in movie_details.get('genres', []):
            movie['genres'] = genre['name']

            if genre["name"] not in list_genres :
                list_genres.append(genre["name"])

    # Sort movies by popularity (highest first)
    cast_movies.sort(key=lambda m: m['popularity'], reverse=True)

    return cast_movies,  list_genres

def get_actor(str_actor_name):…
```

User Interface

The user interface shows the top 50 actress and actors in a table with their important information. The UI must be update, because it is not very user-friendly yet.



	Name	Known For	Birth Date	Movies	Genres	Avg Rating
	Marlon Brando	Apocalypse Now	3. April 1924	, The Godfather, Superman, Apocalypse I	, Drama, War, Science Fiction, Action,	
	Al Pacino	Serpico	25. April 1940	, Knox Goes Away, The Godfather, The G	, Action, Crime, Drama, Thriller, Myste	
	Robert De Niro	Cape Fear	17. August 1943	, Joker, The Godfather Part II, Killers of the	, Comedy, Science Fiction, Crime, Dra	
	Humphrey Bogart	Casablanca	25. Dezember 1899	, Casablanca, Sahara, Sabrina, The Malte	, Drama, Romance, Mystery, Crime, Th	
	Gary Cooper	High Noon	7. Mai 1901	, High Noon, Wings, For Whom the Bell T	, Western, Thriller, Documentary, Com	
	Anthony Hopkins	The Silence of the Lambs	31. Dezember 1937	, How the Grinch Stole Christmas, Transfo	, Crime, Drama, Thriller, Fantasy, Ron	
	Jack Nicholson	Chinatown	22. April 1937	, The Shining, The Departed, Batman, On	, Comedy, Fantasy, Science Fiction, A	
	Daniel Day-Lewis	There Will Be Blood	29. April 1957	, The Unbearable Lightness of Being, The	, Drama, Music, Romance, History, Cr	
	Tommy Lee Jones	The Homesman	15. September 1946	, Mechanic: Resurrection, The Fugitive, N	, Action, Crime, Thriller, Drama, Fanta	
	Dustin Hoffman	Tootsie	8. August 1937	, Kung Fu Panda 4, Megalopolis, Kung Fu	, Drama, Fantasy, Adventure, Comedy	
	Morgan Freeman	Se7en	1. Juni 1937	, The Shawshank Redemption, Gunner, T	, Western, Drama, Adventure, Thriller,	
	Robin Williams	Mrs. Doubtfire	21. Juli 1951	, Good Will Hunting, Aladdin, Dead Poets	, Comedy, Drama, Crime, Thriller, Adv	
	Gary Oldman	The Dark Knight	21. März 1958	, Harry Potter and the Prisoner of Azkaba	, Science Fiction, Action, Adventure, C	
	Kevin Spacey	L.A. Confidential	26. Juli 1959	, Baby Driver, Se7en, A Bug's Life, The U	, Drama, Science Fiction, Mystery, Crii	
	Christopher Plummer	Beginners	13. Dezember 1929	, Up, Twelve Monkeys, The Star, Knives C	, Science Fiction, Thriller, Mystery, Act	
	Christoph Waltz	Django Unchained	4. Oktober 1956	, No Time to Die, Alita: Battle Angel, Djan	, Action, Comedy, Adventure, Drama,	
	George Clooney	Michael Clayton	6. Mai 1961	, IF, Wolfs, The Flash, Ocean's Eleven, To	, Adventure, Comedy, Crime, Thriller,	
	Clint Eastwood	Million Dollar Baby	31. Mai 1930	, American Sniper, The Good, the Bad an	, Western, Crime, Drama, Action, Thril	
	Joe Pesci	Goodfellas	9. Februar 1943	, Home Alone 2: Lost in New York, Goodf	, Drama, Crime, Action, Adventure, Co	
	James Dean	East of Eden	8. Februar 1931	, Giant, Rebel Without a Cause, East of E	, Drama, TV Movie, Western, Docume	
	Samuel L. Jackson	Pulp Fiction	21. Dezember 1948	, The Garfield Movie, Avengers: Infinity W	, Crime, Drama, Thriller, Action, Roma	
	Sean Penn	Mystic River	17. August 1960	, Asphalt City, The Secret Life of Walter M	, Crime, Thriller, Drama, Mystery, Doc	
	Tim Robbins	Mystic River	16. Oktober 1958	, The Shawshank Redemption, Green La	, Adventure, Thriller, Science Fiction, F	
	Geoffrey Rush	The King's Speech	6. Juli 1951	, Pirates of the Caribbean: The Curse of th	, Animation, Family, Adventure, Fantas	
	Denzel Washington	Fences	28. Dezember 1954	, Gladiator II, The Equalizer, The Equalize	, Crime, Drama, Thriller, Action, Histor	
	Tom Hanks	Cast Away	9. Juli 1956	, Here, The Polar Express, Forrest Gump,	, Comedy, Drama, Romance, Animatic	

To Do

The project is not finished yet. There are still some things to do:

- Update the User Interface. The User Interface is not really user-friendly.
- Get awards from the TMDB database
- Show movie year
- Get all time movies