SI507 Final Project Report

Keye Chen (keyechen@umich.edu)

1. Project Code:

- **Link to GitHub Repository:** https://github.com/KeyeChen0/SI507FinalProj.git
- **README:**
  Run the 'final.py' directly.
  The program will read the CSV files and initialize a dataframe called 'extracted_df'.
  The program supports 4 functions:
  1. **Single Movie Lookup:** The user enters a movie name and get relevant information.
  2. **Gener-Based Recommendations:** The user enters either a movie name or a list of genres and get relevant recommendations.
  3. **Distance between Casts/Crews:** The user enters either two crews' names or two casts' names and then get the shortest distance between them.
  4. **Correlation Analysis:** The user enters two movies name and get the similarity score.
- **Any Required Python Packages:** requests, pandas, numpy

2. Data Sources:

2.1 movies_metadata.csv

- **URL:** https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data
- **Format:** CSV
- **Accession and Usage:** Use pd.read_csv to store the data.
- **Summary:** Contains 'original_title', 'release_date', 'overview' columns, storing detailed information about each film record which is distinguished by 'id' column.
- **Note:** Due the uploading file limit in GitHub, the 'movie_metadata' is cut at the first 100 rows. And the partial data is shown in GitHub as 'movie_metadata_cut.csv'.

2.2 credits.csv

- **URL:** https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data
- **Format:** CSV
- **Records:** 45476 records.
- **Accession and Usage:** Use pd.read_csv to store the data.
- **Summary:** Contains 'cast', 'crew' columns, storing the casts and crews information about each film record which is distinguished by 'id' column. 'cast' and 'crew' columns contain complex information other than the name, needed to be manipulated.
- **Note:** Due the uploading file limit in GitHub, the 'credits' is cut at the first 100 rows. And the partial data is shown in GitHub as 'credits_cut.csv'.

2.3 Web API

- **URL:** **https://www.omdbapi.com/**
- **Format:** API

- **Records:** 45466 records.
- **Accession and Usage:** Use pd.read_csv to store the data. The cache is used.
- **Summary:** It is an API that gives responses to queries about movie names. Contains cast, crew, overview, release_year, title information which is needed in this project.

3. Data Structure

3.1 'extracted_df'

- **README:** It is a dataframe data structure. The columns contains 'original title' [string], 'genres' [list series], 'cast' [list series], 'crew' [list series], 'release_year' [int], 'overview' [string]. Each row is a record of a film.
- **Construction Python Code:** See in 'extracted_df_construction_code.py', the code is selected from 'final.py'.
- **JSON file:** See in 'extracted_df.json'. Note: due the uploading file limit in GitHub, the 'extracted_df' is cut at the first 100 rows.
- **Python File Read JSON:** See in 'read_extracted_df.py'.
- **Screenshot:**

```
pd.read_json('extracted_df.json')
```

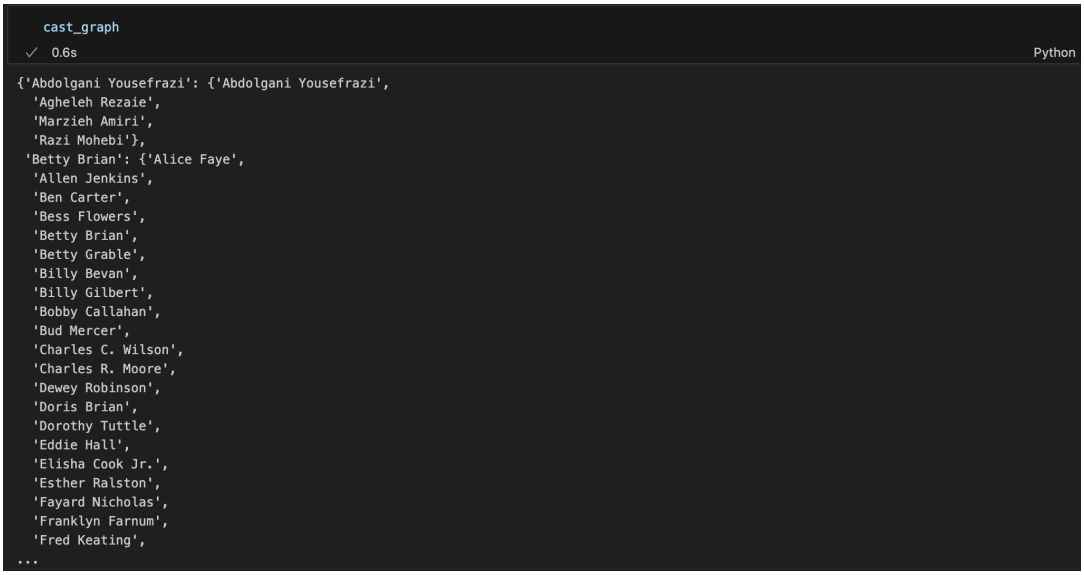
✓ 0.0s Python

	original_title	genres	cast	crew	release_year	overview
0	toy story	[Animation, Comedy, Family]	[Tom Hanks, Tim Allen, Don Rickles, Jim Varney...	[John Lasseter, Joss Whedon, Andrew Stanton, J...	1995	Led by Woody, Andy's toys live happily in his ...
1	jumanji	[Adventure, Fantasy, Family]	[Robin Williams, Jonathan Hyde, Kirsten Dunst,...	[Larry J. Franco, Jonathan Hensleigh, James Ho...	1995	When siblings Judy and Peter discover an encha...
2	grumpier old men	[Romance, Comedy]	[Walter Matthau, Jack Lemmon, Ann-Margret, Sop...	[Howard Deutch, Mark Steven Johnson, Mark Stev...	1995	A family wedding reignites the ancient feud be...
3	waiting to exhale	[Comedy, Drama, Romance]	[Whitney Houston, Angela Bassett, Loretta Devi...	[Forest Whitaker, Ronald Bass, Ronald Bass, Ez...	1995	Cheated on, mistreated and stepped on, the wom...
4	father of the bride part ii	[Comedy]	[Steve Martin, Diane Keaton, Martin Short, Kim...	[Alan Silvestri, Elliot Davis, Nancy Meyers, N...	1995	Just when George Banks has recovered from his ...
...	...	...	...	...	...	...
95	la haine	[Drama]	[Vincent Cassel, Hubert Koundé, Saïd Taghmaoui...	[Mathieu Kassovitz, Christophe Rossignon, Gill...	1995	Aimlessly whiling away their days in the concr...
96	shopping	[Action, Adventure, Drama, Science Fiction, Th...	[Sadie Frost, Jude Law, Sean Pertwee, Fraser J...	[Paul W.S. Anderson, Paul W.S. Anderson, Lauri...	1994	A dark, hip, urban story of a barren and anony...
97	heidi fleiss: hollywood madam	[Documentary]	[Nick Broomfield, Heidi Fleiss, Madam Alex, Iv...	[Nick Broomfield, Nick Broomfield, Paul Kloss,...	1995	A documentary crew from the BBC arrives in L.A...
98	city hall	[Drama, Thriller]	[Al Pacino, John Cusack, Bridget Fonda, Danny ...	[Jerry Goldsmith, Harold Becker, Harold Becker...	1996	The accidental shooting of a boy in New York l...
99	bottle rocket	[Comedy, Crime, Drama]	[Luke Wilson, Owen Wilson, Lumi Cavazos, Andre...	[Wes Anderson, Wes Anderson, Owen Wilson, Wes ...	1996	Upon his release from a mental hospital follow...

3.2 'cast_graph'

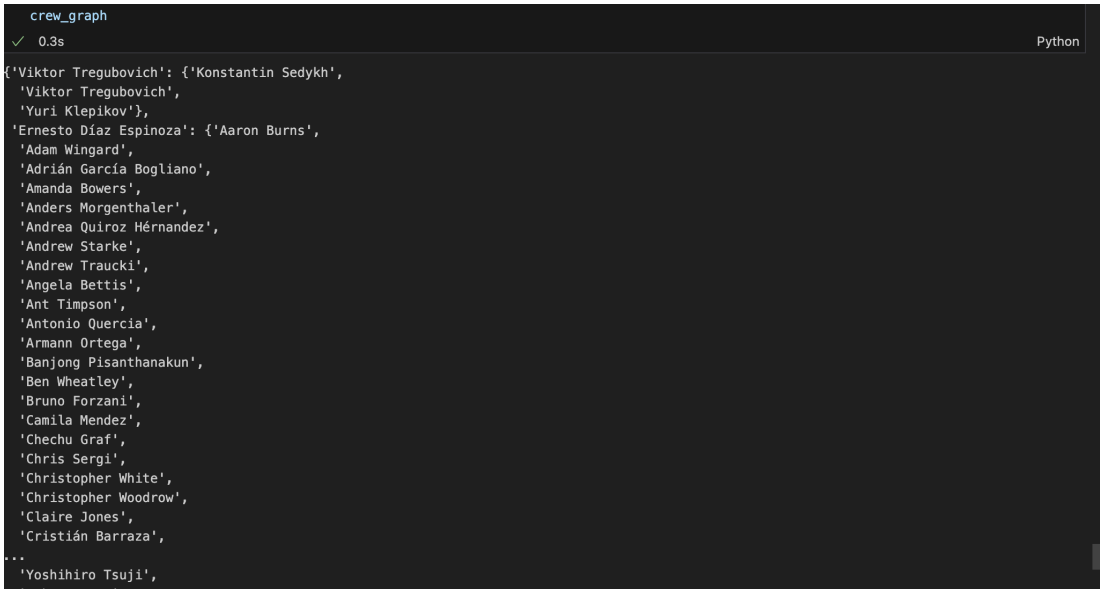
- **README:** It is a dictionary. The keys are casts' name, and the items are sets containing casts' name who have cooperated with the keys.
- **Construction Python Code:** See in cast_graph_construction_code.py', the code is selected from 'final.py'.
- **JSON file:** See in 'cast_graph.json'.
- **Python File Read JSON:** See in 'read_cast_graph.py'.
- **Note:** Due the uploading file limit in GitHub, the 'cast_graph.json' only contains the first 1000 casts.

- **Screenshot:**



3.3 'crew_graph'

- **README:** It is a dictionary. The keys are crews' name, and the items are sets containing crews' name who have cooperated with the keys.
- **Construction Python Code:** See in crew_graph_construction_code.py', the code is selected from 'final.py'.
- **JSON file:** See in 'crew_graph.json'.
- **Python File Read JSON:** See in 'read_crew_graph.py'.
- **Note:** Due the uploading file limit in GitHub, the 'crew_graph.json' only contains the first 1000 casts.
- **Screenshot:**



4. Interaction/Presentation

- **README:**

  Run the 'final.py' directly.

  The program will read the CSV files and initialize a dataframe called 'extracted_df'.

  The program supports 4 functions:

  1. **Single Movie Lookup:** The user inputs '1' to use this function. Then the user is asked to input a movie name. If the movie does not exist, the program will use the API to grab the movie information and add this to both the 'cache.json' and the bottom of the 'extracted_df'. Then the program will output an overview of the movie. If there is more than one movie sharing the same title, the program will ask the user to input the specific release year and then based on this output an overview. If the movie still does not exist, the program will output 'Movie not found, please try again.'
  2. **Gener-Based Recommendations:** The user inputs '2' to use this function. The user could either enter the genres or the movie names to get the recommendations.

     2.1 Enter 'genre': The user will be asked to enter a series of genres. Additionally, this part supports fuzzy search. If the user enters something not relevant to the genres, or misspelling the genre words, the program informs the user of the relevant genre choices. Eventually, the program will give a list of the 10 most relevant movie titles.

     2.2 Enter 'movie': The user will be asked to enter the movie name. Then the program will give a list of the 10 most relevant movie titles.
  3. **Distance between Casts/Crews:** The user inputs '3' to use this function. The user could explore the shortest path between two either casts or crews. The initialization of this graph would take a few seconds.

     3.1 Enter 'cast': The user will be asked to enter two actors' names. The program would output the shortest and detailed path between them. If the path doesn't exist, the program would output 'There is no path exists.' If any actor's name doesn't exist, the program would put 'Actor not found, please try again.'

     3.2 Enter 'crew': Similar to the 3.1.
  4. **Correlation Analysis:** The user inputs '4' to use this function. The user would be asked to input two correct movie names, and the program would output a correlation score based on the similarity of their crews, casts, and genres.
  5. **Exit:** The user inputs '5' to exit.

5. Demo Link:

https://drive.google.com/file/d/1Y9r2EOmTXU6OlQUJl4o8XJMPDbPi9REN/view?usp=sharing