‘Bingewatch’ Movie Recommendation System

Bianca Zlavog
Florencia Marcaccio
Mansi Rathod
Sanjana Gupta

Outline

- Background
- Data
- Use Cases
- Demo
- Design
- Project Structure
- Lessons Learned
- Future Work

Background

With so many streaming platforms available, it can be overwhelming to browse every catalog to find a movie that you'll like.

We aim to provide a personalized streaming experience based on the user's preferences with a movie recommendation tool.

Data

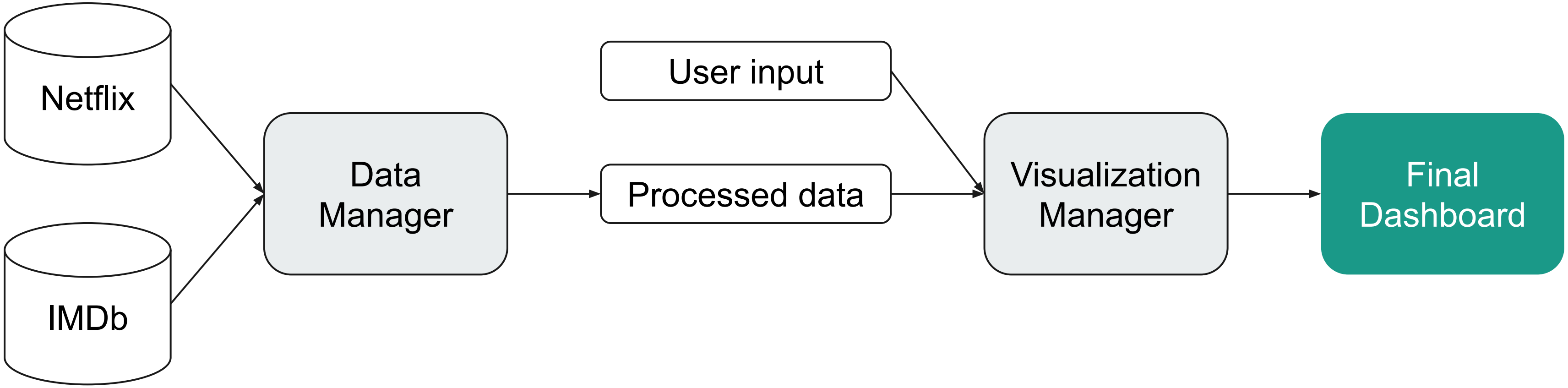
Source	Files	Features	Highlights
Netflix Prize Data (Kaggle Dataset)	Movie Ratings: combined_data_1.txt, combined_data_2.txt, combined_data_3.txt, combined_data_4.txt	MovieID, UserID, Rating, Date of Rating	Contains 100,480,507 ratings that 480,189 users gave to 17,770 movies.
	Movie Titles: movie_titles.csv	MovieID, Title, Year Released	Contains the movie title and released year for the 17,770 movies in the dataset.
IMDb Datasets	Basic information of each title: title.basics.tsv.gz	Identifier, Title, Content Type, Genres, Start and End Year, Minutes	Contains 6,842,632 titles, but only 736,380 are movies/tv series.
	Rating: title.ratings.tsv.gz	Identifier, Average Rating, Number of Votes	Only 305,357 of the previously filtered titles appear on this file.

Use Cases

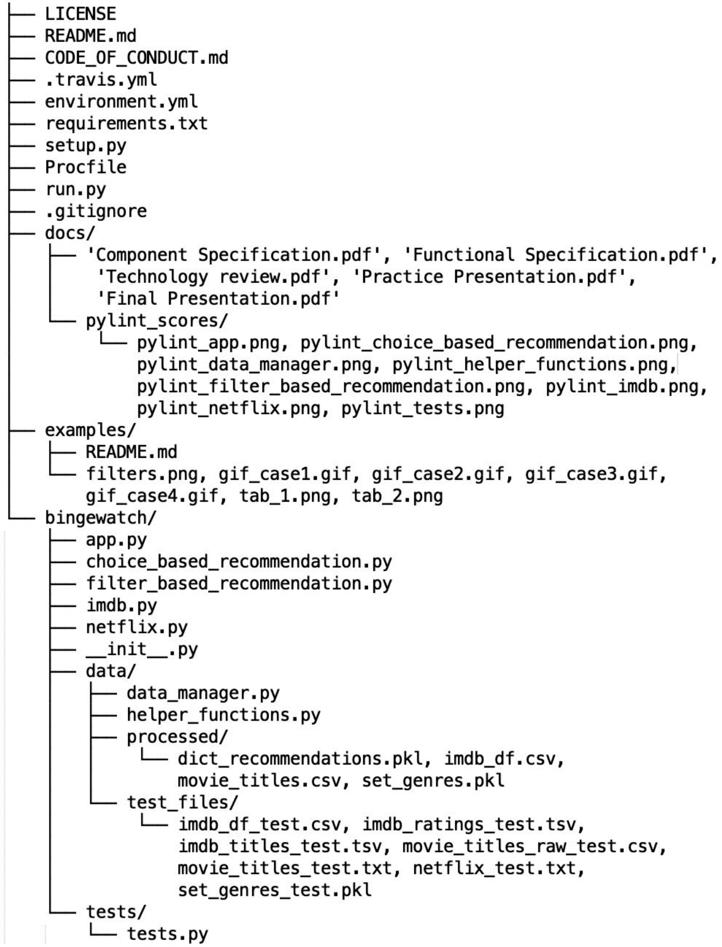
Use Case	**The user wants to watch a movie similar to a movie of their choice.**	**The user wants to see the top 10 movies of a particular year.**	**The user wants to see the top 10 movies from a particular genre.**
User	Inputs the movie they liked.	Inputs a particular year.	Inputs a genre.
Tool	Displays a list of the 10 most similar movies to the one input by the user, based on cosine similarity.	Displays a list of the top 10 movies from that year, based on average rating and number of votes.	Displays a list of the top 10 movies from that genre, based on average rating and number of votes.

Demo

Design


Netflix
IMDb
Data Manager
User input
Processed data
Visualization Manager
Final Dashboard

Project Structure



```
├── LICENSE
├── README.md
├── CODE_OF_CONDUCT.md
├── .travis.yml
├── environment.yml
├── requirements.txt
├── setup.py
├── Procfile
├── run.py
├── .gitignore
├── docs/
│   ├── 'Component Specification.pdf', 'Functional Specification.pdf',
│   │    'Technology review.pdf', 'Practice Presentation.pdf',
│   │    'Final Presentation.pdf'
│   └── pylint_scores/
│        └── pylint_app.png, pylint_choice_based_recommendation.png,
│            pylint_data_manager.png, pylint_helper_functions.png,
│            pylint_filter_based_recommendation.png, pylint_imdb.png,
│            pylint_netflix.png, pylint_tests.png
├── examples/
│   ├── README.md
│   └── filters.png, gif_case1.gif, gif_case2.gif, gif_case3.gif,
│       gif_case4.gif, tab_1.png, tab_2.png
└── bingewatch/
    ├── app.py
    ├── choice_based_recommendation.py
    ├── filter_based_recommendation.py
    ├── imdb.py
    ├── netflix.py
    ├── __init__.py
    ├── data/
    │   ├── data_manager.py
    │   ├── helper_functions.py
    │   ├── processed/
    │   │    └── dict_recommendations.pkl, imdb_df.csv,
    │   │        movie_titles.csv, set_genres.pkl
    │   └── test_files/
    │        └── imdb_df_test.csv, imdb_ratings_test.tsv,
    │            imdb_titles_test.tsv, movie_titles_raw_test.csv,
    │            movie_titles_test.txt, netflix_test.txt,
    │            set_genres_test.pkl
    └── tests/
        └── tests.py
```

Lessons Learned

- Difficulty working between languages (R and Python mainly)
- Building dashboards using Dash app
- Integrating tabs in Dash
- Maintaining correct file/module address in codes
- Travis CI and testing from the beginning

Future Work

- Add more filtering variables. For example, filter by decade instead of year, by director or actors, by MPAA content rating, etc.
- Add data source in order to include TV Shows in the Recommendation System based on a title chosen by the user.
- Continue improving the recommendation system itself.

Questions?

Thank you!