# GARKL Project Final Documentation

### Team Information

| Team Member | NetID |
|---|---|
| Captain: Abi Venkat | abinand2 |
| Karan Gulati | kgulati2 |
| Gabin Ntankeu | ntankeu2 |
| Leon Li | angl2 |
| Ratul Saha | ratuls2 |

## (1) Overview of the project

The purpose of this project was to give a better base recommendation than what Netflix has to offer. A user would input a Netflix title, and we would recommend other Netflix titles most similar to the input. By combining different movie data sources like IMDb, TMDb, and many others we were able to get more third party metadata about the Netflix title. We even created our own composite scoring rank based on this metadata so we could give better recommendations. The user also has the option to filter the recommendation results based on genre and/or actor, but the base recommendation is done on the metadata we have collected for the input movie title.

## (2) Team member contributions

Abi - Coordinated project meetings and tasks and worked on a majority of the documents required. Helped in research of datasets, algorithms to use that could relate to class topics, and pre-processing steps. Primarily worked on pre-processing of the data, improving accuracy of recommendation function, user error checking for the interface code, and fixing bugs within the recommendation function. Also worked on final video tutorial documentation.

Karan - Researched on datasets and pre-processing steps, worked on both proposal documents as well as progress report. Created and updated interface code of project, integrated other team members functional code into user UI. Spoke about interface in video recording.

Gabin - set up discord group for occasional meeting, researched datasets, contribute in structuring the code and assist the video recording

Leon - Researched and found datasets and processed data. Worked on the first version of the recommendation function. Worked on documents.

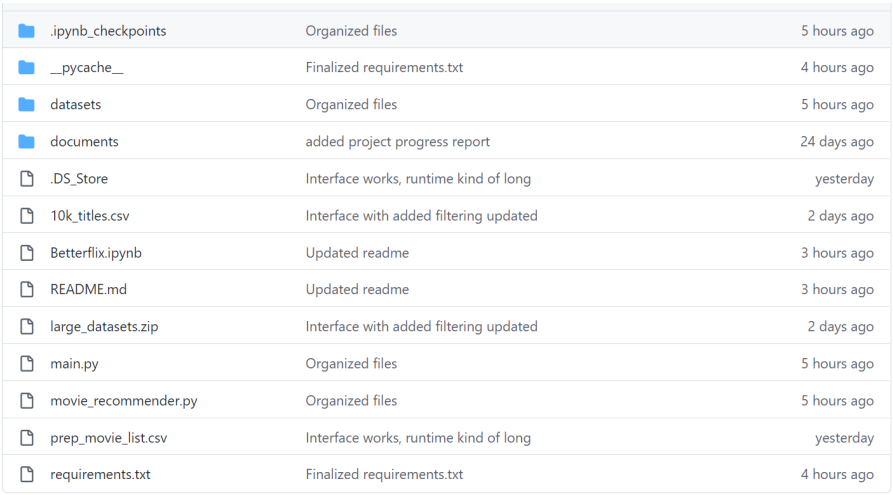
Ratul - Ideation of projects; conceptualizing features for the movie recommender system; identified useful datasets with data attributes that might be useful for analysis; assisted in development of main recommender logic; testing; documentation.

## (3) Related work and used libraries / models / previous projects

We used this basic movie recommendation tutorial as a starting point/reference https://www.kaggle.com/code/gazu468/movie-recommendation-system-with-basic-concept.
We used pandas and numpy to read and process the data from the csv files. Once we created our post-processed data set we used sklearn to do feature extraction and vectorization of the data set. Sklearn also handled computing the cosine similarity between the feature vectors of the user's imputed movie title, and the rest of the data. Finally we used PySimpleGUI to create a python graphic interface to make the software look like an application.

## (4) Code structure

Below is the view of our project folder in github.
Github location: https://github.com/Atven27/BetterNetflixRecommender



| | | |
|---|---|---|
| .ipynb_checkpoints | Organized files | 5 hours ago |
| __pycache__ | Finalized requirements.txt | 4 hours ago |
| datasets | Organized files | 5 hours ago |
| documents | added project progress report | 24 days ago |
| .DS_Store | Interface works, runtime kind of long | yesterday |
| 10k_titles.csv | Interface with added filtering updated | 2 days ago |
| Betterflix.ipynb | Updated readme | 3 hours ago |
| README.md | Updated readme | 3 hours ago |
| large_datasets.zip | Interface with added filtering updated | 2 days ago |
| main.py | Organized files | 5 hours ago |
| movie_recommender.py | Organized files | 5 hours ago |
| prep_movie_list.csv | Interface works, runtime kind of long | yesterday |
| requirements.txt | Finalized requirements.txt | 4 hours ago |

The project folder contents are detailed below:

- **datasets** folder: contains all the raw data we use. These datasets are from kaggle and github.
- **main.py**: entrance file for our project. Running this file will show the user interface. This file contains logic for creating and handling input from the user interface.
- **movie_recommender.py:** This file contains a movie recommendation function.
- **prep_movie_list.csv:** This is the processed data and used for recommendation.

- **betterflix.ipynb:** This is a jupyter notebook version of our interface application.
- **requirements.txt:** Requirement file which is used by pip to install necessary packages.
- **documents:** This folder contains all the report documents.

The code runs like this. First a user enters an input. Then we load in our prep_movie_list.csv data into a dataframe and computer feature vectors on our data. In our recommender code we look up the input movie title and do a cosine similarity between the input movie's feature vectors and the rest of the data set. Then we output the top 10 movies that have the highest composite score along with other information about those titles. To compute the composite score we used a 60% weighting on IMDb score and 40% weighting on popularity score.

**(5) Detailed instructions for reviewers to set up and run code, including possible errors or blockers**

Our code uses Python to run. Please have at least version 3.9 and above and have pip installed. Once you are inside the repository and have a terminal running, run our software by simply typing:

```
pip install -r requirements.txt
```

This will install all necessary packages.
Then simply to run the program by typing:

```
python ./main.py
```

It may take 15-20 seconds for the code to compile depending on your machine. There are quite a lot of features to extract and unfortunately we couldn't find a way to offload this into a readable text file or something so it needs to be compiled each time.

If you would like a faster version, use the Jupyter notebook Betterflix.ipynb file version we have. There are instructions in that version of our application for how to use it, and our video documentation has instructions as well.

The application will take a Netflix title input from the user. The user has to enter a Netflix title, but the other two inputs (genre and cast member) are optional. Genre and cast member are used to filter the recommendation results even further. The results returned are a list of movies that are most similar to the input Netflix title. If a user would like to then filter those results to see only titles from that list that are of a specific genre or have a specific actor they can do that with the genre and cast member inputs. Examples of this are shown in the video documentation.