

COSC480 PROJECT REPORT

Anime Recommendation System

Name: Shantikrishna Panicker

Student ID: 26037298

1. Anime Recommendation App - Structure

The objective of this project is to develop a recommendation system that helps users discover new anime based on their viewing history. This system uses a hybrid methodology that combines content-based and collaborative filtering techniques. Built using Python, the project includes various components such as data preprocessing, recommendation algorithms, and visual pattern analysis.

The dataset features a range of attributes related to anime, including genres, ratings, and popularity indicators. The application is designed to be user-friendly, allowing users to input their preferences, which the system then uses to generate personalized recommendations.

2. Development Process

To initiate the project, both the front-end and back-end components must be launched independently.

2.1. Launch Front-end

For the front-end, begin by opening the Anaconda Prompt and confirming that Streamlit is installed. If it is not present, use pip to install Streamlit, and if it is already installed, ensure it is updated to the latest version. For further installation guidance, consult ChatGPT.

```
pip install streamlit
```

2.2. Launch Back-end

Regarding the back-end, ensure that Python and Jupyter Notebook are installed on your system. Check that the Anaconda Prompt is functioning properly, and navigate to the project directory that contains the relevant Jupyter Notebooks and scripts.

2.3. Preprocessing Data

The 'process_data.py' script is responsible for preparing the dataset, which involves cleaning up missing values, converting genres into a suitable format, and making sure the dataset is ready for analysis.

2.4. Recommendation Logic

The 'recommendation_logic.py' script serves as the foundation of the recommendation engine, using a combination of content-based and collaborative filtering methods. This

module is essential for the recommendation system, as it identifies which anime to recommend to users.

2.5. Visualization

The 'viz_patterns.py' script is used for data visualization as it highlights trends, including the most favored genres, top-rated anime, and those with the highest user engagement.

3. Instructions to Use this App

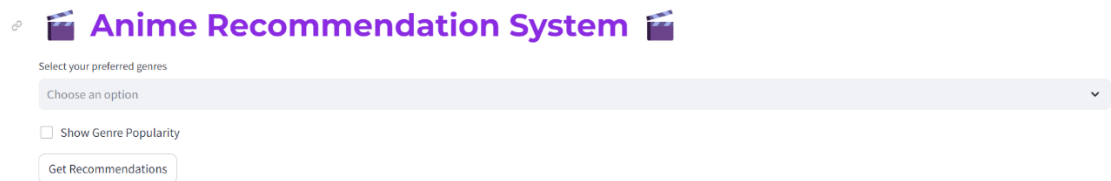
To launch the anime recommendation system using Streamlit, first open the Anaconda terminal. Navigate to the directory containing the downloaded zip file, extract the folder, and enter the command "streamlit run app.py" in the terminal. This will open a web browser displaying an interactive graphical user interface for anime recommendations. The project provides a user-friendly platform that will enable users to input their anime preferences, resulting in personalized suggestions.

```
>streamlit run app.py
```

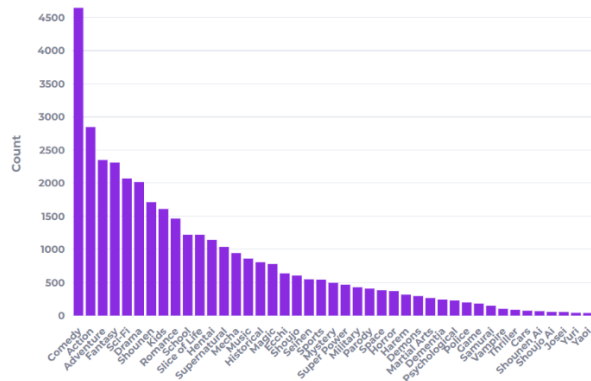
3.1. Navigation in GUI

When users open the application, they can input a list of anime they have watched before, which will trigger the recommendation system to generate a variety of new anime suggestions. The interface allows users to filter these recommendations by factors like genre, rating, or popularity.

- Interactive GUI will display as shown below:

The screenshot shows the web interface of the 'Anime Recommendation System'. At the top, there is a header with a small icon, the title 'Anime Recommendation System' in purple, and another small icon. Below the header, there is a section titled 'Select your preferred genres' which contains a dropdown menu with the text 'Choose an option'. Underneath the dropdown, there is a checkbox labeled 'Show Genre Popularity'. At the bottom of this section, there is a button labeled 'Get Recommendations'.

- Click on 'Show Genre Popularity' to display a graph of popular genres.

☒ Show Genre Popularity

- [Get Recommendations](#)

	name	genre	rating
0	Kimi no Na wa.	Drama, Romance, School, Supernatural	9.37
10	Clannad: After Story	Drama, Fantasy, Romance, Slice of Life, Supernatural	9.06
16	Shigatsu wa Kimi no Uso	Drama, Music, Romance, School, Shounen	8.92
21	Rurouni Kenshin: Meiji Kenkaku Romantan - Tsuioku-hen	Action, Drama, Historical, Martial Arts, Romance, Samurai	8.83
25	Suzumiya Haruhi no Shoushitsu	Comedy, Mystery, Romance, School, Sci-Fi, Supernatural	8.81
26	Monogatari Series: Second Season	Comedy, Mystery, Romance, Supernatural, Vampire	8.8
35	Howl no Ugoku Shiro	Adventure, Drama, Fantasy, Romance	8.74
39	Bakuman. 3rd Season	Comedy, Drama, Romance, Shounen	8.71
45	Kara no Kyoukai 5: Mujun Rasen	Action, Drama, Mystery, Romance, Supernatural, Thriller	8.68
50	Yojouhan Shinwa Taikei	Mystery, Psychological, Romance	8.65

4. Things That Went Well

The recommendation system successfully delivers tailored suggestions by merging user input with smooth integration of content-based filtering methods. The data preprocessing phase was enhanced to so that the dataset was thoroughly cleaned and optimally prepared for the recommendation system's functionality. Visual representations effectively represent trends and patterns in anime ratings and popularity, offering users valuable insights.

5. Challenges Faced

- Addressing the issue of missing data in the anime dataset presented significant challenges, necessitating various preprocessing steps.
- Modifying the recommendation algorithm to achieve an effective balance between content-based and collaborative filtering was a complex task, as it was essential to properly weigh both approaches.
- Improving the system's ability to handle large datasets while maintaining performance was a challenging task that required multiple iterations.

6. Future Developments

- Improving the accuracy of recommendations can be accomplished by utilizing advanced algorithms such as matrix factorization or neural collaborative filtering.
- Incorporating user feedback mechanisms will enable ongoing adjustments to the recommendation engine based on genuine user preferences.
- Adding interactive features that allow users to emphasize specific criteria, like genre, rating, or release date, can further tailor the experience.
- Additionally, leveraging real-time data from external anime databases will help keep recommendations current.
- Lastly, expanding the dataset to include a wider range of user interactions, such as reviews and viewing history, will greatly improve the precision of predictive results.

7. Appendix

I used ChatGPT to help integrate and understand the Streamlit graphical user interface.

- Prompt: *"What steps do I need to follow to add my anime recommendation feature to a Streamlit application?"*